Computing 60, 29–53 (1998)

Computing

On Computing Derivatives for C^1 Interpolating Schemes: an Optimization

S. De Marchi, Udine

Abstract

The application of Powell–Sabin's or Clough–Tocher's schemes to scattered data problems, as known requires the knowledge of the partial derivatives of first order at the vertices of an underlying triangulation. We study a *local method* for generating partial derivatives based on the minimization of the energy functional on the star of triangles sharing a node that we called a *cell*. The functional is associated to some piecewise polynomial function interpolating the points. The proposed method combines the *global Method II* by Renka and Cline (cf. [16, pp. 230–231]) with the variational approach suggested by Alfeld (cf. [2]) with care to efficiency in the computations. The locality together with some implementation strategies produces a method well suited for the treatment of a big amount of data. An improvement of the estimates is also proposed.

AMS Subject Classifications: 65D05, 65D10, 65D15.

Key words: Interpolation, scattered data fitting, functional minimization.

1. Introduction

The problem of fitting scattered data with smooth surfaces has been solved by different methods (see for example [3, 13, 14, 16] and the references therein). The majority of the methods makes use of polynomials, usually in Bernstein–Bézier form. It is well-known that in order to obtain a C^1 surface by a global interpolating polynomial, its degree has to be at least five [20]. As in the univariate case, in order to use polynomials of lower degree, methods that make use of piecewise interpolating polynomials were developed. Many of these methods are based on various splittings of the underlying triangulation. Two well suited splittings are: the cubic *Clough–Tocher* (CT) and the quadratic *Powell–Sabin* (PS). Those splittings require the further knowledge of the partial derivatives of first order at the data points. Hence, if an interpolation problem has to be solved by those methods, derivatives data must be computed from the existing positional data (cf., i.e. [1, 11, 15, 16, 19]). A survey on the most popular techniques used to compute estimates of the derivatives is the paper by L. L. Schumaker [18].

In this paper we present a method for estimating first and/or second order partial derivatives by minimizing the *energy functional* associated to three piecewise interpolating polynomials: the Q_{18} [3], the Clough–Tocher [5] and the Powell–Sabin [14]. The method works *independently* of the scheme that will be used for the construction of the interpolating function. The minimum of the functional is taken on subsets of the triangulation that we shall call *cells*. A cell is formed by the set of triangles that share a common point of triangulation, that we call its 'center'. A cell formed by N nodes will be termed a *N-cell*. Cells can be *closed*, namely when the chosen point is internal to the triangulation, otherwise they are *open*. The number of cells is exactly the number of the data points. The method is *local* and allows to determine all the derivatives on the whole triangulation by computing the derivatives on all the cells of the triangulation. Our method simply extends an idea suggested in [16] but not used there due to its high computational effort. A similar approach based on the idea of the *minimum norm networks* was suggested by Nielson in [13]. There, the gradients were estimated by considering the problem of finding the interpolating curve network which minimizes the energy functional along the arc length on the curve made by the line segments with endpoint the vertices of a triangle.

Thus, a triangle is considered at most three times because it belongs to three different cells. Therefore, we paid attention to the possibility of reusing the quantities already computed. In fact, as we shall discuss later, our method in its *actual form* is computationally more efficient than the global method of Alfeld (cf. [2]) even though it still is a little less precise (see Section 3). The linear system that one has to solve at each step is of small dimension, depending on the dimension of the cell. Since we want to estimate the partial derivatives of first and second order, if the ith cell is composed of n_i nodes, the dimension of the system will be $sn_i \times sn_i$, s being 2 or 5. Since the best way to triangulate a domain is by equilateral triangles, the *optimal situation* is when $n_i \leq 7$. This is also the upper bound for the average number of vertices in a cell. The latter consideration suggests that the method is particularly well suited for scattered data problems of large dimension.

So far, many attempts have been made to solve the problem of estimating derivatives for scattered data problem by local methods. We simply quote one of the most recent papers (cf. [11]) in which the authors compute the estimates of the gradients by a convex combination of the derivatives on the interpolating planes on the cell corresponding to a given node. This method simply generalizes the one-dimensional case of finding the derivatives of an interpolating quadratic polynomial as a convex combination of the gradients of a set of points. Our approach is more general since it applies to (many) different polynomial patches and allows to estimate derivatives of higher order.

The paper is organized as follows. In Section 2 after a small set of notations, we present the method and some of its properties. Details on the implementation are discussed in the Subsection 2.4. In Section 3 we present some applications of the method and some details of the present implementation. We shall also discuss the strategy followed in order to improve its efficiency. Finally, in the same section, we present the comparison with two known methods.

2. The Method

2.1. Preliminaries

Let $\mathscr{P} = \{P_1, \ldots, P_M\} \subset \mathbf{R}^2$ be the set of the projections of the data points on a suitable reference plane. Let $\tau = \bigcup_{i=1}^{t} T_i$ be an associated triangulation of $\mathscr{P}$, T_i being the ith triangle, provided as usual, that the intersection of two triangles is either empty, a vertex or a common edge. We do not require that τ has *convex* polygonal boundary and moreover it may have some constraints (for an example of constraint region we refer to [6]).

We start by generating the *Delaunay* triangulation of $\mathscr{P}$. As known, this kind of triangulation has the optimal property, that is it avoids long thin triangles since it is based on the *max-min angle criterion* for the connections (cf. [17, 18]).

Let T be a general triangle and $\mathbf{V}_{T,1}$, $\mathbf{V}_{T,2}$, $\mathbf{V}_{T,3}$ its vertices ordered as in the triangulation procedure. Let us assume that the interpolation scheme requires the knowledge of the derivatives up to the order R, at each vertex. For functions of 2 independent variables the partial derivatives of order R are $R(R+3)/2$. Thus, the scheme will globally need the knowledge of $S = 3s$ derivatives, where $s = 3R(R+3)/2$, that is s per vertex. For example, for the element Q_{18}, $S = 15$, while for Clough–Tocher and Powell–Sabin elements $S = 6$.

The *Gaussian curvature* K and the *average curvature* H of a surface or more generally of a variety, satisfy the following equation (cf. [12, pp. 188–190]):

$$k^2 - 2Hk + K = 0. \tag{2.1}$$

The solutions $k_{1,2} = H \pm (H^2 - K)^{1/2}$ are such that $k_1^2 + k_2^2 = 2(2H^2 - K)$. In the case of scalar functions $f(x, y)$ they reduce to the expressions

$$K = \frac{f_{xx} f_{yy} - f_{xy}^2}{\left(1 + f_x^2 + f_y^2\right)^2}, \qquad H = \frac{\left(1 + f_x^2\right) f_{yy} - 2 f_x f_y f_{xy} + \left(1 + f_y^2\right) f_{xx}}{2\left(1 + f_x^2 + f_y^2\right)^{3/2}}. \tag{2.2}$$

From (2.2) we get:

$$\begin{aligned} k_1^2 + k_2^2 = \frac{1}{\left(1 + f_x^2 + f_y^2\right)^3} \Big\{ & f_{xx}^2\left(1 + f_y^2\right)^2 + f_{yy}^2\left(1 + f_x^2\right)^2 \\ & + 2 f_{xy}^2\left(1 + f_x^2 + f_y^2 + 2 f_x^2 f_y^2\right) + 2 f_x^2 f_y^2 f_{xx} f_{yy} \\ & - 4 f_x f_y f_{xy}\left[f_{xx}\left(1 + f_y^2\right) + f_{yy}\left(1 + f_x^2\right)\right]\Big\}. \end{aligned} \tag{2.3}$$

Let us assume that f_x and f_y are nearly zero. This hypothesis seems to be reasonable when one is looking for a smooth interpolating surface having small variations on its curvature. Then, f_x^2, f_y^2 and $f_x f_y$ are smaller and can be taken to be zero. This implies that $k_1^2 + k_2^2 = f_{xx}^2 + 2 f_{xy}^2 + f_{yy}^2$ holds for each point of the domain D, D being a subset of the triangulation τ.

