Lagrange Interpolation by C^1 Cubic Splines on Triangulated Quadrangulations

Günther Nürnberger[1)], Larry L. Schumaker[2)], and Frank Zeilfelder[3)]

Abstract. We describe local Lagrange interpolation methods based on C^1 cubic splines on triangulations obtained from arbitrary strictly convex quadrangulations by adding one or two diagonals. Our construction makes use of a fast algorithm for coloring quadrangulations, and the overall algorithm has linear complexity while providing optimal order approximation of smooth functions.

§1. Introduction

In this paper we are interested in constructing local Lagrange interpolation methods which are based on the space

$$\mathcal{S}_3^1(\triangle) := \{s \in C^1(\Omega) : s|_T \in \mathcal{P}_3, \text{ all } T \in \triangle\}$$

of C^1 cubic splines defined on a triangulation $\triangle$ of a planar domain Ω. While there are several C^1 cubic spline interpolation methods which make use of *Hermite data*, it is only recently that local schemes based only on *Lagrange data* have been developed, see [24] and the references therein. There are two approachs to developing such schemes. The first involves modifying a given triangulation by applying a Clough-Tocher split to some of the triangles, see [25–27]. The second approach works with certain triangulations which have been obtained from quadrangulations by inserting one diagonal in some of the quadrilaterals and both diagonals in others, see [21,22]. This paper is a continuation of [21,22], where we considered the following problem:

[1)] Institut für Mathematik, Universität Mannheim, D-618131 Mannheim, GERMANY nuern@euklid.math.uni-mannheim.de

[2)] Center for Constructive Approximation, Department of Mathematics, Vanderbilt University, Nashville, TN 37240, USA, s@mars.cas.vanderbilt.edu. Supported in part by the Army Research Office under grant DAAD-19-99-1-0160.

[3)] Institut für Mathematik, Universität Mannheim, D-618131 Mannheim, GERMANY zeilfeld@euklid.math.uni-mannheim.de

Problem 1.1. *Let $\mathcal{V} := \{\xi_i\}_{i=1}^n$ be a set of points in the plane, and let $\diamondsuit$ be a quadrangulation with vertices at the points of $\mathcal{V}$. Find a triangulation $\triangle$ of $\diamondsuit$ and a set of additional points $\{\xi_i\}_{i=n+1}^N$ such that for every choice of the data $\{z_i\}_{i=1}^N$, there is a unique C^1 cubic spline $s \in \mathcal{S}_3^1(\triangle)$ satisfying*

$$s(\xi_i) = z_i, \qquad i = 1, \ldots, N. \tag{1.1}$$

We call $P := \{\xi_i\}_{i=1}^N$ and $\mathcal{S}_3^1(\triangle)$ a Lagrange interpolation pair.

In [21] we solved this problem for a special class of so-called checkerboard quadrangulations, see Remark 7.1, while in [22] we extended the results to a larger class of so-called separable quadrangulations, see Remark 7.2. Here we deal with completely arbitrary strictly convex quadrangulations. In particular, we give explicit algorithms for creating the Lagrange interpolation pair P and $\mathcal{S}_3^1(\triangle)$ in such a way that for any given data, the interpolating spline depends locally on the data and can be constructed in $\mathcal{O}(n)$ operations. The associated triangulation $\triangle$ is constructed by first coloring $\diamondsuit$, and then using the coloring to add either one or two diagonals to each quadrilateral. In addition, the coloring is used to organize the interpolation points such that the method becomes local. We also show that the method produces optimal order approximation of smooth functions.

Throughout the paper we shall make extensive use of the well-known Bernstein-Bézier representation of splines, see [1,2,12–16,21,22] and references therein. We recall that in this representation of a cubic spline, for each triangle $T = \langle v_1, v_2, v_3\rangle$ in $\triangle$, the corresponding polynomial piece $p = s|_T$ is written in the form

$$p = \sum_{i+j+k=3} c_{ijk}^T B_{ijk}, \tag{1.2}$$

where B_{ijk} are the Bernstein basis polynomials of degree 3 associated with T. As usual, we identify the coefficients $\{c_{ijk}^T\}_{i+j+k=3}$ with the set of domain points $\mathcal{D}_T := \{\xi_{ijk}^T := (iv_1+jv_2+kv_3)/3\}_{i+j+k=3}$. Then a spline in $\mathcal{S}_3^0(\triangle)$ is uniquely determined by a set of coefficients where there is one coefficient associated with each point in the set

$$\mathcal{D} := \bigcup_{T\in\triangle} \mathcal{D}_T. \tag{1.3}$$

The paper is organized as follows. In Section 2 we describe an algorithm for coloring a quadrangulation to separate its quadrilaterals into classes. The result is used in Section 3 to define an associated triangulation $\triangle$ and corresponding set P of interpolation points. This section also contains the main result of the paper, which shows that the constructed P and $\mathcal{S}_3^1(\triangle)$ form a Lagrange interpolating pair. In Section 4 we compute the dimension of the spline space $\mathcal{S}_3^1(\triangle)$, and show that the associated cardinal basis splines form a stable local basis for $\mathcal{S}_3^1(\triangle)$. In Section 5 we define a corresponding interpolation operator, and show that it provides optimal order approximation of smooth functions. In Section 6 we discuss a simplification of the method which is based on a different coloring of quadrangulations. Finally, several remarks are collected in Section 7.

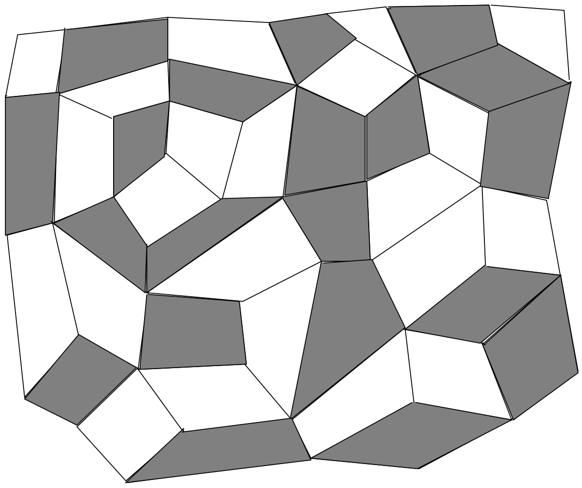

Fig. 1. A typical coloring produced by Algorithm 2.1.

§2. Coloring a Quadrangulation

A finite collection $\diamondsuit$ of quadrilaterals is said to be a quadrangulation of a connected set $\Omega \subset \mathbb{R}^2$ provided the union of the quadrilaterals in $\diamondsuit$ is equal to Ω, and provided the intersection of any two quadrilaterals in $\diamondsuit$ is either empty, a single point, or a common edge. We call $\diamondsuit$ a strictly convex quadrangulation if the largest angle in any quadrilateral $Q \in \diamondsuit$ is less than π. For references on constructing quadrangulations, see Remark 7.3.

Throughout the remainder of this paper we assume that $\diamondsuit$ is a strictly convex quadrangulation of a connected set Ω. We say that two quadrilaterals in $\diamondsuit$ are neighbors provided they have a common edge. Given a quadrangulation $\diamondsuit$, we write $V_\diamondsuit$ and $E_\diamondsuit$ for the number of vertices and edges, respectively.

We now present an algorithm for coloring quadrangulations which will be of importance later as a means for organizing the quadrilaterals of $\diamondsuit$ so that a Lagrange interpolation pair can be constructed.

Algorithm 2.1. *Start with any black and white coloring of $\diamondsuit$. Repeat until every quadrilateral of $\diamondsuit$ has at most two neighbors of the same color:*

1) *Choose a Q with at least three neighbors of the same color.*
2) *Switch the color of Q.*

Discussion: It is easy to see that the number of edges shared by two quadrilaterals with the same color decreases at each step. It follows that the algorithm terminates after at most $E_\diamondsuit$ steps. $\square$

Figure 1 shows an example of a quadrangulation that has been colored by this algorithm. As we shall see, it is convenient to deal with groups of quadrilaterals of the same color. Following [22], we say that a collection $\mathcal{C} := \{Q_1, \ldots, Q_m\}$ of quadrilaterals in $\diamondsuit$ is a connected component (of length m) provided that Q_i and Q_{i+1} are neighbors for each $i = 1, \ldots, m-1$. We call a connected component $\mathcal{C}$ a

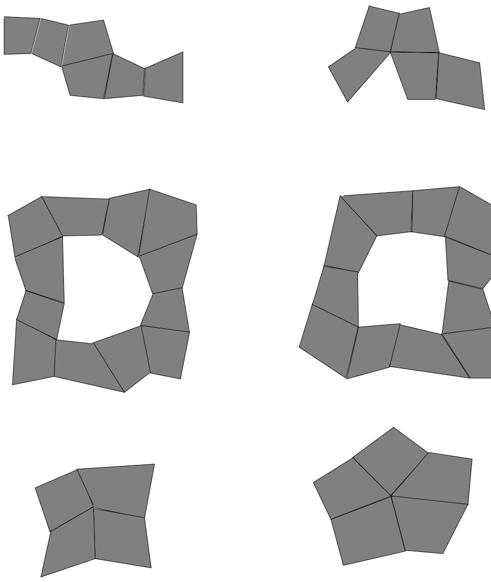

Fig. 2. Six different kinds of black chains.

chain if $|j-i|>1$ implies Q_i and Q_j are not neighbors (except that we allow Q_m and Q_1 to be neighbors, in which case we have a closed chain). It is clear that for any quadrangulation, Algorithm 2.1 produces a coloring which involves only chains in each of the two colors. Figure 2 shows the six different kinds of black chains that can arise.

