Numerical Algorithms 5 (1993) 577–587

Scattered data interpolation using minimum energy Powell–Sabin elements and data dependent triangulations

Shmuel Rippa *

Department of Applied Mathematics and Theoretical Physics, Silver Street, Cambridge, England CB3 9EW

A popular approach for obtaining surfaces interpolating to scattered data is to define the interpolant in a piecewise manner over a triangulation with vertices at the data points. In most cases, the interpolant cannot be uniquely determined from the prescribed function values since it belongs to a space of functions of dimension greater than the number of data points. Thus, additional parameters are needed to define an interpolant and have to be estimated somehow from the available data. It is intuitively clear that the quality of approximation by the interpolant depends on the choice of the triangulation and on the method used to provide the additional parameters. In this paper we suggest basing the selection of the triangulation and the computation of the additional parameters on the idea of minimizing a given cost functional measuring the quality of the interpolant. We present a scheme that iteratively updates the triangulation and computes values of the additional parameters so that the quality of the interpolant, as measured by the cost functional, improves from iteration to iteration. This method is discussed and tested numerically using an energy functional and Powell–Sabin twelve split interpolants.

1. Introduction

The problem of scattered data interpolation in two dimensions is that of constructing a function, say F_I, such that

$$F_I(x_i, y_i) = f_i, \quad (x_i, y_i) \in V, \tag{1}$$

where $V = \{(x_i, y_i)\}_{i=1}^N$ is a set of distinct and non-collinear points, called data points, and $f_i, i = 1, \ldots, N$, are real numbers that are usually thought of as being the values of a function F, that is to be approximated by F_I, at the data points. This problem is important in many areas of science and technology and many methods for its solution were suggested in the literature; see e.g., the survey papers of Barnhill [2], Franke [9] (containing a numerical comparison between 29 methods), Nielson and Franke [12], and Schumaker [16].

* E-mail: shmulik@orbot.co.il

In this paper we restrict our attention to a class of schemes in which the interpolant is constructed over a triangulation with vertices at the data points in such a way that its restriction to each triangle of the triangulation is a function of a simple form, e.g. a low order polynomial, and is moreover continuously differentiable over the triangulated region. The dimension of the space of functions containing the interpolant is usually greater that N and thus the interpolation conditions (1) do not define it uniquely. By using a suitable basis we can write all interpolants defined over a fixed triangulation T as

$$S_{T,f,g}(x,y) = \sum_{i=1}^{N} f_i P_i(x,y) + \sum_{i=1}^{M} g_i Q_i(x,y), \tag{2}$$

where $f = (f_1, \ldots, f_N), g = (g_1, \ldots, g_M)$ is a real-valued vector, called the parameter vector, containing the additional parameters needed to specify an interpolant from the space and P_i and Q_i are basis functions that depend on T. It is common to construct the interpolation space so that the components of g determine values, or values of derivatives, of the interpolant at various points in the triangulated region. For example, in many practical cases the elements of g determine first order derivatives of the interpolant at the data points, i.e. we have that $\partial S_{T,f,g}/\partial x|_{(x_i,y_i)} = g_{2i-1}, \partial S_{T,f,g}/\partial y|_{(x_i,y_i)} = g_{2i}, i = 1, \ldots, N$. It is intuitively clear that the quality of approximation by $S_{T,f,g}$ depends on the choice of triangulation and parameter vector and that their computation is central to any interpolation scheme. A common approach is to generate the interpolant in two steps: First a "good" triangulation is constructed and then the parameter vector is computed.