2.2. The Functional

Let Q be a piecewise interpolating polynomial in BB-form on the whole triangulation and Q_T being its restriction to the triangle T. For each triangle T, let $\mathbf{v}_T = \{v_{T,i}\}_{i=0}^{S-1}$ be the array whose elements are the derivatives

$$\frac{\partial Q_T}{\partial x}(\mathbf{V}_{T,1}),\ \frac{\partial Q_T}{\partial y}(\mathbf{V}_{T,1}),\ \frac{\partial^2 Q_T}{\partial x^2}(\mathbf{V}_{T,1}),\ \ldots,$$

$$\frac{\partial^2 Q_T}{\partial y^2}(\mathbf{V}_{T,1}),\ \ldots,\ \frac{\partial^R Q_T}{\partial x^R}(\mathbf{V}_{T,1}),\ \ldots,\ \frac{\partial^R Q_T}{\partial y^R}(\mathbf{V}_{T,1});$$

$$\frac{\partial Q_T}{\partial x}(\mathbf{V}_{T,2}),\ \frac{\partial Q_T}{\partial y}(\mathbf{V}_{T,2}),\ \frac{\partial^2 Q_T}{\partial x^2}(\mathbf{V}_{T,2}),\ \ldots,$$

$$\frac{\partial^2 Q_T}{\partial y^2}(\mathbf{V}_{T,2}),\ \ldots,\ \frac{\partial^R Q_T}{\partial x^R}(\mathbf{V}_{T,2}),\ \ldots,\ \frac{\partial^R Q_T}{\partial y^R}(\mathbf{V}_{T,2});$$

$$\frac{\partial Q_T}{\partial x}(\mathbf{V}_{T,3}),\ \frac{\partial Q_T}{\partial y}(\mathbf{V}_{T,3}),\ \frac{\partial^2 Q_T}{\partial x^2}(\mathbf{V}_{T,3}),\ \ldots,$$

$$\frac{\partial^2 Q_T}{\partial y^2}(\mathbf{V}_{T,3}),\ \ldots,\ \frac{\partial^R Q_T}{\partial x^R}(\mathbf{V}_{T,3}),\ \ldots,\ \frac{\partial^R Q_T}{\partial y^R}(\mathbf{V}_{T,3}). \tag{2.4}$$

Then Q_T can be expressed as a function of these S parameters. Let N be the set of data points in D. We may assume that for each T there exists a function $l_T : \{0, \ldots, S-1\} \to \{0, \ldots, sN-1\}$, called the *conversion function* (see Section 2.5), mapping the index $i \in \{0, \ldots, S-1\}$, corresponding to an element of the vector $\mathbf{v}_T$, into the index $l_T(i) \in \{0, \ldots, sN-1\}$ corresponding to the same element now seen as an element of the vector $\mathbf{v} = \{v_j\}_{j=0}^{sN-1}$. Obviously, when D reduces to a single triangle, then l_T will be the identity. Our aim is to *minimize* the quadratic functional

$$\phi(\mathbf{v}) = \sum_{T \in D} \int_T \left(Q_{T,xx}^2 + 2Q_{T,xy}^2 + Q_{T,yy}^2\right) dx\, dy, \tag{2.5}$$

that is, finding the vector $\mathbf{v}$ that minimizes the square of the Frobenius norm of the Hessian of Q_T. We note that this functional is indeed a special case of a more general one, as discussed in [2, Eq. (2.2), p. 283].

In order to find that minimum first we give an explicit expression of the functional ϕ on the generic triangle T. We start by expressing Q_T in a generalized Hermite in terms of the function values and the derivatives up to order R (along the axis x and y) at the vertices $V_{T,i}$, $i = 1, 2, 3$. We refer to these parameters as $Q_{T,i}$, $Q_{T,i,x}$, $Q_{T,i,y}$, $Q_{T,i,xx}$, $Q_{T,i,xy}$, $Q_{T,i,yy}$, $\ldots$, $i = 1, 2, 3$. This means that we have to determine suitable functions $f_{T,i}^0(x, y)$, $f_{T,i,x}^1(x, y)$, $f_{T,i,y}^1(x, y)$, $f_{T,i,xx}^2(x, y)$, $f_{T,i,xy}^2(x, y)$, $f_{T,i,yy}^2(x, y), \ldots$ so that $Q_T(x, y)$ can be written as:

$$Q_T = \sum_{i=1}^{3} \big(Q_{T,i} f^0_{T,i} + Q_{T,i,x} f^1_{T,i,x} + Q_{T,i,y} f^1_{T,i,y} + Q_{T,i,xx} f^2_{T,i,xx} + Q_{T,i,xy} f^2_{T,i,xy} + Q_{T,i,yy} f^2_{T,i,yy} + \dots\big). \tag{2.6}$$

In the case of the element Q_{18} this formula can be easily obtained from [3, Eq. (16)] by developing in Cartesian coordinates and expressing the resulting formula as a function of the partial derivatives vector **v** instead of the directional derivatives.

By means of the vector **v** and suitable functions $s_j(x,y)$, $j \in \{0,\dots,sN-1\}$, the formula (2.6) can be rewritten as follows:

$$Q_T = \sum_{i=1}^{3} Q_{T,i} f^0_{T,i} + \sum_{i=0}^{S-1} v_{l_T(i)} s_{l_T(i)}. \tag{2.7}$$

In this form we have separated function values from the unknown parameters v_j.

Let us consider the bilinear form:

$$a_T(F,G) = \int_T \big(F_{T,xx} G_{T,xx} + 2F_{T,xy} G_{T,xy} + F_{T,yy} G_{T,yy}\big)\, dx\, dy,$$

where F and G are two functions of sN parameters, and F_T and G_T are their restrictions to the triangle T. It is worth noting that the functions F_T and G_T depend on S unknown parameters, i.e. the ones listed in (2.4). Moreover, let $a(F,G) = \sum_{T\in D} a_T(F,G)$ be a bilinear symmetric form.

Proposition 2.1. *The functional $\phi(\mathbf{v})$ is the bilinear-quadratic form a evaluated in Q.*

By some mathematics we can write:

$$\begin{aligned}
\phi(\mathbf{v}) &= \sum_{T\in D} a_T(Q,Q) \\
&= \sum_{T\in D} a_T\left(\sum_{i=1}^{3} Q_{T,i} f^0_{T,i} + \sum_{i=0}^{S-1} v_{l_T(i)} s_{l_T(i)},\ \sum_{j=1}^{3} Q_{T,j} f^0_{T,j} + \sum_{j=0}^{S-1} v_{l_T(j)} s_{l_T(j)}\right) \\
&= \sum_{T\in D} \sum_{i,j=0}^{S-1} v_{l_T(i)} v_{l_T(j)} a_T\big(s_{l_T(i)}, s_{l_T(j)}\big) \\
&\quad + 2 \sum_{T\in D} \sum_{i=0}^{S-1} v_{l_T(i)} a_T\left(s_{l_T(i)},\ \sum_{j=1}^{3} Q_{T,j} f^0_{T,j}\right) \\
&\quad + \sum_{T\in D} a_T\left(\sum_{i=1}^{3} Q_{T,i} f^0_{T,i},\ \sum_{j=1}^{3} Q_{T,j} f^0_{T,j}\right) \\
&= \frac{1}{2} \sum_{i,j=0}^{sN-1} v_i v_j a_{ij} + \sum_{i=0}^{S-1} 2 \sum_{T\in D} v_{l_T(i)} a_T\left(s_{l_T(i)},\ \sum_{j=1}^{3} Q_{T,j} f^0_{T,j}\right)
\end{aligned}$$

$$+ \sum_{i,j=1}^{3} \sum_{T \in D} Q_{T,i} Q_{T,j} a_T\left(f_{T,i}^0, f_{T,j}^0\right)$$

$$= \frac{1}{2} \sum_{i,j=0}^{sN-1} v_i v_j a_{ij} + \sum_{i=0}^{sN-1} v_i g_i + \sum_{i,j=1}^{3} \sum_{T \in D} Q_{T,i} Q_{T,j} a_T\left(f_{T,i}^0, f_{T,j}^0\right).$$

Thus,

$$\phi(\mathbf{v}) = \tfrac{1}{2}\mathbf{v}^T\mathbf{A}\mathbf{v} + \mathbf{g}^T\mathbf{v} + c. \tag{2.8}$$

The matrix $\mathbf{A} = (a_{ij})_{i,j=0}^{sN-1}$, the array $\mathbf{g} = (g_i)_{i=0}^{sN-1}$ and the scalar c have the following expressions:

$$a_{ij} = \begin{cases} 2\sum_T a_T(s_i, s_j) & \forall T \in D \text{ such that } \exists\, i', j' \in \{0, \dots, S-1\} \\ & \text{with } i = l_T(i'),\, j = l_T(j') \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

$$g_i = 2\sum_T \sum_{j=1}^{3} a_T\left(s_i, Q_{T,j} f_{T,j}^0\right) \quad \forall T \in D \text{ such that } \exists i' \in \{0, \dots, S-1\}$$
$$\text{with } i = l_T(i') \tag{2.10}$$

$$c = \sum_{i,j=1}^{3} \sum_{T \in D} D_{T,i} Q_{T,j} a_T\left(f_{T,i}^0, f_{T,j}^0\right). \tag{2.11}$$

Proposition 2.2.