§3. Construction of a Lagrange Interpolation Pair

We begin with an algorithm for triangulating $\diamondsuit$.

Algorithm 3.1. *Suppose $\diamondsuit$ is a quadrangulation that has been colored by Algorithm 2.1.*

1) *For every black component $\mathcal{C} := \{Q_1, \ldots, Q_m\}$, insert one diagonal in each of the odd numbered quadrilaterals of $\mathcal{C}$.*
2) *Insert both diagonals in all remaining quadrilaterals.*

We denote the triangulation produced by this algorithm by $\triangle$. Figure 3 shows $\triangle$ for the colored quadrangulation $\diamondsuit$ in Figure 1. The inserted diagonals are shown with dotted lines. In this example, $\diamondsuit$ consists of 42 quadrilaterals, where Algorithm 3.1 has inserted one diagonal in 15 of the quadrilaterals, and two diagonals in the remaining 27 quadrilaterals.

We are now ready to describe an algorithm for constructing a corresponding set P of interpolation points. As in [21,22], we shall choose P as a subset of the set $\mathcal{D}$ of domain points (1.3) associated with the Bernstein-Bézier representation of cubic splines. As is well known, $\mathcal{D}$ contains one point at each vertex v of $\triangle$, two points on each edge $e := \langle u, v\rangle$ located at $(2u+v)/3$ and $(u+2v)/3$, and one point at the center $\xi_T := (u+v+w)/3$ of each triangle $T := \langle u, v, w\rangle$. For each $v \in \triangle$, we

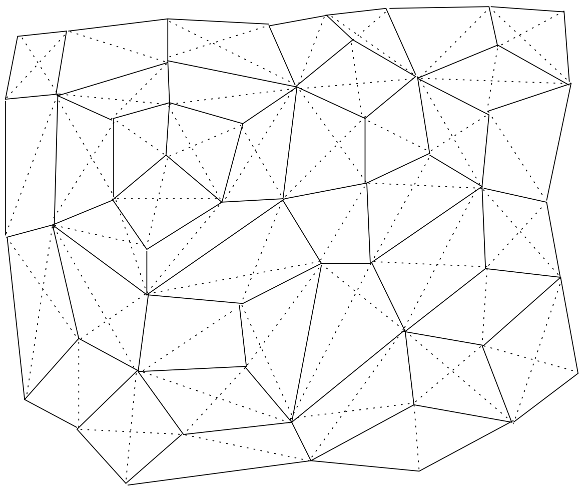

Fig. 3. The triangulation $\triangle$ resulting from Algorithm 3.1.

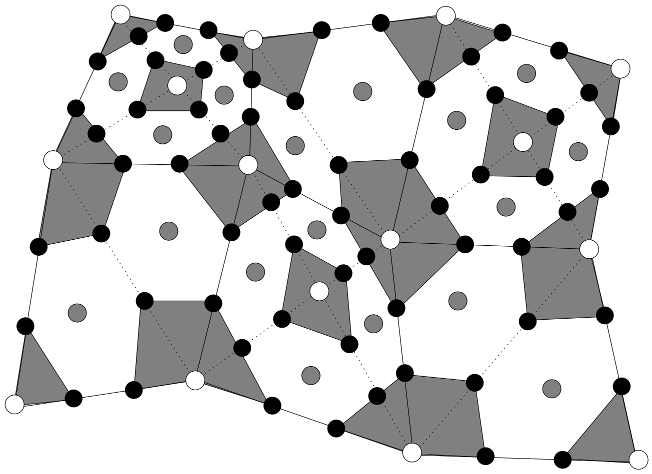

Fig. 4. Domain points: center points, rings and disks.

define the ring of radius one around v to be the set $R(v)$ consisting of the closer of the two domain points in the interior of each edge attached to v. We define the disk of radius one around v as $D(v) := R(v) \cup \{v\}$. Figure 4 illustrates these definitions. Points in the centers are marked with grey dots. Points in disks of radius one are shown in black and white, with those on rings of radius one being marked in black.

For each quadrilateral $Q \in \diamondsuit$ with two diagonals, we write v_Q for the point where the two diagonals of Q intersect. For convenience, we suppose the quadrilaterals in each component are numbered starting with one. We refer to a quadrilateral in a chain as being either even or odd depending on its subscript. We say that Q_1 and Q_m form an odd pair of quadrilaterals provided they are neighbors and m is odd. Clearly, such pairs can occur only in an odd closed chain.

Algorithm 3.2. *Initialize P by including all of the vertices of $\diamondsuit$. Then*

1) *For $\ell = 4, 3, 2, 1$, repeat as often as possible: If there exists $Q \in \diamondsuit$ with ℓ unmarked vertices $v_1, \ldots, v_\ell$, then for each $i = 1, \ldots, \ell$, add the points in $R(v_i)$ that lie on edges of Q, and mark $v_1, \ldots, v_\ell$.*
2) *Suppose $\mathcal{C}$ is a black component.*
   a) *For each odd pair Q_1, Q_m of quadrilaterals in $\mathcal{C}$, add the center point of one of the subtriangles of Q_1.*
   b) *For all other odd quadrilaterals in $\mathcal{C}$, add the center point of one of the subtriangles of Q.*
3) *For each even quadrilateral Q in a black component, let k be the number of other black quadrilaterals that share an edge with Q. We call such edges determined edges.*
   a) *If $k = 1$, pick some triangle T in Q and add the three points in $T \cap D(v_Q)$.*
   b) *If $k = 2$, add the point v_Q and one additional point ξ on $R(v_Q)$. If the two determined edges e_1, e_2 meet at a vertex u of Q, then ξ should be chosen on the edge $\langle v_Q, w \rangle$, where $w \neq u$ is an endpoint of either e_1 or e_2.*
4) *For each even quadrilateral Q in a white component, let k be the number of edges of Q shared with black quadrilaterals, and call these determined edges.*
   a) *If $k = 0$, choose some triangle T in Q and add the center ξ_T and the three points in $T \cap D(v_Q)$.*
   b) *If $k = 1$, choose some triangle T in Q and add the three points in $T \cap D(v_Q)$.*
   c) *If $k = 2$, add the point v_Q and and one additional point ξ on $R(v_Q)$. If the two determined edges e_1, e_2 meet at a vertex u of Q, choose ξ on the edge $\langle v_Q, w \rangle$, where $w \neq u$ is an endpoint of either e_1 or e_2.*
   d) *If $k = 3$, add v_Q.*
5) *For each odd quadrilateral Q in a white component (except for the last quadrilateral in an odd closed chain), let k be the number of edges of Q that are shared with either black or other white quadrilaterals. Call these determined edges, and choose additional points for P as in Step 4.*
6) *If Q_m is the last quadrilateral in an odd closed white chain, let k be defined as in Step 5, but do not count the edge between Q_m and Q_1. Then add additional points to P as in Step 4.*

Figure 5 shows the result of applying Algorithm 3.2 to the quadrangulation in Figure 1, where the interpolation points are marked with black and white dots.

Before showing that the algorithm produces a Lagrange interpolation pair, we make some remarks on the steps of the algorithm. Step 1 is the key to the whole process, as the points chosen there uniquely determine s on all of the edges of $\diamondsuit$ (in Figure 5 these points are marked with white dots). The order in which the points are chosen is critical for ensuring that the resulting interpolation method is local (see below). Steps 2 and 3 deal with chains of black quadrilaterals, and the idea is

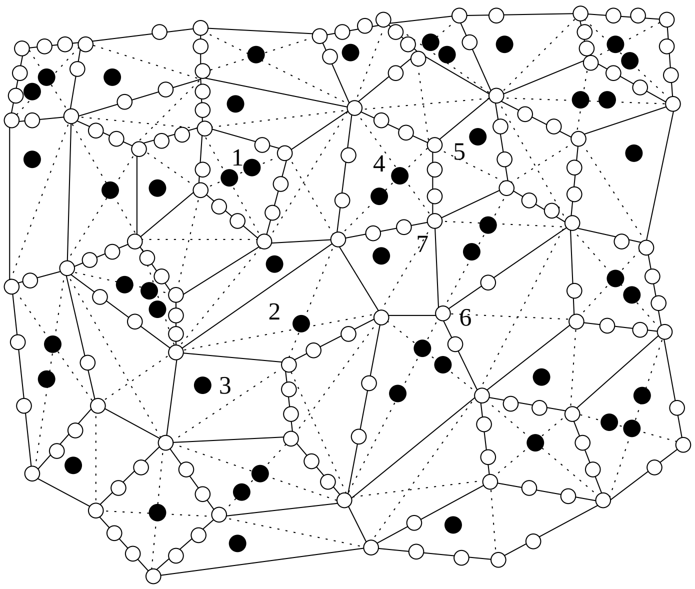

Fig. 5. The point set P produced by Algorithm 3.2.