Traditional triangulation methods [17] aim at constructing triangulations that contain as many "well shaped", nearly equiangular, triangles as possible. Long and thin triangles were considered to be bad for interpolation and were to be avoided whenever possible. An attractive choice for a well shaped triangulation is the Delaunay triangulation [11,17] that maximizes (over all triangulations of the same set of points) the minimal angle (in a triangulation). This extensively studied triangulation has many interesting properties and efficient algorithms exist for its construction. Thus it became a de facto standard for the choice of an optimal, geometrically well shaped, triangulation. However, recent studies (see Rippa [15] and references therein) suggest that, for piecewise linear approximation, it is important that the shape of triangles in the triangulation is adapted to the behavior of the approximated function. In particular, well shaped triangulations, e.g. Delaunay, are not well suited for approximating functions with a preferred direction, that is, a direction in which the function has high curvature as compared to that in the perpendicular direction. In such cases better approximation can be provided by triangles that are thin in the preferred direction and long in the orthogonal direction. Dyn et al. [5] proposed algorithms for producing *data dependent* triangulations that are adjusted to the behavior of the approximated function and show that a significant improvement in the quality of fit can be obtained when the

interpolant is constructed over such triangulations instead of the Delaunay triangulation.

In order that the interpolant (2) can be fully defined on a given triangulation T we have to compute the parameter vector g. When components of g determine values, or values of derivatives, of $S_{T,f,g}$ then a natural approach is to assign to these components corresponding values, or values of derivatives, of some higher order approximation to F; see Nielson and Franke [12] and Stead [18] for further discussion and references. A different approach [1,4,12] is to compute g so that $S_{T,f,g}$ is optimal in some sense, e.g. it minimizes a given functional, among all interpolants $S_{T,f,h}, h \in \mathbb{R}^M$.

In this paper we suggest considering the generation of the triangulation and the computation of the parameter vector in a unified framework, combining the data dependent approach for constructing triangulations with the method of computing the parameter vector by functional minimization. To this aim we associate with any interpolant $S_{T,f,g}$ a value $C(S_{T,f,g})$ of a cost functional C measuring the quality of the interpolant, i.e. the smaller is $C(S_{T,f,g})$, the better is $S_{T,f,g}$. To make the presentation clearer we restrict ourselves to a specific interpolation scheme and a specific cost functional. The interpolant is called the twelve split Powell–Sabin element (PS12) and is described in section 2 and C is the "roughness", or energy, functional defined by

$$C(S_{T,f,g}) = \sum_{T_i \in T} \int_{T_i} \left(\left(\frac{\partial^2 S_{T,f,g}}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 S_{T,f,g}}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 S_{T,f,g}}{\partial y^2} \right)^2 \right) \mathrm{d}x\mathrm{d}y, \quad (3)$$

where $T_i \in T, i = 1, \ldots, N$, are the triangles in the triangulation T. This functional has an interesting physical interpretation as measuring the bending energy of a surface and was considered by many authors, see, e.g., Alfeld [1] and Cline and Renka [4].

This paper is organized as follows. In section 2 we describe the PS12 interpolant and in section 3 we consider the problem of computing an optimal, minimum energy, interpolant among all interpolants defined over all possible triangulations and parameter vectors. Since, in most cases, it is very difficult to attain the optimal interpolant we also present efficient, locally optimal, schemes in which the triangulation and the parameter vector are updated iteratively. The iterations continue as long as the energy of the interpolant is reduced from iteration to iteration. In section 4 we present a summary of our numerical experiments with the suggested schemes and in section 5 we make concluding remarks.

2. Powell-Sabin 12 split interpolant

Two common requirements for an interpolant defined by (2), in order that it can be computed efficiently, are that its restriction to any triangle of the triangula-

tion (i) is a function of a simple form and (ii) is completely defined by specifying its values, or values of its derivatives, on vertices, edges, and points in the interior of the triangle. The degree of the lowest order polynomial satisfying these requirements is five (see, e.g., [8]) which is too high for practical applications. Furthermore, the necessary data for specifying such an interpolant must include derivatives up to second order whilst we are only interested in C^1 continuity. An alternative approach is to divide each triangle, called a macrotriangle in this context, into several subtriangles and to construct the interpolant as a low order polynomial on each subtriangle. Powell and Sabin [13] proposed to split each macrotriangle into twelve subtriangles (see fig. 1) and to define the interpolant to be quadratic on each of them. They showed that an interpolant, that we shall refer to as PS12, is uniquely defined as a C^1 function over the triangulated region by specifying its values and the values of its first order derivatives at all vertices and the values of its normal derivatives (outward or inward normal) at the midpoints of all edges of the triangulation. The parameters vector g of (2) contains values of first order derivatives of the interpolant at all data points and values of its normal derivatives (it does not matter in which direction) at midpoints of all edges of the triangulation. Since the number of edges in a triangulation is not greater than $3N$, N being the number of vertices, we have that the dimension of g is not larger than $5N$. A common way to reduce the dimension is to require that the normal derivative of the interpolant is a linear polynomial (instead of being piecewise linear) on each edge. In this way the dimension of the parameter vector is only $2N$ while the result-