(i) *The functional $\phi(\mathbf{v})$ is a semidefinite positive quadratic form.*
(ii) *The matrix $\mathbf{A}$ is symmetric positive definite.*

Proof: The statement (i) is obvious, since the functional $\phi(\mathbf{v})$ is the sum of positive integrals.

About (ii), the symmetry of the matrix A follows by the definition of its coefficients. Concerning the positiveness we must prove that for each $\mathbf{v} \in \mathbf{R}^{sN}$ $\mathbf{v}^T\mathbf{A}\mathbf{v} \geq 0$ with the equality if and only if $\mathbf{v} = \mathbf{0}$. Let us consider the interpolation problem in which $Q_{T,i} = 0$, $i = 1, 2, 3$, $\forall T \in D$. This implies $c = 0$ and $\mathbf{g} = \mathbf{0}$. Then $\phi(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{A}\mathbf{v}$. Since $\phi(\mathbf{v}) = a(Q, Q) = 0$, it follows that $Q_{T,xx}$, $Q_{T,xy}$, $Q_{T,yy}$ and all the derivatives of higher order vanish on all triangles. This implies that on each triangle T, the polynomial $Q_T(x, y)$ is a linear function of x and y. On the other hand, $Q_{T,i} = 0$, $\forall T \in D$, $i = 1, 2, 3$, then $Q_T \equiv 0$, $\forall T$, that is $\mathbf{v} = \mathbf{0}$. ■

2.3. The Minimization Procedure

In BB-form the restriction Q_T to a generic triangle T of a polynomial patch Q of degree s can be written as follows:

$$Q_T(x, y) = \sum_{|\mathbf{r}|=s} d_{\mathbf{r}} B_{\mathbf{r}}^s(\mathbf{b}(x, y)), \tag{2.12}$$

where the control points $d_{\mathbf{r}}$ are linear functions of the unknown coefficients v_i, $i = 0, \ldots, S-1$, and $\mathbf{b}(x, y)$ are the barycentric coordinates of the point (x, y) with respect to the triangle T of vertices $\mathbf{V}_{T,1}$, $\mathbf{V}_{T,2}$ and $\mathbf{V}_{T,3}$.

Hence, the partial derivatives of Q_T up to order 2 along the axes x and y become:

$$Q_{T,x} = -\frac{s}{2A(T)} \sum_{|\mathbf{r}|=s-1} \left(\sum_{i=1}^{3} d_{\mathbf{r}+\mathbf{e}_i} e_i^y \right) B_{\mathbf{r}}^{(s-1)}(\mathbf{b}), \tag{2.13}$$

$$Q_{T,y} = \frac{s}{2A(T)} \sum_{|\mathbf{r}|=s-1} \left(\sum_{i=1}^{3} d_{\mathbf{r}+\mathbf{e}_i} e_i^x \right) B_{\mathbf{r}}^{(s-1)}(\mathbf{b}), \tag{2.14}$$

$$Q_{T,xx} = \frac{s(s-1)}{(s-1)A^2(T)} \sum_{|\mathbf{r}|=s-2} \left(\sum_{i,j=1}^{3} d_{\mathbf{r}+\mathbf{e}_i+\mathbf{e}_j} e_i^y e_j^y \right) B_{\mathbf{r}}^{(s-2)}(\mathbf{b}), \tag{2.15}$$

$$Q_{T,xy} = -\frac{s(s-1)}{(s-1)A^2(T)} \sum_{|\mathbf{r}|=s-2} \left(\sum_{i,j=1}^{3} d_{\mathbf{r}+\mathbf{e}_i+\mathbf{e}_j} e_i^y e_j^x \right) B_{\mathbf{r}}^{(s-2)}(\mathbf{b}), \tag{2.16}$$

$$Q_{T,yy} = \frac{s(s-1)}{(s-1)A^2(T)} \sum_{|\mathbf{r}|=s-2} \left(\sum_{i,j=1}^{3} d_{\mathbf{r}+\mathbf{e}_i+\mathbf{e}_j} e_i^x e_j^x \right) B_{\mathbf{r}}^{(s-2)}(\mathbf{b}). \tag{2.17}$$

Here $2A(T)$ denotes the area of the triangle T and $\mathbf{e}_i$ the canonical vectors of $\mathbf{R}^3$. Explicit expressions of the control points for the elements Q_{18}, Clough–Tocher and Powell–Sabin are listed for example in [10, pp. 55–57].

Making use of the matrix representation of $\phi(\mathbf{v})$ and the symmetry of the matrix $\mathbf{A}$, we obtain:

$$\frac{\partial^2 \phi}{\partial v_i \partial v_j} = \frac{a_{ij} + a_{ji}}{2} = a_{ij}. \tag{2.18}$$

Proposition 2.3. *The coefficients a_{ij} in (2.18) are given by the following relations*

$$a_{ij} = \begin{cases} 2 \sum_{T \in D} \int_T \left(\frac{\partial Q_{T,xx}}{\partial v_i} \frac{\partial Q_{T,xx}}{\partial v_j} + 2 \frac{\partial Q_{T,xy}}{\partial v_i} \frac{\partial Q_{T,xy}}{\partial v_j} + \frac{\partial Q_{T,yy}}{\partial v_i} \frac{\partial Q_{T,yy}}{\partial v_j} \right) dx\,dy \\ 0 \end{cases} \tag{2.19}$$

where $a_{ij} \neq 0$ if there exist indices $i', j' \in \{0, \ldots, S-1\}$ such that $i = l_T(i')$, $j = l_T(j')$.

Proof: We begin the proof by noting that for each triangle T the control points $d_{\mathbf{r}}$ are linear functions of $\mathbf{v}_T$ which depend on $\mathbf{v}$. From the formula of directional derivatives for triangular BB-polynomials (cf. [8, 9]), we notice that Q_T, $Q_{T,xx}$, $Q_{T,xy}$ and $Q_{T,yy}$ are linear functions of the control points $d_{\mathbf{r}}$. Thus, the following relations hold:

$$\frac{\partial^2 Q_{T,xx}}{\partial x^2}=0, \qquad \frac{\partial^2 Q_{T,xy}}{\partial x \partial y}=0, \qquad \frac{\partial^2 Q_{T,yy}}{\partial y^2}=0.$$

Using the identity

$$\frac{\partial^2 F^2}{\partial x \partial y}=\frac{\partial}{\partial x}\left(2F\frac{\partial F}{\partial y}\right)=2\frac{\partial F}{\partial x}\frac{\partial F}{\partial y}+2F\frac{\partial^2 F}{\partial x \partial y},$$

and the '*Lebesgue theorem on dominated convergence*', we may differentiate under the integral symbol since for any triangle T one can find a function $z(x,y)\in L^1(T;\mathbf{R})$ such that

$$\left|\frac{\partial^2 Q^2_{T,xx}}{\partial v_i \partial v_j}\right|<z(x,y),$$

holds for each point in T (similarly for $Q_{T,xy}$ and $Q_{T,yy}$). A simple choice could be

$$z(x,y)=\max_{P\in T}\left|\frac{\partial^2 Q^2_{T,xx}}{\partial v_i \partial v_j}(P)\right|.$$

The maximum exists because of the continuity of $\dfrac{\partial^2 Q^2_{T,xx}}{\partial v_i \partial v_j}$ on the compact T. Then, the conclusion comes easy. ■

For the constant term c in (2.8) we get

$$c=\phi(\mathbf{0})=\sum_{T\in D}\int_T\left(Q^2_{T,xx}(0)+2Q^2_{T,xy}(0)+Q^2_{T,yy}(0)\right)dx\,dy,$$

where $\mathbf{0}$ is the zero of $\mathbf{R}^{sN}$. Moreover,

$$\phi(\mathbf{e}_i)=\sum_{T\in D}\int_T\left(Q^2_{xx}(\mathbf{e}_i)+2Q^2_{xy}(\mathbf{e}_i)+Q^2_{yy}(\mathbf{e}_i)\right)dx\,dy,$$

where $\mathbf{e}_i$ are the canonical vectors of $\mathbf{R}^{sN}$. Therefore, we have

$$g_i=\phi(\mathbf{e}_i)-\tfrac{1}{2}a_{ii}-c \quad \forall i\in\{0,\dots,sN-1\}.$$