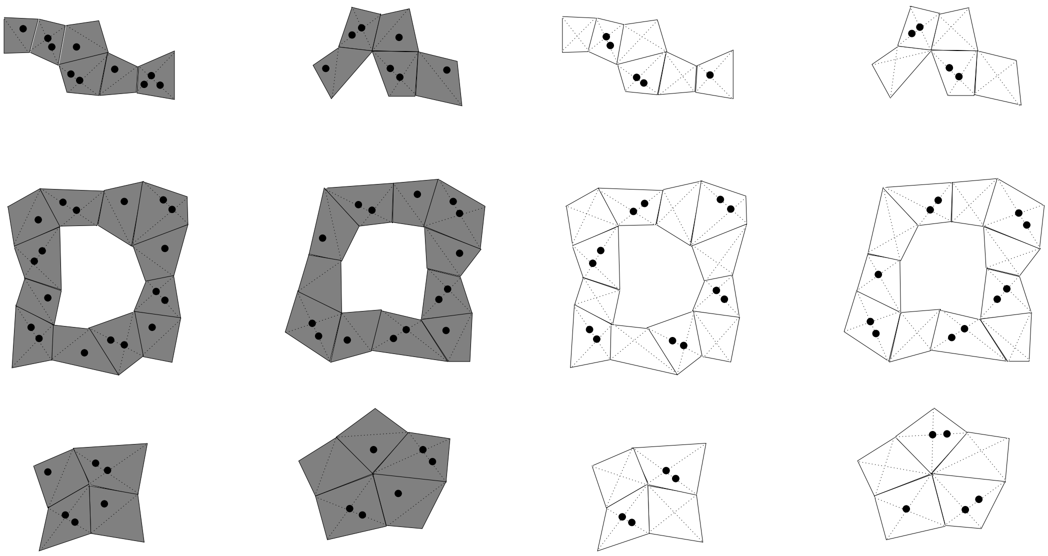

Fig. 6. Standard choices of interpolation points in Steps 2– 6 of Algorithm 3.2.

to choose enough points in each even black quadrilateral to uniquely determine s there. Steps 4–6 deal with chains of white quadrilaterals. Here we first do the even white quadrilaterals, then the odd ones. The interpolation points chosen in Steps 2–6 of the algorithm are marked with black dots in Figure 5.

To further illustrate the algorithm, in Figure 6 we show some standard choices of points for both black and white components. If a white quadrilateral contains boundary edges, then Algorithm 3.2 picks certain additional points. Figure 7 shows the various cases which can arise in Step 4 when we add one, two, three, or four

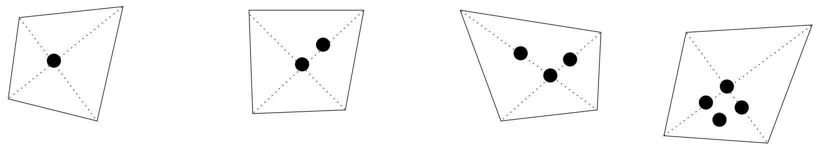

Fig. 7. Choice of points in Step 4 of Algorithm 3.2.

points, respectively.

As we shall see, in each of these steps the interpolation points have been chosen carefully to ensure that the corresponding interpolating spline depends only locally on the data. We emphasize that, in contrast with the case of Hermite interpolation (see [2,9,12,14]), except at vertices, there are no simple relations between an interpolation condition at a point of $\mathcal{D}$ and the Bernstein-Bézier coefficient associated with that point.

In the proof of our main result on interpolation (Theorem 3.4, below) we make use of the following elementary lemma.

Lemma 3.3. *Suppose $\triangle_Q$ is the triangulation that is obtained by inserting both diagonals in a quadrilateral $Q := \langle v_1, v_2, v_3, v_4\rangle$, and suppose $T_i = \langle v_Q, v_i, v_{i+1}\rangle$, $i = 1,\ldots,4$, are the four triangles of $\triangle_Q$. Let Γ_E be the set of 12 domain points situated on the edges of Q, and let*

$$\begin{aligned}
&\mathcal{M}_1 = \{\xi_{T_1}, \xi_{T_2}, \xi_{T_3}, \xi_{T_4}\}, \quad \mathcal{M}_2 = \{\xi_{T_1}, \xi_{T_2}, \xi_{T_3}, v_Q\},\\
&\mathcal{M}_3 = \{\xi_{T_1}, \xi_{T_3}, v_Q, (2v_Q + v_1)/3\}, \quad \mathcal{M}_4 = \{\xi_{T_1}, \xi_{T_2}, v_Q, (2v_Q + v_1)/3\},\\
&\mathcal{M}_5 = \{\xi_{T_1}, v_Q, (2v_Q + v_1)/3, (2v_Q + v_2)/3\}.
\end{aligned}$$

Then for each $\ell \in \{1,\ldots,5\}$, the set

$$\Gamma_\ell := \Gamma_E \cup \mathcal{M}_\ell$$

is a minimal determining set for $\mathcal{S}^1_3(\triangle_Q)$, i.e., any spline $s \in \mathcal{S}^1_3(\triangle_Q)$ is uniquely determined by the B-coefficients associated with the domain points Γ_ℓ.

Proof: It is clear that given coefficients associated with Γ_E, we can uniquely compute the other coefficients of s associated with domain points in the disks $D(v_i)$, $i = 1, 2, 3, 4$. This leaves the nine coefficients $c_1,\ldots,c_8, c_Q$ associated with the domain points $a_1,\ldots,a_8, a_Q$ shown in Figure 8. Then the C^1 smoothness conditions imply that

$$\begin{aligned}
c_5 &= rc_1 + (1-r)c_2, & c_6 &= \tilde r c_2 + (1-\tilde r)c_3,\\
c_7 &= rc_4 + (1-r)c_3, & c_8 &= \tilde r c_1 + (1-\tilde r)c_4,\\
c_Q &= rc_8 + (1-r)c_6, & c_Q &= \tilde r c_5 + (1-\tilde r)c_7.
\end{aligned} \tag{3.1}$$

where $0 < r, \tilde r < 1$ are such that

$$v_Q = rv_1 + (1-r)v_3 = \tilde r v_2 + (1-\tilde r)v_4.$$

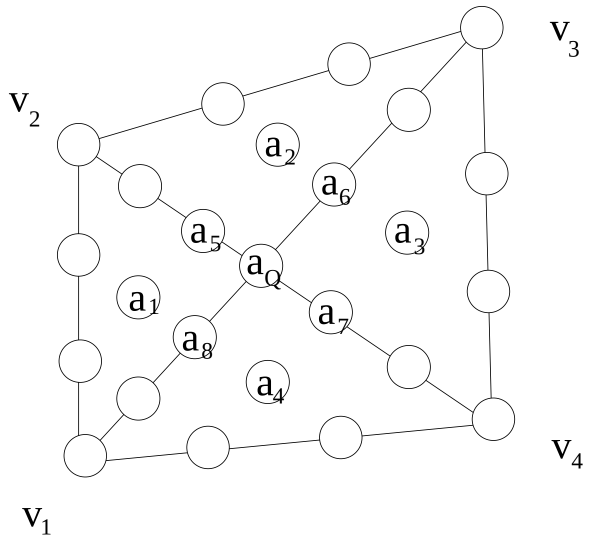

Fig. 8. Notation for the domain points in Lemma 3.3.

We now claim that in each of the five cases, the coefficients $c_1, \ldots, c_9$ which are not set are uniquely determined by (3.1). This is well known for the case $\ell = 1$, see [10,12,28]. Indeed, in this case we can use (3.1) directly to compute c_5, c_6, c_7, c_8 and c_Q. While it appears that c_Q has been determined in two possibly different ways, in fact both expressions reduce to

$$c_Q = r\tilde{r}(c_1 - c_2 + c_3 - c_4) + r(c_4 - c_3) + \tilde{r}(c_2 - c_3) + c_3.$$

We now consider the case $\ell = 2$, and use (3.1) to compute c_5, c_6, c_7, c_8 in that order. Then c_4 can be determined in two different ways, but we get the same value either way. The other cases are similar. $\square$

We are now ready to state and prove our main result on interpolation.

Theorem 3.4. *Let $\triangle$ and $P := \{\xi_i\}_{i=1}^N$ be the triangulation and point set constructed by Algorithm 3.2. Then P and $\mathcal{S}_3^1(\triangle)$ form a Lagrange interpolation pair, and $\dim \mathcal{S}_3^1(\triangle) = N$.*

Proof: We show that given any data $z := \{z_i\}_{i=1}^N$, there exists a *unique* $s \in \mathcal{S}_3^1(\triangle)$ satisfying the interpolation conditions (1.1). We suppose s is represented in Bernstein-Bézier form as described in the introduction. We need to show that each of the B-coefficients $\{c_\xi\}_{\xi \in \mathcal{D}}$ is uniquely defined by the data, where $\mathcal{D}$ is the set of domain points (1.3). First, we note that for each domain point ξ lying at a vertex of $\diamondsuit$, the corresponding coefficient is equal to the data value associated with that point.