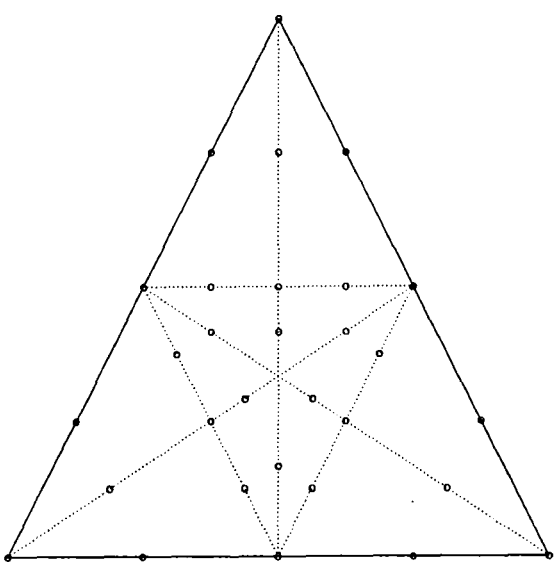

Fig. 1. A twelve-split macrotriangle. The PS12 interpolant, that is quadratic on each subtriangle, is defined by specifying its value and gradient at all vertices and its normal derivative at the midpoint of all edges of the macrotriangle.

ing *reduced* interpolant, that we call R-PS12, is still a C^1 function over the triangulated region.

Other popular interpolants of this type are [8] the Clough–Tocher interpolant obtained by subdividing each triangle into three and constructing the interpolant to be cubic on each subtriangle (this is probably the best known of those schemes) and the Powell–Sabin six-split interpolant defined by dividing each triangle into six and constructing the interpolant to be quadratic on each subtriangle.

3. Minimizing the energy of a PS12 element

Let us consider the bilinear form a,

$$a(u,v) = \sum_{T_i \in T} \int_{T_i} (u_{xx}v_{xx} + 2u_{xy}v_{xy} + u_{yy}v_{yy})\mathrm{d}x\mathrm{d}y\,,$$

defined over a fixed triangulation T. The energy functional (3) can be easily expressed, by using the representation (2), as a quadratic form in g, that is, as

$$C(S_{T,f,g}) = a(S_{T,f,g}, S_{T,f,g}) = (Ag,g) + 2(g,b) + c\,,$$

where the $M \times M$ matrix A, the M-dimensional vector b and the constant c are defined by

$$A_{ij} = a(Q_i,Q_j), \quad b_i = \sum_{j=1}^{N} f_j a(Q_i,P_j), \quad c = a\left(\sum_{j=1}^{N} f_j P_j, \sum_{j=1}^{N} f_j P_j\right).$$

It is easy to verify that the (sparse) matrix A is positive definite (see also [1,4]) and thus the vector $g_o(T) \in \mathbb{R}^M$ defined by $g_o(T) = -A^{-1}b$ is the unique energy minimizing vector, i.e. for which $C(S_{T,f,g_o(T)}) = \min_{g \in \mathbb{R}^M} C(S_{T,f,g})$.

Since there are finitely many triangulations of a set of points and to each one of them there corresponds a unique optimal parameter vector, it follows that the optimal interpolant $S_{T^*,f,g_o(T)^*)}$, for which $C(s_{T^*,f,g_o(T^*)}) \leqslant C(S_{T,f,g})$ for any triangulation T and any M-dimensional vector g, always exists (of course, in general, it need not be unique).