Finally, $\dfrac{d\phi}{d\mathbf{v}}=\mathbf{A}\mathbf{v}+\mathbf{g}$, by the symmetry and definite positiveness of the matrix $\mathbf{A}$, the solution v^* that minimize $\phi(\mathbf{v})$ is the unique solution of the system

$$\mathbf{A}\mathbf{v}=-\mathbf{g}. \tag{2.20}$$

2.4. Outline of the Algorithm

1. The integrals on a triangle T of a polynomial

$$\mathbf{p}=\sum_{r+s+t=n}c_{r,s,t}B^n_{r,s,t}(b_1,b_2,b_3)$$

of degree n in BB-form, are computed by the formula (cf. [2]):

$$\int_T \mathbf{p}\,dx\,dy = 2A(T)\int_0^1\int_0^{1-b_1}\mathbf{p}\,db_2\,db_1 = \frac{2A(T)}{(n+1)(n+2)}\sum_{r+s+t=n} c_{r,s,t}, \quad (2.21)$$

since

$$\int_0^1\int_0^{1-b_1} B^n_{r,s,t}(b_1,b_2,b_3)\,db_2\,db_1 = \frac{1}{(n+1)(n+2)} \quad \forall r,s,t,b,\ r+s+t=n. \quad (2.22)$$

2. The matrix A can be quickly computed by the following steps.
   - Initialize A to the null matrix.
   - On each triangle T compute

$$a_{T,i'j'} = 2\int_T\left(\frac{\partial Q_{T,xx}}{\partial v_{i'}}\frac{\partial Q_{T,xx}}{\partial v_{j'}} + 2\frac{\partial Q_{T,xy}}{\partial v_{i'}}\frac{\partial Q_{T,xy}}{\partial v_{j'}} + \frac{\partial Q_{T,yy}}{\partial v_{i'}}\frac{\partial Q_{T,yy}}{\partial v_{j'}}\right)dx\,dy, \quad (2.23)$$

   with $i' \in \{0,\ldots,S-1\}$, $j' \in \{0,\ldots,i'\}$. Depending on the interpolant chosen for the minimization, making use of Eqs. (2.15), (2.16), (2.17) and the linearity of the derivation with respect to $v_{i'}$ we may compute the derivatives in (2.23). In order to do this, we need to know the derivatives with respect to the unknown parameters of the control points and the product of two triangular polynomial patches (in BB-form). The latter can be done by the following result (cf. [10, p. 61]):

Lemma 2.1. *Let*

$$P_1(\mathbf{u}) = \sum_{|\mathbf{r}|=m} b_{\mathbf{r}} B^m_{\mathbf{r}}(\mathbf{u}), \qquad P_2(\mathbf{u}) = \sum_{|\mathbf{r}|=n} c_{\mathbf{r}} B^n_{\mathbf{r}}(\mathbf{u}),$$

be two polynomials in BB-form of degree m and n, respectively. Then

$$P_1(\mathbf{u})P_2(\mathbf{u}) = \sum_{|\mathbf{r}|=m+n} a_{\mathbf{r}} B^{m+n}_{\mathbf{r}}(\mathbf{u}),$$

where

$$a_{\mathbf{r}} = a_{r,s,t} = \sum_{(r,s,t)\in\tilde{S}} b_{\mathbf{r}_1} c_{\mathbf{r}_2} \frac{\binom{m}{r_1,s_1,t_1}\binom{n}{r_2,s_2,t_2}}{\binom{m+n}{r_1+r_2,s_1+s_2,t_1+t_2}},$$

$$\tilde{S} = \left\{\begin{array}{c} \mathbf{r}=(r,s,t),\ 0\le r,s,t\le m+n;\ r=r_1+r_2,\ s=s_1+s_2,\ t=t_1+t_2,\\ 0\le r_1,s_1,t_1\le m,\ 0\le r_2,s_2,t_2\le n,\\ r_1+s_1+t_1=m,\ r_2+s_2+t_2=n, \end{array}\right\}$$

and

$$\binom{p}{i,j,k} = \frac{p!}{i!j!k!}.$$

- Determine $i = l_T(i')$, $j = l_T(j')$ and set $a_{ij} := a_{ij} + a_{T,i'j'}$.

3. Similarly compute $\phi(\mathbf{e}_i)$, $i \in \{0, \ldots, sN-1\}$ and $\phi(\mathbf{0})$. Now let

$$\phi_T(\mathbf{v}) = \int_T \left(Q^2_{T,xx}(\mathbf{v}) + 2Q^2_{T,xy}(\mathbf{v}) + Q^2_{T,yy}(\mathbf{v})\right) dx\, dy \quad \forall \mathbf{v} \in \mathbf{R}^S.$$

To compute $\phi(\mathbf{0})$, perform the following steps:

- Set $c_0 := 0$.
- Compute $c_0 := c_0 + \phi_T(0)$, $\forall T \in D$, where 0 stands for the zero of $\mathbf{R}^S$. At the end, $c_0 = \phi(\mathbf{0})$.

Finally the $\phi(\mathbf{e}_i)$ are computed as follows:

- Set $c_i := 0$ $\forall i \in \{0, \ldots, sN-1\}$.
- Compute $c_i := c_i + \phi_T(\epsilon_{i'})$ $\forall T \in D$ so that $\exists i' \in \{0, \ldots, S-1\}$ with $i = l_T(i')$, and ϵ_j $\forall j \in \{0, \ldots, S-1\}$ are the canonical vectors of $\mathbf{R}^{S-1}$.
- Compute $c_i := c_i + \phi_T(0)$ $\forall T \in D$ so that $\nexists i' \in \{0, \ldots, S-1\}$ with $i = l_T(i')$ where 0 is again the zero of $\mathbf{R}^S$. At the end, $c_i = \phi(\mathbf{e}_i)$.

Notice that the control points of the interpolant are functions of the null vector and the canonical vectors of $\mathbf{R}^S$.

2.5. The Conversion Function l_T

The nodes of a triangulation are numbered in a natural way by their insertion order. This index will be termed the *global index*. The nodes of a cell can *locally* be numbered in counterclockwise way (see Figs. 1 and 2) by labeling with zero the *base* node[1] of the cell. In this case, the numbering is unique for open cells while there should be some ambiguity in the choice of the node labeled by 1 for

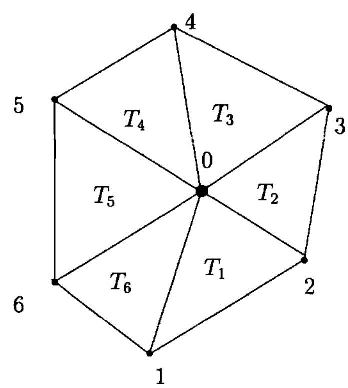

Figure 1. Closed cell with local indices

[1]The *base* node is the one on which the cell is built around.

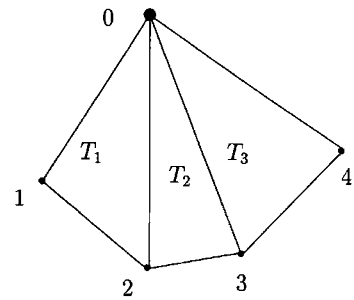

Figure 2. Open cell with local indices

closed cells. This problem can be solved by considering the same order defined by the triangulation algorithm. That is, if two nodes were close in the triangulation they will continue to be in the cell. We are potentially able to compute for each cell C, the function λ_C mapping local indices to global ones.

Let C be a N-cell formed by $N-1$ triangles if it is closed and by $N-2$ triangles if it is open. Moreover, let T_j be the triangle with vertices $(\mathbf{V}_{T_j,1}, \mathbf{V}_{T_j,2}, \mathbf{V}_{T_j,3})$, where $\mathbf{V}_{T_j,1}$ is associated to $\lambda_C(0)$, $\mathbf{V}_{T_j,2}$ is associated to $\lambda_C(1)$ and $\mathbf{V}_{T_j,3}$ to $\lambda_C(j \bmod (N-1)+1)$. Hence, the required conversion function is

$$l_{T_j}(i) = \begin{cases} i \bmod s & \text{if } i \text{ div } s = 0 \\ sj + i \bmod s & \text{if } i \text{ div } s = 1 \\ 5(j \bmod (N-1)+1) + i \bmod s & \text{if } i \text{ div } s = 2 \end{cases} \tag{2.24}$$

that holds for all $i \in [0, \ldots, S-1\}$.

2.6. Optimization

The approach presented above has some difficulties. In fact, for each triangle T of an N-cell we have to calculate the quantities

$$a_{T,i',j'}, \quad \forall i' \in \{0, \ldots, sN-1\}, \quad \forall j' \in \{0, \ldots, i'\} \tag{2.25}$$

$$\phi_T(\epsilon_{i'}), \quad \forall i' \in \{0, \ldots, sN-1\}, \tag{2.26}$$

$$\phi_T(0). \tag{2.27}$$

But a triangle belongs at least to three different cells. Hence, the same triangle has different labeling of the vertices depending on the cell and on the order on which the nodes are considered except, obviously, for the quantity $\phi_T(0)$ that remains unchanged. Therefore, we introduce the *canonical* representation for the generic triangle T.