We now show how to compute the coefficients of s associated with domain points on the edges of $\diamondsuit$. Algorithm 3.2 divides the quadrilaterals of $\diamondsuit$ into four classes $\diamondsuit_\ell$, where we say that $Q \in \diamondsuit_\ell$ if Q was a quadrilateral with ℓ unmarked vertices in carrying out Step 1 of the algorithm. Now consider $Q \in \diamondsuit_4$. For each edge e of Q, P includes the two domain points on the interior of e, and thereby uniquely determines the B-coefficients associated with those two domain points, see Remark 7.11. By the C^1 smoothness, this uniquely determines all B-coefficients of s in the disks $D(v)$ where v is a vertex of a quadrilateral in $\diamondsuit_4$.

We now examine quadrilaterals $Q \in \diamondsuit_3$. Suppose $Q := \langle u_1, u_2, u_3, u_4 \rangle$ is such a quadrilateral. Then Q shares exactly one vertex, say u_4, with some quadrilateral in $\diamondsuit_4$. The C^1-continuity at u_4 determines all the B-coefficients associated with domain points in the disk $D(u_4)$. The B-coefficients associated with domain points in the interiors of the edges $\langle u_1, u_2 \rangle$ and $\langle u_2, u_3 \rangle$ are uniquely determined by the interpolation conditions on those edges. On the edge $e := \langle u_1, u_4 \rangle$, we already know all but one coefficient, and that is determined by the one interpolation condition associated with the point $R(u_1) \cap e$, see Remark 7.11. Similarly, the coefficient associated with the point $R(u_3) \cap \langle u_3, u_4 \rangle$ is determined by the interpolation condition at that point.

We repeat this argument for quadrilaterals in $\diamondsuit_2$, and then finally for quadrilaterals in $\diamondsuit_1$ to complete the proof that all of the B-coefficients of s corresponding to domain points on the edges of $\diamondsuit$ are uniquely determined. It follows that all of the coefficients corresponding to domain points in the disks $D(v)$ surrounding vertices of $\diamondsuit$ are also uniquely determined.

It remains to consider B-coefficients associated with domain points lying inside quadrilaterals. We begin by examining quadrilaterals that are part of black chains. Suppose Q is an odd black quadrilateral which is not part of an odd pair of black quadrilaterals. Then it has been split into two triangles T_1 and T_2, and we have an interpolation condition at a center point, say ξ_{T_1} of one of the two triangles. Since we already know all B-coefficients associated with domain points on the edges of both T_1 and T_2, the interpolation condition at ξ_{T_1} uniquely determines the B-coefficient associated with that domain point, see Remark 7.12. The B-coefficient associated with the domain point ξ_{T_2} is then uniquely determined by C^1 smoothness across the edge between T_1 and T_2. If Q is part of an odd pair of black quadrilaterals, then the arguments are similar.

We now consider even black quadrilaterals. Suppose Q is such a quadrilateral which has been split into four subtriangles $T_1, \ldots, T_4$, and let $k \in \{1, 2\}$ be the number of determined edges of Q. For each such edge, C^1 smoothness uniquely defines the B-coefficient corresponding to the center of the triangle T_i sharing that edge. But then the $4 - k$ interpolation conditions associated with points chosen in Step 3 of the algorithm coupled with C^1 smoothness across the edges meeting at v_Q uniquely determine all remaining B-coefficients. To see this, suppose $k = 1$. Then using the data from Step 3a of Algorithm 3.2, we can apply Remark 7.11 to compute the coefficients corresponding to the domain points in the set $\mathcal{M}_5$ of Lemma 3.3, and then apply the lemma. If $k = 2$, we can use the data from Step 3b of Algorithm 3.2 and Remark 7.11 to get coefficients in one of the sets $\mathcal{M}_3$ or $\mathcal{M}_4$ of the lemma.

Now consider even white quadrilaterals. These quadrilaterals are also split into four subtriangles. Given such a quadrilateral Q, let $k \in \{0, \ldots, 4\}$ be the number of determined edges. As before, for each such edge C^1 smoothness uniquely determines the B-coefficient corresponding to the center of the triangle sharing that edge. But then the $4-k$ interpolation conditions corresponding to the points assigned in Step 4 of the algorithm coupled with C^1 smoothness conditions serve to uniquely determine all remaining B-coefficients associated with points in Q. For example, with $k = 0$ we use the data in Step 4a of Algorithm P and apply Remark 7.11 to compute the B-coefficients associated with $D(v_Q)$. Then we proceed as in Remark 7.12 to compute the B-coefficient associated with the center point ξ_{T_1} of one subtriangle T_1 of Q and we use Lemma 3.3 with $\ell = 5$ to uniquely compute the remaining coefficients. The other cases are similar.

To complete the proof that P and $\mathcal{S}_3^1(\triangle)$ form a Lagrange interpolation pair, we deal with the odd white quadrilaterals in the same way as the even white quadrilaterals, but using the points assigned in Step 6 of the algorithm, followed by those in Step 5.

Finally, the fact that interpolation at the points of P *uniquely* determines an $s \in \mathcal{S}_3^1(\triangle)$ immediately implies that $\dim \mathcal{S}_3^1(\triangle) = \#P = N$. $\square$

The proof of Theorem 3.4 describes a step–by–step algorithm for computing a spline s interpolating a given set of data. In particular, after setting the coefficients corresponding to the domain points at the vertices of $\diamondsuit$, we determine the remaining coefficients as follows:

1) by computing the coefficients of the univariate cubic polynomials that represent s on a subset of the edges of $\diamondsuit$,
2) by using C^1 smoothness conditions to determine all remaining coefficients in the disks $D(v)$,
3) by solving for the coefficient corresponding to a Bernstein-basis function B_{111}^T associated with a triangle T by enforcing an interpolation condition at the domain point ξ_{111}^T at the center of T,
4) by computing the coefficients of the univariate cubic polynomials that represent s on certain edges of $\triangle$ inside the quadrilaterals of $\diamondsuit$,
5) by using C^1 smoothness conditions across edges of $\triangle$ to determine the remaining coefficients.

In particular, as described in the proof of Theorem 3.4, coefficients are either determined by explicit equations, or by solving 2×2 linear systems.

Theorem 3.5. *Suppose P and $\triangle$ are a Lagrange interpolating pair which have been constructed from a given quadrangulation $\diamondsuit$ with Algorithm 3.2. Then the corresponding Lagrange interpolating spline can be constructed in $\mathcal{O}(n)$ operations, where n is the number of vertices of $\diamondsuit$. Moreover, there exists a constant K depending only on*

$$\alpha_\diamondsuit := \textit{smallest angle in } \diamondsuit, \qquad \beta_\diamondsuit := \textit{largest angle in } \diamondsuit, \tag{3.2}$$

such that

$$\|c\| \le K\|z\|,$$

z and c are the vectors of data values and coefficients of s, respectively, and where $\|\cdot\|$ measures the maximum norm.

Proof: The first assertion follows from the fact that each of the five computations listed above requires only a few flops. We now claim that each of these computations is also stable. First, we note (cf. Remark 7.11) that the computations in 1) and 4) are absolutely stable (*i.e.*, they don't depend on any property of $\Diamond$). As observed in Remark 7.12, the computations in 3) are also absolutely stable. The direct use of smoothness conditions as in 2) is well known to be stable, but the constant depends on both $\alpha_\Diamond$ and $\beta_\Diamond$. Finally, the use of C^1 smoothness conditions as in 5) is also stable as can be seen by examining the proof of Lemma 3.3. $\square$

We conclude this section by noting that this computational algorithm is also local in the sense that for each domain point ξ, the corresponding B-coefficient c_ξ depends only on the data values at points in a neighborhood of ξ. Another way to say this is as follows. Suppose we set a particular data value z_η to one, and set all other data values to zero. Then, as we shall show in Theorem 4.3, the corresponding interpolating spline (sometimes called a cardinal spline) will have nonzero coefficients only for domain points which are sufficiently close to η.

§4. Dimension and a Local Basis for $\mathcal{S}_3^1(\triangle)$

If P and $\mathcal{S}_3^1(\triangle)$ form a Lagrange interpolation pair, then as proved in Theorem 3.4, the dimension of $\mathcal{S}_3^1(\triangle)$ is equal to the number of points in P. In this section we give an explicit formula for this number which (although it may be of no particular practical importance) is useful for comparing sizes of spline spaces.

We recall that $V_\Diamond$ and $E_\Diamond$ denote the number of vertices and edges of $\Diamond$, respectively. Let n_1 be the number of quadrilaterals for which Algorithm 3.1 inserts just one diagonal in constructing $\triangle$.