In practice it might be very difficult to obtain an optimal interpolant and we are usually content with interpolants that are only locally optimal in some sense. The schemes that we present below iteratively update the triangulation and recompute the parameter vector in such a way that the energy of the interpolant is reduced from iteration to iteration. The initialization of the iterations is completed by computing an initial triangulation $T^{(0)}$ (the Delaunay triangulation is a sensible choice since it can be computed efficiently) and an initial parameter vector $g^{(0)}$ (in our numerical experiments we took $g^{(0)}$ to be the energy minimizing vector $g_o(T^{(0)})$ but, of course, it is possible to use any available method, for example [1,4,18], to compute a suitable initial vector). At the ith iteration we start with an interpolant

$S_{T,f,g}$ corresponding to a triangulation and a parameter vector computed in the previous iteration. An iteration step consists of transforming T into a new triangulation and computing a new parameter vector so that the energy of the interpolant defined over the transformed triangulation with the new parameter vector is smaller than that of $S_{T,f,g}$. A simple transformation of T was suggested by Lawson [10,11]: Consider an interior edge e of T that is the diagonal of a convex quadrilateral formed from the two triangles in T having e as their common edge. The triangulation T can be transformed into a triangulation, which we call $T^{(e)}$, by replacing e by the opposite diagonal of the quadrilateral. This local and simple transformation is also very general since, as proved by Lawson [10] and Dyn et al. [7], any triangulation can be transformed to any other one by a sequence of such transformations. An edge e is eligible for such a swap only if it is a diagonal of a convex quadrilateral and if $C(S_{T^{(e)},f,g^{(e)}}) < C(S_{T,f,g})$, where $g^{(e)}$ is a parameter vector computed for the transformed triangulation, e.g. by one of the procedures described below. If more than one edge is eligible for swapping then we have to decide which one to swap. Different strategies are presented and compared in [6] but none was found to be significantly better than the others. In our implementation we swap the edge E for which $C(S_{T^{(E)},f,g^{(E)}}) \leqslant C(S_{T^{(e)},f,g^{(e)}})$ for all eligible edges e. If no interior edge can be swapped, then the iterations terminate, and the resulting triangulation is called locally optimal. The iterations may end also if a prescribed number of edge swaps was made. Since the number of triangulations is finite and each edge swap results in reduction in the energy of the interpolants, the above scheme converges after a finite number of edge swaps to a locally optimal triangulation provided that the vector $g^{(e)}$ is defined uniquely by $T^{(e)}$ and f.

We turn now to the question of computing $g^{(e)}$ in order to define $S_{T^{(e)},f,g^{(e)}}$ over the transformed triangulation $T^{(e)}$. A possible method is to use the energy minimizing factor $g_o(T^{(e)})$ but this requires the solution of a system of linear equations and thus may be impractical for large sets of points. When g defines values of derivatives of the interpolant at the data points, as is the case for the (reduced) RPS12 interpolant, and the components of the initial vector $g^{(0)}$ contain adequate approximation to derivatives of F at the data points, then we can take $g^{(e)}$ to be equal to the initial vector for all iterations. For the (full) PS12 interpolant, the parameter vector contains also components defining values of its normal derivatives at midpoints of all edges of the triangulation and thus, when computing $g^{(e)}$, we must take into account the fact that the location of the midpoint and the direction of the normal to an edge change after an edge swap. Assuming that the components of g contain adequate approximations to derivatives of F at all data points and to normal derivatives of F at midpoints of all edges of T, then we suggest the following procedure: Take all components of $g^{(e)}$, *except* the one that defines the normal derivative of the resulting interpolant $S_{T^{(e)},f,g^{(e)}}$ at the midpoint of the swapped edge, to be equal to the corresponding components of g. The remaining element is computed so that the energy of $S_{T^{(e)},f,g^{(e)}}$ is minimized. This is equivalent to minimizing a quadratic function, with a positive leading coefficient, and is very simple

to implement. We note that the resulting parameter vector depends on $T^{(e)}$, on f and, also, on the previous parameter vector g. Thus the above iterative scheme is no longer guaranteed to converge in a finite number of steps. This did not cause any problems in our numerical experiments, in which we always observed convergence after a finite number of swaps.