Definition 2.1. *Let T be a triangle of a triangulation τ. Its canonical representation, $\sigma(T)$, is a function whose image is a triplet of indices (i_1, i_2, i_3) in the global labeling, such that $i_1 < i_2$, $i_1 < i_3$.*

Remark. We adopted the convention to run into the vertices in counterclockwise way. Therefore, independently to the vertex we started with, the canonical representation of a triangle is *unique*. In fact, there is only one configuration for which $i_1 < i_2$, $i_1 < i_3$. Moreover, between the indices i_2 and i_3 there is not any link.

Thus, the proposed optimization must perform the following steps.

1. Compute the functions $a_{T,i',j'}$, $\phi_T(\epsilon_{i'})$ and $\phi_T(0)$ in the global numbering and store them in a file (or in the memory). In the general case each triangle contributes to $(S^2+3S+2)/2$ elements.
2. Rotate the triangle counterclockwise a number of times necessary to reset the canonical representation. If $k(T)$ is this number, easily we get $k(T) \in \{0,1,2\}$. When the original situation has been reached, instead of computing the new fundamentals elements (2.25) and (2.26), we simply take them from their position (in the file or in the memory). This is done by checking that the following relations are true.

$$a_{T,i',j'} = a_{\sigma(T),(i'+k(T)2s)\mathrm{mod}\,S,(j'+k(T)2s)\mathrm{mod}\,S}, \tag{2.28}$$

$$\phi_T(\epsilon_{i'}) = \phi_{\sigma(T)}(\epsilon_{(i'+k(T)2s)\mathrm{mod}\,S}), \tag{2.29}$$

$$\phi_T(0) = \phi_{\sigma(T)}(0), \qquad \forall i', j' \in \{0,\dots,S-1\}. \tag{2.30}$$

The only necessary preliminary work is the knowledge of the progressive order of the canonical representations of all the triangles in order to quickly access to these information.

2.7. Improving the Estimates of the Derivatives

In order to improve the method, there is a simple way to weight the gradients, so that the resulting energy is lower or of the same order of the *optimal* energy estimated by the global method. The method consists in weighting the value of the gradient at the point P_i with respect to the points P_j, $j=1,\dots,k$, that is the vertices of the cell having P_i as its 'center'. In formulae

$$\mathbf{g}_i = \mathbf{g}_{i,i} \cdot (1-q) + \frac{\sum_{j=1}^{k} \frac{\alpha e^{-\beta d_{i,j}^2}}{d_{i,j}^{\gamma}} \mathbf{g}_{i,j}}{\sum_{j=1}^{k} \frac{\alpha e^{-\beta d_{i,j}^2}}{d_{i,j}^{\gamma}}} \cdot q, \tag{2.31}$$

where $q \in [0,1]$ is a parameter ($q=0$ is the local method above presented), $d_{i,j}$ are the distances between the centers of neighboring cells and $\mathbf{g}_{i,j}$ the gradients of the j-th point estimated by considering the i-th cell. The formula (2.31) reads as follows. The gradient at the i-th node is a convex combination of the gradient $\mathbf{g}_{i,i}$ computed by the local method and the weighted sum of the gradients $\mathbf{g}_{i,j}$

evaluated again by the local method corresponding to the vertices of the triangles composing the i-th cell. The weights chosen are related to the distances between P_i and P_j by an exponential function. As shown in [6], this new variant of the local method already reduces the energy to the same order of the global method by choosing q close to 0.5 and weights $d_{i,j}^{-2}$. This is equivalent to a topology with regular cells.

In Table 5 we show the energy for the Clough–Tocher element computed by using equation (2.31) with different values of the parameters. By default we took the values $\alpha = 1$, $\beta = 0$, $\gamma = 1$ which is equivalent to weight with the inverse of the distances $d_{i,j}$. The values showed are almost the best one obtained by using different values of the parameters: obviously the parameter α has no relevance in the final results. Finally in Table 6 we compare the energy associated to the Clough–Tocher element by three different methods, that is the global method by Alfeld, the local scheme by Goodman et al. and our method. We may conclude that our local method is competitive with the global one giving energies of the same order of the global method, that can be considered as optimal.

2.8. Local versus Global Estimation: Comparison of Time Complexity

The global method by Alfeld (cf. [2]) solves a linear system of dimension $Ms \times Ms$, M being the number of nodes in the triangulation.

Our method instead has to solve M systems of lower dimension. In fact, if the ith cell is a n_i-cell, at each step the system will be of dimension $s \cdot n_i \times s \cdot n_i$. Obviously, the most interesting part of the solution vector is the first part formed by the s very first components. This says that in evaluating the time complexity of the (local) method one has to consider the time for solving the linear systems and the time for the construction of the matrices. The construction is made by the optimization procedure described above so that we may assume that the time spent for this construction will be almost the same. Actually, the time for the construction of the systems for the local method are relatively longer due to the repeated use of the *fundamental elements* in different cells and the necessity of determining the right indices in the *canonical representation.* Despite of this, the solver used in both cases is the same, that is the *Cholesky decomposition.* As is known, the time complexity of Cholesky decomposition if $\mathcal{O}(M^3/6)$ for a system of size $M \times M$. We notice that a direct method for the global approach can also be used, as advised in [2, p. 284]. About the time complexity we can state the following result.

Proposition 2.4. *Using Cholesky's factorization, the minimum number M of nodes so that the local method is faster than the global one must satisfy the inequality*

$$m^3 \leq M^2 \tag{2.32}$$

where

$$m = \frac{1}{M}\sum_{i=1}^{M} n_i.$$

Proof: Thanks to the linearity of the $\mathcal{O}$ symbol, we must solve the inequality

$$\sum_{i=1}^{M} \frac{(n_i s)^3}{6} \le \frac{(Ms)^3}{6}.$$

For large M, let

$$m = \frac{1}{N}\sum_{i=1}^{M} n_i$$

be the average number of nodes in a cell. Then, the conclusion becomes trivial. ■

Considering also the time for the backward substitution step, we easily obtain the following lower bounds for M:

$$m^3 \le M^2,$$

$$m^2 \le M.$$

Remark. The relation (2.32) will be the basic one to refer to in order to decide which approach to use for the estimation of the derivatives. We emphasize that even for a small number of points our method seems to be faster. In fact, also in the case of a random distribution of points, for large M, $m \le 7$ (see Table 1). If $m \le 7$, from (2.32) we have $M \ge 18.52$, a value relatively small.

3. Examples and Applications

We have developed an interactive package, called **LABSUP** [7], that allows to compute, visualize, modify and manipulate interpolating surfaces. The graphic representations have been obtained by the OpenGL graphics library. All tests were done on an IBM RISK 6000 with 128 Mbytes of RAM. Starting from the file containing the data points, the package has separate modules that generate the file of the Delaunay triangulation and the file of the derivatives. Moreover, it allows to visualize the interpolating surface by the Q_{18}, the Clough–Tocher or the Powell–Sabin polynomials patches. For more details on the implementation we refer the user to [6, Ch. 2] and [10, Appendix D].

Some words on the symbols used in the annexed tables and pictures. For the tests we have used the exponential *Franke's function* indicated as function f, the *sombrero-like hat function* indicated by h and the *saddle function* s. Their analytical expressions are as follows:

$$f(x,y) = \tfrac{3}{4}e^{-((9x-2)^2+(9y-2)^2)/4} + \tfrac{3}{4}e^{-((9x+1)^2/49)-(9y+1)/10}$$
$$+\tfrac{1}{2}e^{-((9x-7)^2+(9y-3)^2)/4} - \tfrac{1}{5}e^{-(9x-4)^2-(9y-7)^2},$$

$$h(x,y) = \frac{1}{1+x^2+y^2} - e^{-(x^2+y^2)},$$

$$s(x,y) = \frac{1.25+\cos(5.4y)}{6(3x-1)^2+6}.$$

Their domains are $[0,1]\times[0,1]$, $[-3,3]\times[-3,3]$, $[0,1]\times[0,1]$, respectively. We also used a set of 25 random generated 3D points on the square $[0,2]\times[0,2]$ and $z\in[0.5,1]$. In all the figures and tables the subscripts indicate the number of points on which the functions are evaluated. For example, f_{50} denotes Franke's function evaluated on 50 points.