Theorem 4.1. *Suppose P and $\mathcal{S}_3^1(\triangle)$ are a Lagrange interpolation pair produced by Algorithm 3.2. Then*

$$\dim \mathcal{S}_3^1(\triangle) = 3V_\Diamond + E_\Diamond - 3n_1. \tag{4.1}$$

Proof: Consider $\mathcal{S}_3^1(\diamondplus)$, where $\diamondplus$ is the triangulation obtained from $\Diamond$ by drawing both diagonals in each quadrilateral. As shown in [12], the dimension of this space is $3V_\Diamond + E_\Diamond$. By construction, the only difference between $\diamondplus$ and our triangulation $\triangle$ is that in some quadrilaterals we have only one diagonal instead of two. Such quadrilaterals are either isolated, or occur in neighboring pairs. For an isolated quadrilateral Q, it is easy to see that the dimensions of the C^1 cubic spline space on these two different triangulations of Q are 16 and 13, respectively, and so the difference in dimension is exactly three. In the case of a neighboring pair, the difference in dimension is exactly six. $\square$

Example 4.2. *Let $\Diamond$ be the quadrangulation shown in Figure 1.*

Discussion: For this quadrangulation we have $V_\diamondsuit = 53$ and $E_\diamondsuit = 94$, and Algorithm 3.1 inserts only one diagonal for $n_1 = 15$ quadrilaterals of $\diamondsuit$, and both diagonals for the remaining quadrilaterals. Thus, Theorem 4.1 gives

$$\dim \mathcal{S}_3^1(\triangle) = 3 \cdot 53 + 94 - 3 \cdot 15 = 208.$$

For comparision purposes, we note that $\dim \mathcal{S}_3^1(\diamondplus) = 253$ when $\diamondplus$ is the triangulation obtained from $\diamondsuit$ by drawing both diagonals in each quadrilateral. The point set P of the Lagrange interpolation pair P for $\mathcal{S}_3^1(\triangle)$ produced by Algorithm 3.2 (based on the coloring of Figure 1) is shown in Figure 5, where the interpolation points are marked with black and white dots. Counting the black and white dots gives 159 and 49, respectively, which sums up to $N = 208$. $\square$

Although not needed for computational purposes, as a theoretical tool it is convenient to introduce the cardinal basis functions associated with the Lagrange interpolation pair P and $\mathcal{S}_3^1(\triangle)$. These are the unique splines $B_\xi \in \mathcal{S}_3^1(\triangle)$ satisfying

$$B_\xi(\eta) = \delta_{\xi,\eta}, \qquad \text{all } \xi, \eta \in P. \tag{4.2}$$

These basis functions form a stable local basis in the sense that there exist constants K and ℓ depending only on the constants $\alpha_\diamondsuit$ and $\beta_\diamondsuit$ defined in (3.2) such that

1) $\|B_\xi\| \le K$ for all $\xi \in P$,
2) for each $\xi \in P$, there exists a quadrilateral Q such that $\operatorname{supp}(B_\xi) \subseteq \operatorname{star}^\ell(Q)$,

where as usual, $\operatorname{star}(Q)$ is the union of all quadrilaterals that intersect with Q in at least one point, and $\operatorname{star}^m(Q) := \operatorname{star}^{m-1}(\operatorname{star}(Q))$ for all $m > 0$. Indeed, we can take K to be the same constant mentioned in the stability discussion in Sect. 3, and the following result shows that we can take $\ell = 9$.

Theorem 4.3. *Given $\xi \in P$, let B_ξ be the corresponding cardinal spline satisfying (4.2). Suppose Q_ξ is a quadrilateral which contains the point ξ. Then*

1) *if ξ is strictly inside Q_ξ, then* $\operatorname{supp}(B_\xi) \subseteq \operatorname{star}^5(Q_\xi)$.
2) *if ξ lies on the boundary of Q_ξ, then* $\operatorname{supp}(B_\xi) \subseteq \operatorname{star}^9(Q_\xi)$.

Proof: We analyse the worst cases, i.e., where the support of B_ξ is of maximal size.

Case 1. Suppose ξ lies strictly inside the quadrilateral Q_ξ. In this case we have zero data values associated with all vertices of $\diamondsuit$ and with all of the points added to P in Step 1 of Algorithm 3.2. This implies (cf. the proof of Theorem 3.4) that all B-coefficients associated with domain points on the edges of $\diamondsuit$ must be zero, or equivalently, s must vanish on this net of edges. It also implies that all coefficients associated with the disks $D(v)$ surrounding vertices of $\diamondsuit$ are also zero.

To see what can happen with the remaining coefficients, suppose $\mathcal{C} := \{Q_1, \ldots, Q_m\}$ is an odd black closed chain with $m \ge 5$ and that ξ is the center of one of the two subtriangles T_1 and T_2 of Q_1. Then the interpolation condition

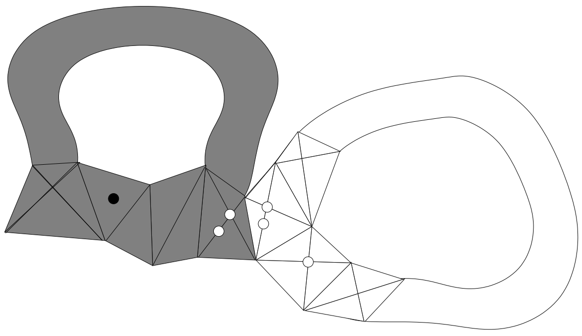

Fig. 9. Maximal support of B_ξ when ξ is inside a quadrilateral Q_ξ.

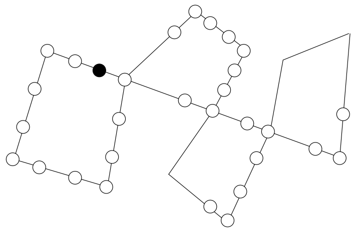

Fig. 10. Maximal propagation along the edges of $\diamondsuit$.

$B_\xi(\xi) = 1$ implies that c_ξ is nonzero, and using the C^1 smoothness condition, we see that the coefficient associated with the center of T_2 can also be nonzero. Now since Q_m is also odd, C^1 smoothness across the common edge with Q_1 can lead to nonzero coefficients associated with the centers of its two subtriangles. These nonzero values can propagate into both Q_2 and Q_{m-1}, but cannot propagate any further in $\mathcal{C}$ since Q_3 and Q_{m-2} are odd, and the data implies all coefficients on these quadrilateral must be zero.

We now trace the propagation into white quadrilaterals. Suppose $\tilde{\mathcal{C}} := \{\tilde{Q}_1, \ldots, \tilde{Q}_{\tilde{m}}\}$ is an odd white closed chain with $\tilde{m} \geq 5$ and that the quadrilaterals Q_{m-1} and $\tilde{Q}_{\tilde{m}-1}$ share a common edge. The C^1 smoothness across this common edge can lead to nonzero coefficients associated with domain points inside of $\tilde{Q}_{\tilde{m}-1}$. Then we can get additional propagation into the quadrilaterals $\tilde{Q}_{\tilde{m}-2}$, $\tilde{Q}_{\tilde{m}}$, and $\tilde{Q}_1$. However, the propagation now stops since $\tilde{Q}_2$ and $\tilde{Q}_{\tilde{m}-3}$ are both even, and the data implies all coefficients on these quadrilaterals must be zero. It follows that the support of B_ξ in this case can be as much as $\mathrm{star}^5(Q_\xi)$. Since it is not hard to see that this is the worst case under the assumption that ξ is strictly inside the quadrilateral Q_ξ, this completes the proof of Case 1.

Case 2. Suppose ξ lies in the interior of an edge $e := \langle u_1, \tilde{u}_1 \rangle$ of the quadrilateral Q_ξ (the case that ξ is a vertex of $\diamondsuit$ can be treated analogously). To identify the support of B_ξ in this case, we now examine the computation of its coefficients following the proof of Theorem 3.4 and using the notation there.

Suppose $Q_0 := Q_\xi$ lies in the class $\diamondsuit_4$. Then the first step of finding the coefficients of B_ξ involves computing coefficients corresponding to domain points

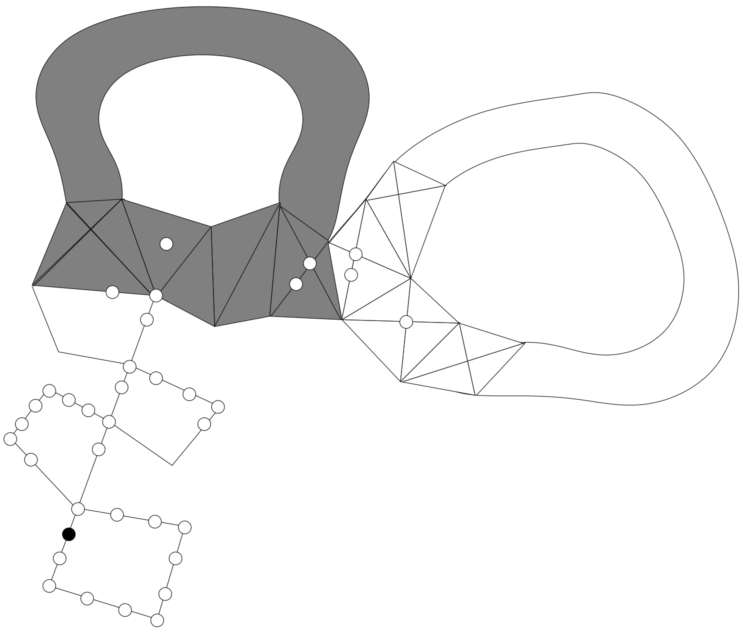

Fig. 11. Maximal support of B_ξ when ξ is on the edge of a quadrilateral Q_ξ.

on the edges of all quadrilaterals in $\diamondsuit_4$. All such coefficients must be zero except for those associated with points on the edge e of Q_0. This implies that all coefficients associated with points in the disks $D(v)$ surrounding vertices of $\diamondsuit_4$ must be zero, except for the two disks $D(u_1)$ and $D(\tilde{u}_1)$.