Other variants of the above scheme can be obtained by considering different initial triangulations and parameter vectors, different strategies for swapping edges, and different methods for computing the parameter vector on the transformed triangulation. One such variant is the scheme proposed by Quak and Schumaker [14] in connection with (reduced) Clough–Tocher interpolants: The initial triangulation is Delaunay and the initial parameter vector was computed using a standard estimation procedure [4]. The parameter vector $g^{(e)}$ was taken to be equal to this initial vector for all iterations. No specific method for swapping edges was adopted and eligible edges are swapped as soon as detected by the program.

4. The numerical experiments

We have conducted extensive numerical testing of the iterative scheme discussed in section 3 involving the (full) PS12 and the (reduced) R-PS12 interpolants. Two scattered data sets consisting of 33 and 100 points, taken from Franke [9], and two regular grids, of 7×7 and 10×10 points, covering the unit square, were considered. Data vectors $f = (f_1, \ldots, f_N)$ were obtained for each set of points by evaluating various functions, including all the functions of [9], at the data points. Several initial triangulations, always including the Delaunay triangulation, were constructed and the vector minimizing the energy of the interpolant to f defined over the initial triangulation was taken as the initial parameter vector.

The quality of fit of an interpolant to f defined on a given triangulation was measured by computing various norms of the error function (namely the function that is the difference between the interpolant and the function from which f was sampled). We computed the L_1 norm and the squared L_2 norm of this "error" function, and all its first and second order partial derivatives, using a seven-point, fifth-degree, Radon integration formula over all triangles. The tables display the L_1 and L_2 norms of this error function in the "Values" row and the "L_1" and "L_2" columns respectively. The sum of the L_1 norms of all first (resp. second) order partials of the error is displayed in the "Gradient" (resp. "Second-der.") row and the "L_1" column. The rooted sum of the squared L_2 norms of all first (resp. second) order partials of the error is displayed in the "Gradient" (resp. "Second-der.") row and the "L_2" column.

The objective of the first set of experiments was to compare the minimal energy interpolant $S_o = S_{T,f,g_o(T)}$ and the interpolant $S_F = S_{T,f,g_F(T)}$ defined by the vector $g_F(T)$ containing the true derivatives of F at the data points and, for PS12 interpolants, the normal derivatives of F at midpoints of all edges. The results that we

obtained were by and large consistent over the large number of experiments. although there were counterexamples in each case. In general S_o was better than S_F for approximating second order derivatives but worse for approximating function values. The quality of approximation to first order derivatives is comparable but usually S_o is worse when the quality of approximation of measured using L_1 norms of the error in first order derivatives. Table 1 illustrates this behavior. It displays the errors for interpolants to data sampled from the function $F6 = \frac{1}{9}[64 - 81((x-0.5)^2 + (y-0.5)^2)]^{1/2} - 0.5$, and constructed over the Delauney triangulation of 33 points.

The next objective of our experiments was to test the iterative procedure for obtaining the locally optimal (minimal energy) triangulations presented in section 3. We found that this scheme tends to converge very fast to a locally optimal triangulation and, in almost all cases, different locally optimal trinagulations were obtained when starting from different initial triangulations. The Delaunay triangulation was always very close to being locally optimal (very few edge swaps performed) but the energy of the interpolant defined over it was often higher than that of interpolants defined over other triangulations.

Our final observation is that interpolants with higher energy provide in many cases better approximation than interpolants having lower energy. As an example, consider three interpolants $S_{T,f,g_o(T)}$ constructed for data taken from the function $F2 = (\tanh(9y - 9x) + 1)/9$ over three different, locally optimal, triangulations, $T^{(a)}$, $T^{(b)}$ and $T^{(c)}$ (see fig. 2), obtained by applying the iterative procedure starting from different initial triangulations of a set of 33 points. Table 2 displays the deviation of $F2$ and the interpolants defined over those three locally optimal triangulations. The quality of approximation to $F2$ on $T^{(b)}$ is much better than that on triangulation $T^{(a)}$ which was obtained after one edge swap from a Delaunay triangulation. The energy of the interpolant defined on $T^{(b)}$ is smaller than that of the interpolant defined on $T^{(a)}$. The interpolant defined on $T^{(c)}$ has the smallest energy

Table 1
Deviation of the function $F6$ from interpolants constructed over the Delaunay triangulation of 33 points.