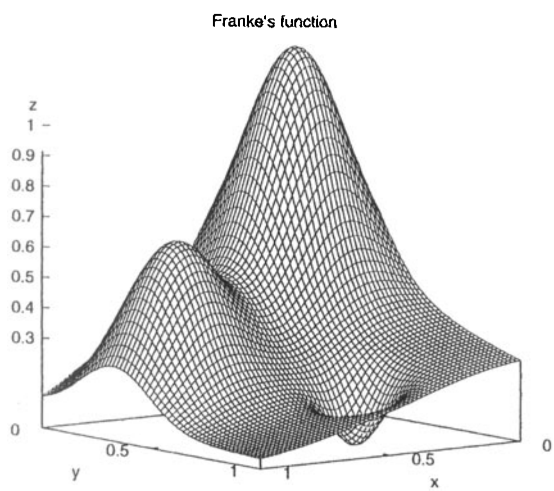

Figure 3. Franke's function f

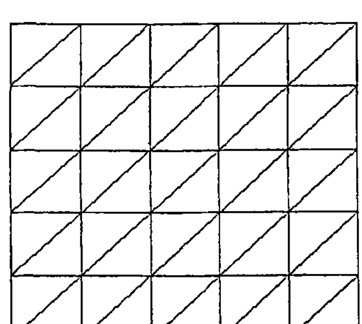

Figure 4. *Type-1* triangulation for f_{36}

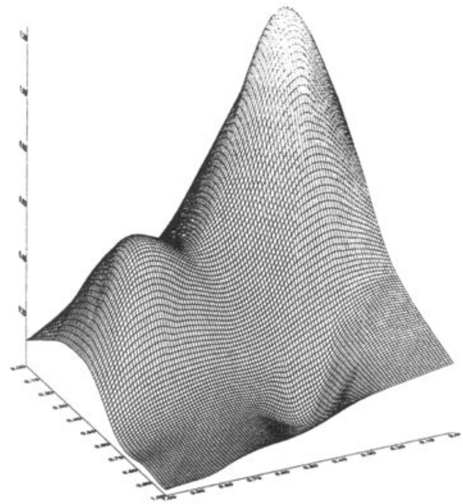

Figure 5. Clough–Tocher interpolant for f_{36} with derivatives D_{ct}^{loc}

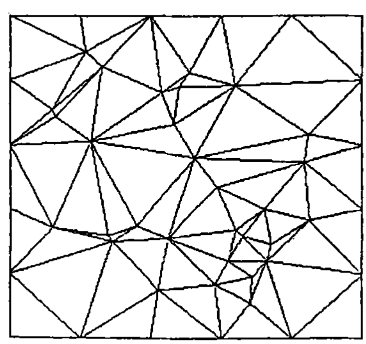

Figure 6. Triangulation for f_{50}

The plots of the surfaces, the corresponding Delaunay triangulations and the interpolating surfaces obtained by our method are shown in Figs. 3–19. In the description of the pictures, the notations are the same ones adopted in Table 1 except for the symbol D_{orig} that indicates the derivatives of the original function (when available).

Table 1 presents the running times (in seconds) for computing the derivatives by both the global method and our local one. The notations read as follows. In the first column we listed the triangulation type followed by the number of data points. The triangulations titled by 'ES*' are the ones used for the functions f and h, while the 'ER*' ones refer to the random generated points. In particular, the triangulation *ES*51 includes the point at which f assumes its maximum, ES36–A is a 'type-1 triangulation' (3-directions mesh), ES36–B is a 'type-2 triangulation' (4-directions mesh), ESA150 is the triangulation used for repre-

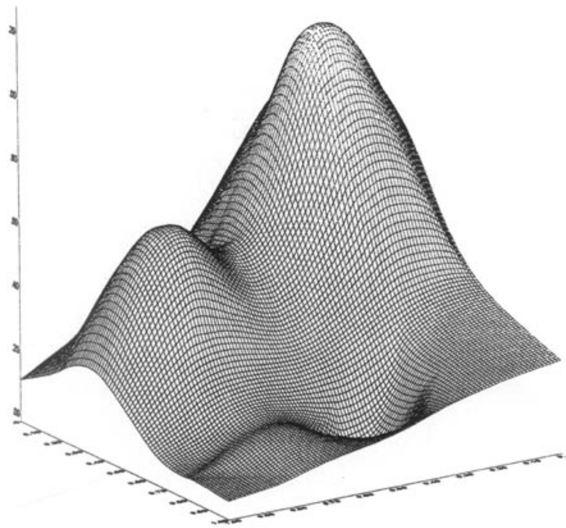

Figure 7. Clough–Tocher interpolant for f_{50} with derivatives D^{loc}_{ct}

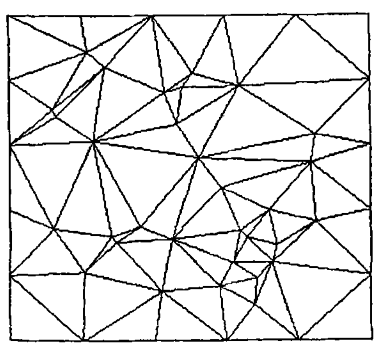

Figure 8. Triangulation for f_{51}

Table 1. Running time (in seconds) for generating the derivatives

Triangulation/Method	D^{loc}_{ct}	D^{loc}_{q18}	D^{loc}_{pws}	D^{gl}_{ct}	D^{gl}_{q18}	D^{gl}_{pws}	max n_i	m
ES16	0.2	0.8	0.3	0.2	0.8	0.3	8	5.625
ES36_A	0.4	2	0.7	0.4	2.6	0.8	7	5.720
ES36_B	0.4	2.1	0.7	0.4	2.6	0.8	7	5.720
ES50	0.7	3.0	1.1	0.7	4.8	1.1	10	6.080
ES51	0.9	4.0	1.5	1	7.8	1.7	9	6.250
ES100_A	1.5	6.8	2.4	2.5	37.3	3.4	11	6.540
ES100	1.6	7.4	2.7	2.6	37.5	3.7	11	6.860
ES150	2.3	10.8	3.7	6.2	188.0	7.5	11	6.690
ER25	0.3	1.5	0.5	0.3	1.5	0.5	8	6.040
ER200	3.3	14.9	5.3	16.9	448.8	18.8	10	6.830
ER2000	43.8	165.0	64.7	**	**	**	14	6.978

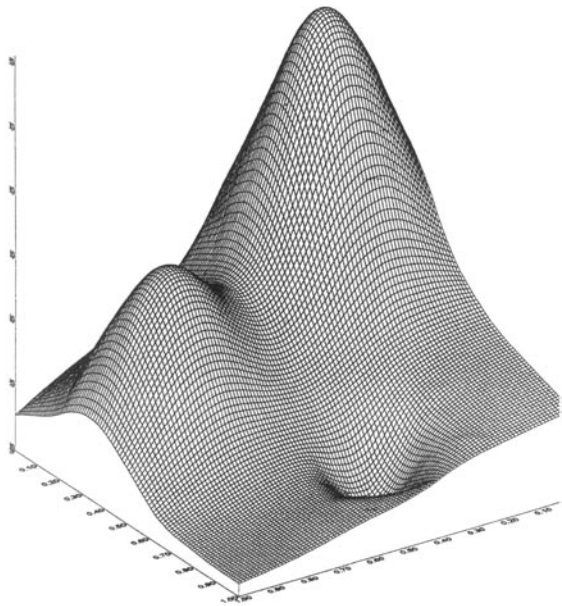

Figure 9. Clough–Tocher interpolant for f_{51} with derivatives D_{ct}^{loc}

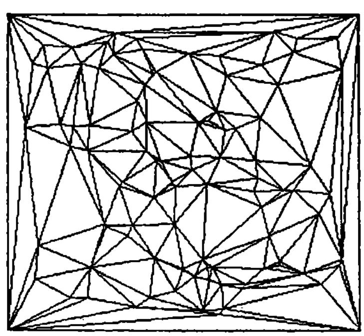

Figure 10. Triangulation for f_{100}

Table 2. The energy $\phi(\mathbf{v})$ for the Clough–Tocher element on different triangulations and different methods of generating derivatives

Triangulation/Method	D_{ct}^{loc}	D_{ct}^{gl}	D_{q18}^{loc}	D_{q18}^{gl}	D_{pws}^{loc}	D_{pws}^{gl}
ES36_A	469.575	221.172	496.079	318.140	471.495	222.988
ES36_B	495.379	219.753	590.215	336.485	509.885	221.292
ES50	1097.329	234.440	879.097	482.874	930.181	240.170
ES100_A	85685.547	196.790	104469.209	296.176	93973.070	199.482
ES100	53492.226	2453.555	50000.941	147674.467	47014.319	5337.791
ES150	62159.855	230.023	24262.619	305.073	64063.157	231.877
EX100	5.544	1.981	6.434	2.930	5.761	1.995
ESA150	337.969	9.533	286.728	11.224	351.711	9.558
ER25	1896.248	420.590	1313.908	2708.996	1713.792	464.216