We now consider quadrilaterals in $\diamondsuit_3$. Suppose Q_1 lies in the class $\diamondsuit_3$. Then, Q_1 has exactly one vertex v in common with a quadrilateral from $\diamondsuit_4$, since otherwise Q_1 would not be an element of $\diamondsuit_3$. If $v \notin \{u_1, \tilde{u}_1\}$, then all coefficients associated with points in the disks $D(v)$ are zero, and therefore all coefficients corresponding to domain points on the edges of Q_1 are zero. Otherwise, if either $v = u_1$ or $v = \tilde{u}_1$, we get zero values for the coefficients associated with domain points on two edges of Q_1, and possibly nonzero values on the two remaining edges $\langle u_2, v\rangle$ and $\langle \tilde{u}_2, v\rangle$ of Q_1. This implies that all coefficients associated with points in the disk $D(w)$ surrounding the vertex w of Q_1 with $w \notin \{v, u_2, \tilde{u}_2\}$ must be zero, and some of the coefficients associated with the points in the disks $D(u_2)$ and $D(\tilde{u}_2)$ are possibly nonzero. Since there does not exist a quadrilateral $\tilde{Q}_1 \in \diamondsuit_3$ different from Q_1 with vertex u_2 or $\tilde{u}_2$ (otherwise $\tilde{Q}_1$ would not be an element of $\diamondsuit_3$) all the possible nonzero coefficients of B_ξ associated with domain points in the quadrilaterals of $\diamondsuit_3 \cup \diamondsuit_4$ lie in $\operatorname{star}^1(Q_0)$.

A similar argument shows that all the possible nonzero coefficients of B_ξ associated with domain points in the quadrilaterals of $\diamondsuit_2 \cup \diamondsuit_3 \cup \diamondsuit_4$ lie in $\operatorname{star}^2(Q_0)$. Note that a possible common vertex of two different quadrilaterals $Q_2, \tilde{Q}_2 \in \diamondsuit_2$ is already a vertex of a quadrilateral from $\diamondsuit_3 \cup \diamondsuit_4$. Finally, looking at quadrilaterals in $\diamondsuit_1$, we see that all the possible nonzero coefficients of B_ξ associated with domain points in the quadrilaterals of $\diamondsuit_1 \cup \diamondsuit_2 \cup \diamondsuit_3 \cup \diamondsuit_4$ lie in $\operatorname{star}^3(Q_0)$. In addition, for every quadrilateral Q_3 from $\diamondsuit_1$ in $\mathrm{star}^3(Q_0)$ some coefficients associated with the points in the disks $D(u_4)$ can possibly be nonzero, where u_4 is the vertex of Q_3 which was marked in Step 1 when $\ell = 1$.

Now there can be no further propagation along the edges of $\diamondsuit$. It is easy to see that starting with Q_ξ in $\diamondsuit_4$ leads to the worst possible case (concerning propagation along the edges of $\diamondsuit$).

We now examine the process of computing the coefficients of B_ξ associated with domain points which lie strictly inside of quadrilaterals. Suppose that Q_4 lies in $\mathrm{star}^4(Q_\xi)$, and B_ξ has nonzero coefficients associated with domain points in the disk $D(u_4)$ for some vertex u_4 of Q_4 (i.e., u_4 was marked in Step 1 when $\ell = 1$). Moreover, assume that Q_4 is the first quadrilateral in an odd black closed chain. Then using the C^1 smoothness condition, we see that the B-coefficient of B_ξ associated with the center of one of the two subtriangles of Q_4 can be nonzero. Therefore, the nonzero coefficients in the disks $D(u_4)$ possibly initiate further chains of propagation through at most two additional black quadrilaterals, and three additional white ones (see Case 1). Combining these observations, we see that the maximal support B_ξ is $\mathrm{star}^9(Q_\xi)$ as asserted. $\square$

Figures 9–11 illustrate the proof of Lemma 4.3 for the two cases. Here the point ξ is marked with a black dot, while the remaining interpolation points involved in the propagation process are marked with white dots. Figure 10 shows the maximal propagation along the edges of $\diamondsuit$ for a point ξ which has been chosen in Step 1 of Algorithm 3.2. Combining this with the maximal propagation for a point ξ which lies strictly inside of a quadrilateral Q_ξ (Figure 9) gives the maximal propagation for Case 2 (Figure 11).

Although in the worst cases there may be cardinal basis functions with rather large supports, such cases are actually quite rare, and most of the time the basis splines have much smaller supports. Indeed, the supports frequently consist of a small chain of quadrilaterals. This can be seen from the following observations:

1) Given a coloring produced by Algorithm 2.1, in general there are only a few closed (odd) chains of one color, see also Remark 7.6.
2) Maximal propagation along the edges can only happen when there exists a chain of edges $e_i = \langle v_i, v_{i+1}\rangle$, $i = 0, \ldots, 2$, of $\diamondsuit$ such that in step 1 of Algorithm 3.2, the vertex $v_{4-\ell}$ was unmarked for $\ell = 4, 3, 2, 1$. By the nature of step 1 (choosing quadrilaterals with ℓ unmarked vertices as long as possible), such chains of edges with maximal length are often avoided automatically.
3) When propagation of the coefficients of B_ξ occurs along edges and through the quadrilaterals in one direction of the quadrangulation, it often happens that the propagation in all other directions is much shorter (or does not occur). Therefore, in most cases there exists a quadrilateral $Q \neq Q_\xi$ such that the support of B_ξ is actually in $\mathrm{star}^{\ell_1}(Q) \subseteq \mathrm{star}^{\ell_2}(Q_\xi)$ with $\ell_1 << \ell_2$.

Figure 12 shows the supports for the cardinal basis functions B_ξ, where ξ are the points numbered from 1 to 7 in Figure 5 for the quadrangulation $\diamondsuit$ (and coloring) in Figure 1.

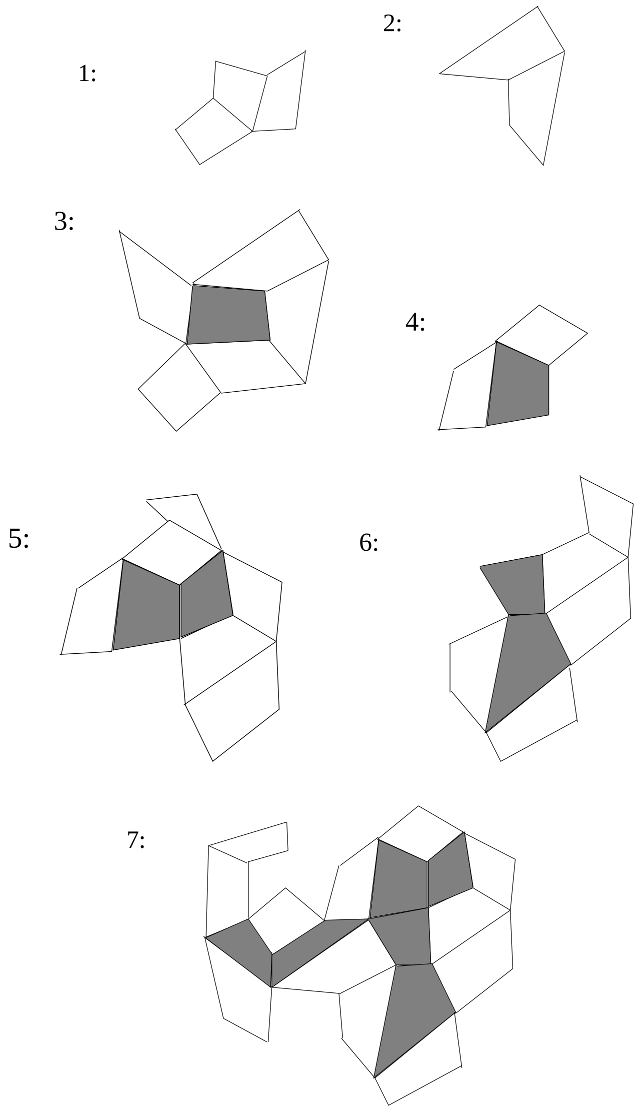

Fig. 12. Supports of B_ξ for the points ξ numbered 1 to 7 in Figure 5.