Deviation in	R-PS12 interpolant			PS12 interpolant		
	L_1	L_2	Energy	L_1	L_2	Energy
Values (S_F)	0.00050	0.00103	21.5569	0.00044	0.00093	11.7800
Values (S_o)	0.00172	0.00254	4.9336	0.00150	0.00226	4.4064
Gradients (S_F)	0.01861	0.03470	21.5569	0.01367	0.02716	11.7800
Gradients (S_o)	0.03260	0.03236	4.9336	0.02832	0.02925	4.4064
Second-der. (S_F)	1.84107	3.93520	21.5569	1.08318	2.53758	11.7800
Second-der. (S_o)	0.79077	0.79226	4.9336	0.71080	0.78743	4.4064

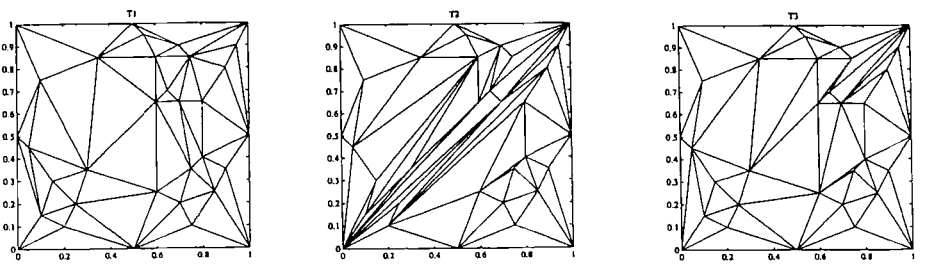

Fig. 2. Triangulations $T^{(a)}$ (left), $T^{(b)}$ (middle), $T^{(c)}$ (right).

but the quality of fit by the interpolant defined over it is similar to that provided by the interpolant defined on $T^{(a)}$, i.e. worse than the quality of fit provided by the interpolant defined on $T^{(b)}$. These findings are consistent with the conclusion of Quak and Schumaker [14] that the energy of an interpolant does not provide a clear indication of the quality of approximation.

Finally we note that the above experiments were also conducted using the six-split Powell–Sabin interpolant and the Clough–Tocher interpolant (both full and reduced types were tested). The results were similar to those obtained for the PS12 and R-PS12 interpolants. Numerical comparison between these three macroelements will be presented elsewhere.

5. Conclusions

In this paper we suggest an iterative procedure for interpolating scattered data which aims at minimizing the energy of the interpolant. The procedure generates an initial triangulation and computes and initial energy minimizing parameter vec-

Table 2
Deviation of the function $F2$ from its R-PS12 interpolant defined over three minimum energy triangulations of 33 points.

Deviation in	L_1	L_2	Energy
Values ($T^{(a)}$)	0.01061	0.01488	48.729
Values ($T^{(b)}$)	0.00405	0.00659	45.699
Values ($T^{(c)}$)	0.00982	0.01411	42.223
Gradient ($T^{(a)}$)	0.26961	0.25340	48.729
Gradient ($T^{(b)}$)	0.11501	0.12336	45.699
Gradient ($T^{(c)}$)	0.23839	0.22888	42.223
Second-der. ($T^{(a)}$)	7.73305	6.83612	48.729
Second-der. ($T^{(b)}$)	4.48987	4.31599	45.699
Second-der. ($T^{(c)}$)	6.70469	6.00280	42.223

tor. The triangulation is then iteratively changed, and the parameter vector recomputed, in such a way that the energy of the interpolant is further reduced.