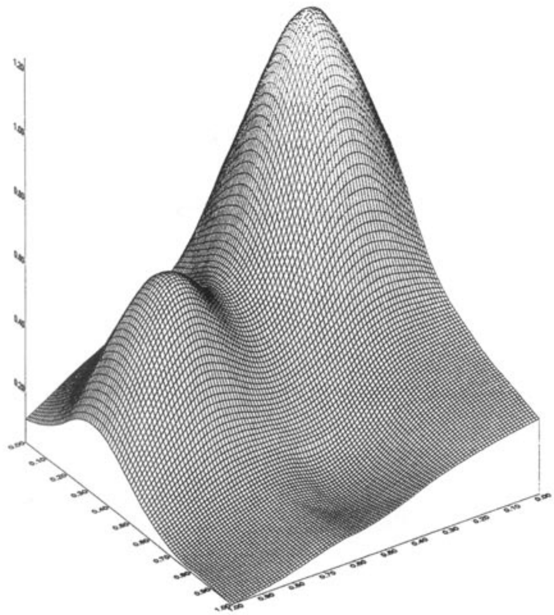

Figure 11. Clough–Tocher interpolant for f_{100} with derivatives $D^{loc}CT$

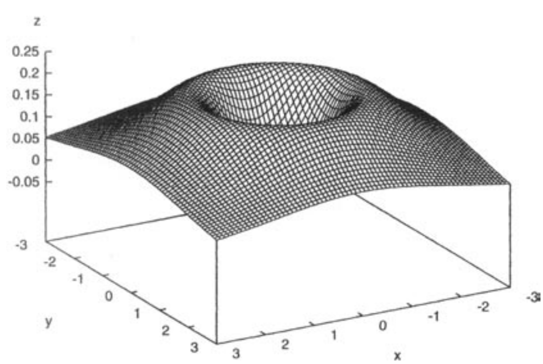

Figure 12. Function h

Table 3. The energy $\phi(\mathbf{v})$ for the Q_{18} element on different triangulations and different methods of generating derivatives

Triangulation/Method	D_{ct}^{loc}	D_{ct}^{gl}	D_{q18}^{loc}	D_{q18}^{gl}	D_{pws}^{loc}	D_{pws}^{gl}
ES36_A	441.424	249.282	275.669	156.416	443.105	255.840
ES36_B	464.473	263.737	312.642	156.673	474.560	269.062
ES50	947.936	326.926	1286.973	162.550	794.993	339.013
ES100_A	82639.411	283.114	74875.010	172.601	90542.759	283.032
ES100	63557.581	6887.502	18683.591	282.977	53989.113	6058.220
ES150	60051.302	348.004	156365.182	198.895	61830.852	355.831
EX100	5.327	2.431	4.637	1.421	5.490	2.448
ESA150	338.268	14.034	924.660	8.700	351.604	14.058
ER25	1738.006	485.068	1094.375	132.730	1611.158	528.639

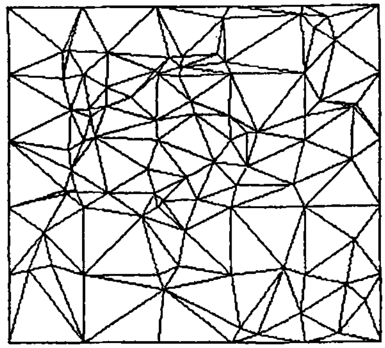

Figure 13. Triangulation for h_{100}

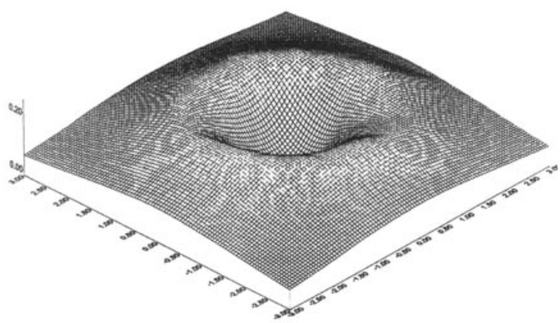

Figure 14. Clough–Tocher interpolant for h_{100} with derivatives D_{ct}^{loc}

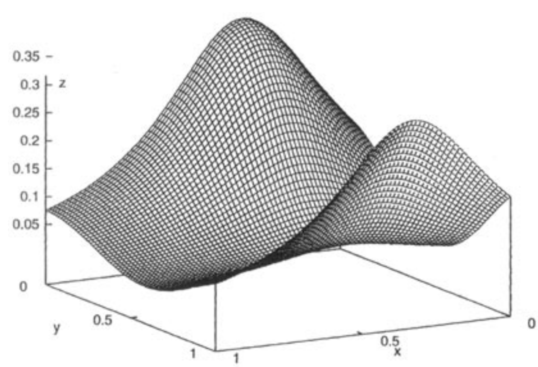

Figure 15. Saddle function s

senting the saddle function s and EX100 refers to a triangulation made by 100 points with equally spaced points (by 0.2) along the boundary used for representing the function h. D_{ct}^{loc} indicates that the derivatives are generated by minimizing the Clough–Tocher patch by means of the local method, D_{ct}^{gl} as before but by the global one. D_{q18}^{loc}, D_{q18}^{gl} indicate that the functional minimized the Q_{18}

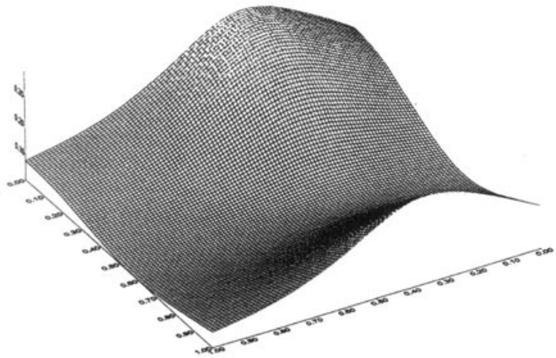

Figure 16. Clough–Tocher interpolant for s_{150} with derivatives D_{ct}^{loc}

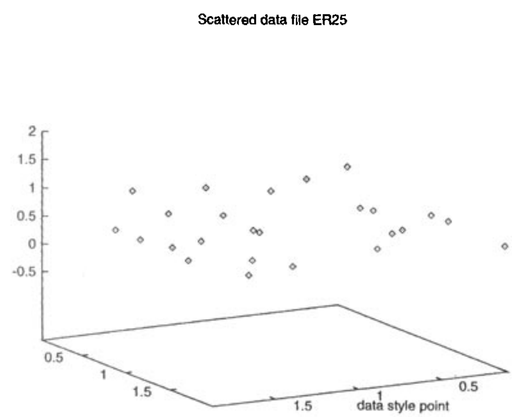

Figure 17. Random data points r_{25}

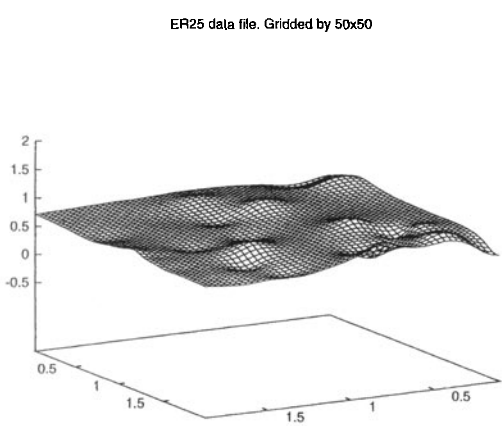

Figure 18. Interpolating on 50×50 grid for r_{25}

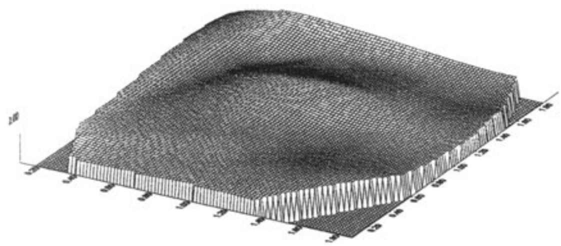

Figure 19. Clough–Tocher interpolant for r_{25} with derivatives D_{ct}^{loc}

polynomial, locally and globally respectively. Finally, D_{pws}^{loc}, D_{pws}^{gl} stand for the same quantities for the Powell–Sabin patch. max n_i and m are self-meaningful. The stars in the last row signify that it was impossible to make the computations due to the dimension of the matrix for the global approach. Thus, also for a relatively small set of points the global method can not be performed without resorting to techniques for large linear systems.