§5. Bounding the Error of Interpolation

In view of (4.2), it is clear that for every $f \in C(\Omega)$, the spline $s \in \mathcal{S}_3^1(\triangle)$ that interpolates f at the points of P is given by the formula

$$s = \mathcal{I}f := \sum_{\xi \in P} f(\xi)\, B_\xi. \tag{5.1}$$

We can regard $\mathcal{I}$ as a linear operator mapping functions in $C(\Omega)$ into splines in $\mathcal{S}_3^1(\triangle)$. It should be emphasized that we have introduced the cardinal basis splines and this formula purely as a theoretical tool. We do not make use of this representation for computing or storing an interpolating spline — for that purpose, the Bernstein-Bézier representation is much better suited. But it is useful for proving

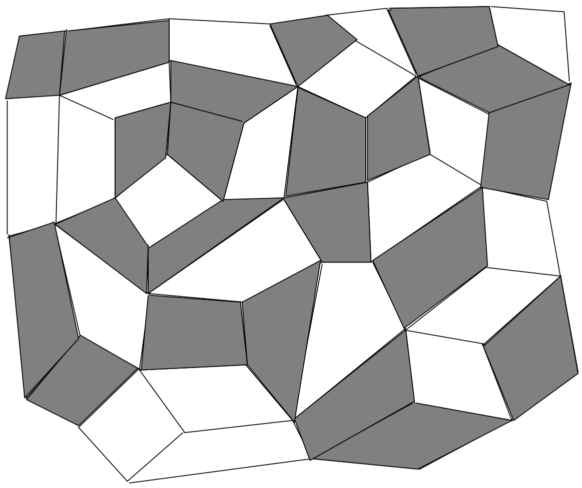

Fig. 13. A $(2,2)^-$ coloring for the quadrangulation from Figure 1.

the following theorem concerning the approximation of functions in the Sobolev space $W_\infty^{m+1}(\Omega)$. Let $|\cdot|_{m+1,\infty}$ be the standard Sobolev seminorm, and let $|\triangle|$ be the maximum of the diameters of the triangles in $\triangle$.

Theorem 5.1. *There exists a constant C depending only on the constants $\alpha_\diamondsuit$ and $\beta_\diamondsuit$ defined in (3.2) such that if f is in the Sobolev space $W_\infty^{m+1}(\Omega)$ with $0 \le m \le 3$,*

$$\|D_x^\alpha D_y^\beta (f - \mathcal{I}f)\|_{\infty,\Omega} \le C\ |\triangle|^{m+1-\alpha-\beta}\ |f|_{m+1,\infty,\Omega}, \tag{5.2}$$

for all $0 \le \alpha + \beta \le m$.

Proof: We apply Theorem 5.1 of [13]. Clearly, $\mathcal{I}p = p$ for all cubic polynomials. The hypothesis (5.3) of that theorem is trivial since $|f(\xi)| \le \|f\|_{T_\xi}$, where T_ξ is the triangle that contains ξ. $\square$

The analog of this error bound also holds for the p-norm, $1 \le p < \infty$. For $p = \infty$, this result can also be established using the Bramble-Hilbert lemma, or by using the weak-interpolation methods described in [8,27].

§6. $(2,2)^-$ Coloring of Quadrangulations

In this section we briefly discuss a class of quadrangulations for which the process of constructing a Lagrange interpolating pair can be somewhat simplified.

Definition 6.1. *A black and white coloring of a quadrangulation $\diamondsuit$ is called a $(2,2)^-$ coloring of $\diamondsuit$ if every quadrilateral of $\diamondsuit$ has at most two neighbors of the same color, and if each chain of quadrilaterals of one color has at most length three. Quadrangulations possessing a $(2,2)^-$ coloring are called $(2,2)^-$ colorable.*

Obviously, the coloring of the quadrangulation $\diamondsuit$ in Figure 1 is not a $(2,2)^-$ coloring of $\diamondsuit$ since there are black and white chains of length four. However, as

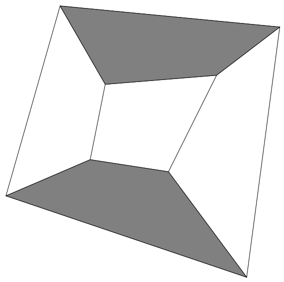

Fig. 14. A quadrangulation, where chains of length three cannot be avoided.

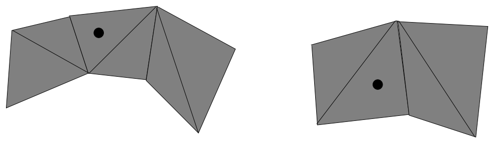

Fig. 15. Interpolation points in the new cases for the simplified algorithm.

shown in Figure 13, a $(2,2)^-$ coloring can easily found for this particular quadrangulation. We observe that many (natural classes of) quadrangulations possess a $(2,2)^-$ coloring.

It is known that there is an algorithm similar to Algorithm 2.1 which can color any given triangulation $\triangle$ with two colors so that no chain of triangles of one color is of length more that two, see [17,27]. However, the situation is more complex for quadrangulations. In this case, chains of length three cannot be avoided in general as can be seen from the example in Figure 14. For further information on $(2,2)^-$ coloring of quadrangulations, see Remark 7.9.

If a quadrangulation $\diamondsuit$ admits a $(2,2)^-$ coloring, then Algorithm 3.2 can be somewhat simplified in that we insert just one diagonal in every black quadrilateral except in the case of closed black chains where we insert both diagonals in one of the quadrilaterals. We then apply Algorithm 3.2 to choose the interpolation points, but with the following simplification of Step 2. For every black component $\mathcal{C}$ which is not a closed chain, choose a quadrilateral Q in $\mathcal{C}$, and add the center point of one of the subtriangles of Q. If the length of $\mathcal{C}$ is three, then this quadrilateral Q should be the second quadrilateral in the chain.

Figure 15 shows the cases for which the triangulation $\triangle$ and the choice of interpolation points is different from that in Section 3. The simplifications which apply for a $(2,2)^-$ coloring are due to the fact that in general it is not necessary to distinguish between even and odd quadrilaterals in dealing with components of one color. Roughly speaking, the locality of the interpolation points inside the chains of one color comes automatically with the coloring.

In general, in this setting we also get a smaller number of interpolation points than with Algorithm 3.2. For instance, consider the quadrangulation $\diamondsuit$ in Figure 13. Then with the coloring shown there, the modified algorithm inserts only one diagonal in 22 quadrilaterals of $\diamondsuit$, and thus the dimension of the corresponding spline space is 187, as compared to 208 as described in Example 4.2.

Using basically the same arguments as in Sections 3 and 4, it is easy to see that the simplified algorithm is stable and produces a Lagrange interpolation pair P and $\mathcal{S}_3^1(\triangle)$ such that the supports of the corresponding basis splines $\{B_\xi\}_{\xi\in P}$ are at most $\operatorname{star}^3(Q_\xi)$ if ξ is strictly inside Q_ξ, and at most $\operatorname{star}^7(Q_\xi)$ if ξ lies on the boundary of Q_ξ. It follows that Theorem 5.1 also holds in this setting.

§7. Remarks

Remark 7.1. A quadrangulation $\diamondsuit$ is said to be a checkerboard quadrangulation provided that its quadrilaterals can be colored black and white in such a way that any two quadrilaterals sharing an edge have the opposite color. An appropriate algorithm (which is significantly simpler than Algorithm 3.2 here) for constructing a Lagrange interpolation pair in this case was provided in [21].

Remark 7.2. A quadrangulation $\diamondsuit$ is said to be separable provided there exists a set $\diamondsuit_0$ of quadrilaterals in $\diamondsuit$ such that for every interior vertex v of $\diamondsuit$, there is a unique quadrilateral $Q \in \diamondsuit_0$ with vertex at v. This case was studied in [22], and the algorithm presented there for constructing a Lagrange interpolation pair in this case is also simpler than Algorithm 3.2. It is easy to see that not all quadrangulations are separable. Figure 1 shows one such example. For an even simpler example, see Figure 3 of [22].

Remark 7.3. The problem of quadrangulating a given set of points is considerably more difficult than the analogous problem of triangulating them, and has only recently been studied in the computational geometry literature [3]. If the number of boundary points is even, there are algorithms which produce some quadrangulation, but it is not guaranteed to be convex. An alternative way to create quadrangulations (which in fact are guaranteed to be convex) is to start with an arbitrary triangulation, and refine it appropriately (by adding new vertices and edges), see [15],

Remark 7.4. We have not assumed that the domain Ω obtained by taking the union of the quadrilaterals in $\diamondsuit$ is simply connected, *i.e.*, Ω may contain holes with polygonal boundaries.

Remark 7.5. It should be clear that starting with a given set of points $\mathcal{V}$ as in Problem 1.1, in general there is more than one way to create an associated Lagrange interpolation pair. Indeed, there are various freedoms in most of the steps outlined above. For example, clearly the coloring depends on the initial coloring and on which quadrilaterals are selected in each step of Algorithm 2.1. Similarly, there is often more than one way to choose the points in the various steps of Algorithm 3.2. Making other choices would lead to different Lagrange interpolation pairs, although all of them would have more or less the same properties.

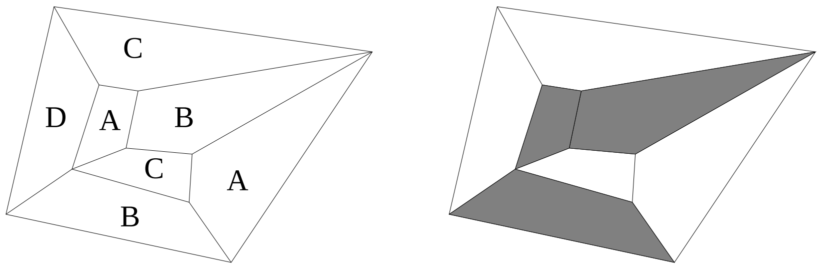

Fig. 16. A quadrangulation which is not 3 colorable but $(2,2)^-$ colorable.