The idea of defining a particular interpolant among all interpolants from a given space of functions by minimizing the energy is not new; see Alfeld [1] and Cline and Renka [4], and we found that this is a particularly good technique for approximating first and second order derivatives. The iterative scheme as suggested in this part is very sensitive to the choice of the initial triangulation and converges rapidly to a locally optimal triangulation. An interesting question is whether techniques like simulated annealing [3] can be used to find better locally optimal triangulations. Such methods try to avoid poor local minima by allowing edges to be swapped, in a controlled way, and also in cases where the energy of the interpolant defined over the transformed triangulation is increased.

The ideas described in this paper are, of course, applicable to any situation where it is desirable to choose the interpolant and the parameter factor so that the value of a given functional is minimized. We think of the functional being minimized as measuring the quality of interpolants, and believe that useful functionals can be constructed by using available information about the approximated function, e.g. convexity, monotonicity, smoothness, etc. The energy functional discussed in this paper seems to be well suited to situations where the approximated function is known to have low energy. In our numerical examples we found that this is not a good criterion for improving the quality of fit to general functions and in many cases interpolants having higher energy provide better approximation than those provided by interpolants with lower energy.

References

[1] P. Alfeld, Derivative generation from multivariate scattered data by functional minimization, Comp. Aided Gem. Des. 2 (1985) 281–296.

[2] R.E. Barnhill, Representation and approximation of surfaces, in: *Mathematical Software III*, ed. J.R. Rice (Academic Press, New York, 1977) pp. 68–119.

[3] G. Baszenski and L.L. Schumaker, Use of simulated annealing to construct triangular facet surfaces, in: *Curves and Surfaces*, eds. P.J. Laurent, A. Le Méhauté and L.L. Schumaker (Academic Press, 1991) pp. 27–32.

[4] A.K. Cline and R.J. Renka, A triangle-based C^1 interpolation method, Rocky Mt. J. Math. 14 (1984) 223–237.

[5] N. Dyn, D. Levin and S. Rippa, Data dependent triangulations for piecewise linear interpolation, IMA J. Numer. Anal. 10 (1990) 137–154.

[6] N. Dyn, D. Levin and S. Rippa, Algorithms for the construction of data dependent triangulations, *Algorithms for Approximation II*, eds. J.C. Mason and M.G. Cox (Chapman and Hall, London, 1990) pp. 185–192.

[7] N. Dyn, I. Goren and S. Rippa, Transforming triangulations in polygonal domains, Report DAMTP NA/13 (1991).

[8] G. Farin, Triangular Bernstein-Bézier patches, Comp. Aided Geom. Des. 3 (1986) 83–127.

[9] R. Franke, Scattered data interpolation: tests of some methods, Math. Comp. 38 (1982) 181–200.

[10] C.L. Lawson, Transforming triangulations, Discr. Math. 3 (1972) 365–372.
[11] C.L. Lawson, Software for C^1 interpolation, in: *Mathematical Software III*, ed. J.R. Rice (Academic Press, New York, 1977) pp. 161–194.
[12] G.M. Nielson and R. Franke, Surface construction based upon triangulations, in: *Surfaces in Computer Aided Design*, eds. R.E. Barnhill and W. Boehm (North-Holland, Amsterdam, 1983) pp. 163–177.
[13] M.J.D. Powell and M.A. Sabin, Piecewise quadratic approximations on triangles, ACM Trans. Math. Software 3 (1977) 316–325.
[14] E. Quak and L.L. Schumaker, Cubic spline fitting using data dependent triangulations, Comp. Aided Geom. Des. 7 (1990) 293–301.
[15] S. Rippa, Long, thin triangles can be good for linear interpolation, SIAM J. Numer. Anal. 29 (1992) 257–270.
[16] L.L. Schumaker, Fitting surfaces to scattered data, in: *Approximation Theory II*, eds. G.G. Lorents, C.K. Chui and L.L. Schumaker (Academic Press, 1976) pp. 203–268.
[17] L.L. Schumaker, Numerical aspects of spaces of piecewise polynomials on triangulations, in: *Algorithm for Approximation*, eds. J.C. Mason and M.G. Cox (Clarendon Press, Oxford, 1987) pp. 373–406.
[18] S.E. Stead, Estimation of gradients from scattered data, Rocky Mt. J. Math. 14 (1984) 265–279.