In Tables 2, 3 and 4 we present the values of the energy obtained by different methods of generating derivatives on different triangulations. The results show that the local method is less precise in the majority of the applications. But, if

Table 4. The energy $\phi(\mathbf{v})$ for the Powell–Sabin element on different triangulations and different methods of generating derivatives

Triangulation/Method	D_{ct}^{loc}	D_{ct}^{gl}	D_{q18}^{loc}	D_{q18}^{gl}	D_{pws}^{loc}	D_{pws}^{gl}
ES36_A	574.881	232.547	609.647	373.065	577.522	230.420
ES36_B	607.908	231.077	728.269	397.193	626.455	299.270
ES50	1389.812	259.795	1210.229	571.694	1185.349	251.132
ES100_A	97890.666	208.486	119283.418	331.588	107339.972	203.870
ES100	94529.937	12856.153	87255.628	248901.829	84316.425	4781.463
ES150	70849.476	238.806	27644.313	342.242	73005.426	235.457
EX100	6.724	2.098	7.862	3.377	6.971	2.079
ESA150	386.650	9.699	328.339	11.824	402.407	9.665
ER25	2657.431	505.357	2052.895	3678.869	2415.336	451.479

Table 5. The energy $\phi(\mathbf{v})$ for the Clough–Tocher element on different triangulations and different values of the parameters q, γ, β

Triang./q	$q=0.5$	$q=0.3$	$q=0.7$	$q=1.0$
ES36_A	373.641, $\gamma=-20$	406.448, $\gamma=-20$	348.255, $\gamma=-20$	**324.029**, $\gamma=-20$
ES36_B	400.798, $\gamma=-20$	476.119	459.508	**350.834**, $\gamma=-9$
ES50	**404.62**, $\gamma=2$	493.257, $\gamma=10$	538.05, $\gamma=2$	1178.217, $\gamma=2$
ES100_A	**304.62**, $\gamma=2$	14145.49, $\gamma=10$	9553.362, $\gamma=1$	51121.257, $\gamma=0.001$
ES100	50013.52, $\gamma=10$	**47661.18**, $\gamma=5$ $\beta=5$	49744.945, $\gamma=4$ $\beta=5$	60286.92, $\gamma=5$
ES150	**370.858**, $\gamma=2$ $\beta=10$	10514.3, $\gamma=3$ $\beta=20$	6805.58, $\gamma=1$ $\beta=1$	640.68, $\gamma=10^{-5}$ $\beta=1$
EX100	3.126, $\gamma=2$	3.73, $\gamma=3$	**2.915**, $\gamma=1$	4.1035, $\gamma=4$
ESA150	**13.206**, $\gamma=2$	63.093, $\gamma=3$	21.04, $\gamma=0.1$	295.6, $\gamma=1$
ER25	860.63, $\gamma=10$	1115.241, $\gamma=4$	**261.017**, $\gamma=1$	1017.456, $\gamma=0.001$

Table 6. The energy $\phi(\mathbf{v})$ for the Clough–Tocher element computed by three different methods

Triangulation/Method	Global by Alfeld	Local by Goodman et al.	Our local optimized
ES36_A	221.172	584.952	324.029
ES36_B	219.753	762.218	350.834
ES50	234.440	798.887	404.62
ES100_A	196.790	2717.501	304.62
ES100	2453.555	66183.515	47661.18
ES150	230.023	739.152	370.858
EX100	1.981	5.055	2.915
ESA150	9.533	739.152	13.206
ER25	420.590	1845.997	261.017

the triangulation is well designed, that is when on the boundary we take equally spaced points, then the results are of the same order of the global method. This is the case for the triangulations EX100, ES36_*. Obviously, the energy depends in some way on the area of the triangles, therefore when the triangulation has long and relatively large triangles, it comes easy to see that the energy grows.

4. Conclusions

In this paper we presented a method which generated partial derivatives (in its present form, up to order 2) well suited for example for scattered data problems requiring the construction of a smooth interpolant. The salient characteristics of the method are the following:

1. The method is *local*, since it generates the derivatives by considering a neighborhood of each node.
2. It is based on the minimization of the functional representing the linearized curvature of a local interpolant. It is important to notice that the linearization of the curvature is an assumption made for the sake of simplifying the computations. This hypothesis, assumed by most part of the methods based upon a variational approach, is indeed not always verified, as can be checked by tedious calculations.
3. The method applies the idea proposed by Renka and Cline [16] but not used there, probably due to the computational effort. Obviously, the method is a little less precise than the global one: this is because of the small set of information used to compute the derivatives in a data point. Despite of this, the method seems to be well-suited for computers with 'small' RAM, since it never requires the solution of a large linear system. The method was satisfactory applied to the representation of the Venice's lagoon bed, a problem consisting of 65 636 points and more then 131 000 triangles. For the discussion of this application and related results we refer the reader to [6].
4. Adding a new node to the triangulation, it only affects the cell to which the point belongs to. This advantage is not present in the global approach.

We conclude these remarks, noticing that the generation of derivatives from given data points is an *unnatural* process independently of the method used to perform the computations. In particular, with our method we have estimated the derivatives by considering an approximation of the curvature of an interpolant. This apparently interesting and natural choice, has the main drawback in the assumption that f_x and f_y are nearly zero. This can be correct in some cases, but generally produces surfaces affected by some noise.

Acknowledgements

This work has been supported by the MURST Numerical Analysis funds (60%-funds) and by a post-doctoral fellowship from the University of Padova. I am really indebted to Dr. Damiano Fasoli for his help in testing the formulas on many examples. Many thanks to Prof. Maria Morandi Cecchi of the University of Padova for giving to me the possibility to study this subject.

References

[1] Akima, H.: On estimating partial derivatives for bivariate interpolation of scattered data. Rocky Mountain J. Math. *14*, 41–52 (1984).

[2] Alfeld, P.: Derivative generation from multivariate scattered data by functional minimization. Comput. Aided Geom. Design 2, 281–296 (1985).

[3] Barnhill, R. E., Farin, G.: C^1 quintic interpolation over triangles: Two explicit representations. Int. J. Numer. Methods Eng. *17*, 1763–1778 (1981).

[4] Cline, A. K., Renka, R. J.: A constrained two-dimensional triangulation and the solution of closest node problems in the presence of barriers. SIAM J. Numer. Anal. *27*, 1305–1321 (1990).

[5] Clough, R. W., Tocher, J. L.: Finite element stiffness matrices for analysis of plates in bending. In: Conference on Matrix Methods in Structural Mechanics, Ohio, 1965. Wright Patterson A. F. B.

[6] De Marchi, S., Fasoli, D., Morandi Cecchi, M.: On representing the Venice's lagoon bed by a C^1 interpolating surface. Submitted to BIT.

[7] De Marchi, S., Fasoli, D., Morandi Cecchi, M.: **LABSUP**: a **LA**boraty for **B**ivariate C^1 **SU**rfaces and **P**atches. Technical Report, 10/96. University of Padova, Italy, 1996.

[8] Farin, G.: Triangular Bernstein–Bézier patches. Comput. Aided Geom. Design *3*, 83–127 (1986).

[9] Farin, G.: Curves and surfaces for CAGD. A practical guide, 3rd ed. London: Academic Press, 1993.

[10] Fasoli, D.: Interpolazione C^1 di dati scattered e procedure per la stima delle derivate parziali. Master's thesis, University of Padova, 1995 (in Italian).

[11] Goodman, T. N. T., Said, H. B., Chang, L. H. T.: Local derivative estimation for scattered data interpolation. Appl. Math. Comp. *68*, 41–50 (1995).

[12] Hsiung, C. C.: A first course in differential geometry. New York: John Wiley, 1981.

[13] Nielson, G. M.: A method for interpolating scattered data based upon a minimum norm network. Math. Comp. *40*, 253–271 (1983).

[14] Powell, M. J. D., Sabin, M. A.: Piecewise quadratic approximations on triangles. ACM Trans. Math. Software *3*, 316–325 (1977).

[15] Quak, E., Schumaker, L. L.: Cubic spline fitting using data dependent triangulations. Comput. Aided Geom. Design *7*, 293–301 (1990).

[16] Renka, R. J., Cline, A. K.: A triangle-based C^1 interpolation method. Rocky Mountain J. Math. *14*, 223–237 (1984).

[17] Risler, J. J.: Mathematical methods for CAD. Cambridge: Cambridge University Press, 1992.

[18] Schumaker, L. L.: Triangulation methods. In: Topics in multivariate approximation, (Schumaker, L. L., Chui, C. K., Utreras, F., eds.), pp. 219–273. New York: Academic Press, 1987.
[19] Stead, S. E.: Estimation of gradients from scattered data. Rocky Mountain J. Math. *14*, 265–279 (1984).
[20] Ženišek, A.: Interpolation polynomials on the triangle. Numer. Math. *15*, 283–296 (1970).

S. De Marchi
Università di Udine
Dipartimento di Matematica e Informatica
Via delle Scienze, 206 (Loc. Rizzi)
33100 Udine
Italy
e-mail: demarch@dimi.uniud.it