Remark 7.6. In principle, Algorithm 2.1 can be started with any initial coloring (for example color all quadrilaterals of $\diamondsuit$ white). In practice, it is more efficient to choose a (maximal) subset $\diamondsuit^*$ of quadrilaterals from $\diamondsuit$ such that the intersection of any two quadrilaterals from $\diamondsuit^*$ is empty or a single point, and color the quadrilaterals of $\diamondsuit^*$ and $\diamondsuit \setminus \diamondsuit^*$ black and white, respectively. In general, this choice results in short black chains, and Algorithm 2.1 terminates after a few steps. In experiments, we observed that the number of closed black chains is small, even in the case when $\diamondsuit$ consist of thousands of quadrilaterals. Moreover, the treatment of some white boundary quadrilaterals can be simplified by switching their color (after Algorithm 2.1 has stopped).

Remark 7.7. We note that Algorithm 2.1 and the modifications suggested in Remark 7.6 are variants of the coloring method in [17] for bounded degree graphs. In graph coloring theory, a graph is said to be (m,d) colorable if its vertices can be colored with m colors such that each vertex has at most d neighbors of the same color. If $d = 0$, this describes an ordinary coloring of a graph. For instance, the Four Color Theorem (cf. [11]) says that every planar graph is $(4,0)$ colorable. For $d \geq 1$, the coloring is called improper or defective [5]. In this context, Algorithm 2.1 provides a $(2,2)$ coloring of the dual graph of $\diamondsuit$, which is a planar graph of maximal degree four.

Remark 7.8. The coloring in Section 2 leads to triangulations in which about one quarter of the quadrilaterals are split into two triangles, while the rest are split into four triangles. In our experiments with $(2,2)^-$ colorable quadrangulations, we found that about one-half of the quadrilaterals are split into two triangles, while the rest are split into four triangles.

Remark 7.9. In Section 6 we have shown that if a quadrangulation can be colored such that each chain of quadrilaterals of one color has at most length three, then the process of constructing a Lagrange interpolating pair can be simplified. Our experience is that many quadrangulations possess a $(2,2)^-$ coloring. On the other hand, no automatic algorithm of linear complexity is known which yields a $(2,2)^-$ coloring for quadrangulations. A natural idea would be to color a given quadrangulation with three colors, and then afterwards switch the quadrilaterals with the third color to either black or white. However, not every quadrangulation can be colored with three colors [18], which can be seen from the example in Figure 16 (left) where the letters symbolize the four colors used. On the other hand, Figure 16 (right) shows that this quadrangulation does possess a $(2,2)^-$ coloring. We note that there are only a few papers on $(m,d)^-$ coloring, see [4,30].

Remark 7.10. Using C^1 cubic splines, it is considerably more difficult to construct a local interpolation method based on Lagrange data than it is for Hermite data. Indeed, given Hermite data at the vertices of a quadrangulation $\diamondsuit$ (along with normal derivatives at the midpoints of the edges of $\diamondsuit$), then we can apply the well-known macro-element methods based on the triangulation $\diamondplus$ obtained by adding both diagonals to each quadrilateral, see [10,12,28]. But of course, to use these methods one has to find accurate estimates for the derivatives at all points where they are not given. We also note that the method described in this paper can be easily modified to produce a method which would make use of Hermite data (function values and gradients) at each vertex of $\diamondsuit$.

Remark 7.11. A univariate cubic polynomial p on an interval $[a,b]$ which satisfies $p(a) = z_0$ and $p(b) = z_3$ can be written in the Bernstein-Bézier form

$$p(x) = \frac{1}{h^3}[z_0(b-x)^3 + 3c_1(x-a)(b-x)^2 + 3c_2(x-a)^2(b-x) + z_3(x-a)^3],$$

where $h = b - a$. Then the coefficients c_1 and c_2 of the unique p that interpolates given values at the points $t_1 := a + h/3$ and $t_2 := a + 2h/3$, can be determined by solving a 2×2 linear system whose matrix is

$$\frac{2}{9}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$

independent of the interval $[a,b]$. Moreover, if we are given c_1, then we can make p interpolate a prescribed value z_2 at t_2 by simply setting

$$c_2 = \frac{27z_2 - z_0 - 6c_1 - 8z_3}{12}. \tag{7.1}$$

Remark 7.12. Suppose s is a cubic polynomial, and that for a given triangle T we know all of its B-coefficients except for the one associated with the domain point ξ^T_{111}. Then since $B^T_{111}(\xi^T_{111}) = 2/9$, we can immediately calculate c^T_{111}. Clearly, this is a stable computation independent of the shape of T.

Remark 7.13. While not directly applicable to interpolation of scattered data, Lagrange interpolating pairs can still be a useful tool for fitting scattered data as a second stage in a two-stage method (cf. [29] for a discussion of general two-stage methods). Here the first stage would involve interpolation of scattered data by a linear spline s_1 over a triangulation $\triangle_1$ of the data points. Then the second stage would involve interpolation of s_1 by a C^1 cubic spline s on a much coarser triangulation $\triangle$, based on samples of s_1 at the points P. In the case when data comes from a C^4 function, the approximation error of s_1 is $\mathcal{O}(h_1^2)$, where h_1 is the maximal diameter of the triangles in $\triangle_1$, and the approximation error of s is $\max\{\mathcal{O}(h_1^2), \mathcal{O}(h^4)\}$, where h is the maximal diameter of the triangles in $\triangle$. If $\triangle_1$ is much finer than $\triangle$, then the error of s would be $\mathcal{O}(h^4)$, thereby providing an efficient compression of the data (cf. [22,26,27]).

Remark 7.14. We have shown that the spline spaces $\mathcal{S}_3^1(\triangle)$ constructed here have full approximation power. This should be contrasted with the situation for the triangulation obtained from a given quadrangulation by inserting one diagonal in each quadrilateral, where it is known [1] that the associated spaces of C^1 cubic splines do not have full approximation power.

Remark 7.15. It is easy to see that the angles of a quadrangulation can be well-behaved in the sense that the largest angle is bounded by a constant $\kappa < \pi$, but the smallest angle $\theta_\triangle$ in the associated triangulation can still be arbitrarily small. However, conversely, clearly $2\theta_\triangle \le \kappa \le \pi - 2\theta_\triangle$.

Remark 7.16. The interpolation method described in this paper is easy to implement, and due to its low complexity, can be used on very large data sets. For some numerical experiments (based on separable quadrangulations) involving both synthetic and real world data, see [21,22].

References

1. Boor, C. de and K. Höllig, Approximation order from bivariate C^1-cubics: a counterexample, Proc. Amer. Math. Soc. **87** (1983), 649–655.
2. Boor, C. de, B-form basics, in *Geometric Modeling*, G. Farin, (ed), SIAM, Philadelphia, 1987, 131–148.
3. Bose, P. and G. T. Toussaint, Characterizing and efficiently computing quadrangulations of planar point sets, Comput. Aided Geom. Design **14** (1997), 763–785.
4. Chartrand, G., P. Geller, and S. Hedetniemi, A generalization of the chromatic number, Proc. Camb. Phil. Soc. **64** (1968), 265–271.
5. Cowen, L., W. Goddard, and C. E. Jesurum, Coloring with defect, in *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*, 1997, 548–557.
6. Davydov, O. and G. Nürnberger, Interpolation by C^1 splines of degree $q \ge 4$ on triangulations, J. Comput. Appl. Math. **126** (2000), 159–183.
7. Davydov, O., G. Nürnberger, and F. Zeilfelder, Cubic spline interpolation on nested polygon triangulations, in *Curve and Surface Fitting*, A. Cohen, C. Rabut, and L. L. Schumaker (eds), Vanderbilt University Press, Nashville, 2000, 161–170.
8. Davydov, O., G. Nürnberger, and F. Zeilfelder, Bivariate spline interpolation with optimal approximation order, Constr. Approx. **17** (2001), 181–208.
9. Farin, G., Triangular Bernstein-Bézier patches, Comput. Aided Geom. Design **3** (1986), 83–127.
10. Fraeijs de Veubeke, B., A conforming finite element for plate bending, J. Solids Structures **4** (1968), 95–108.
11. Jensen, T. R. and B. Toft, *Graph Coloring Problems,* Wiley, New York, 1995.

12. Lai, M.-J., Scattered data interpolation and approximation using bivariate C^1 piecewise cubic polynomials, Comput. Aided Geom. Design **13** (1996), 81–88.

13. Lai, M.-J. and L. L. Schumaker, On the approximation power of bivariate splines, Advances in Comp. Math. **9** (1998), 251–279.

14. Lai, M.-J. and L. L. Schumaker, Scattered data interpolation using C^2 super-splines of degree six, SIAM J. Numer. Anal. **36** (1999), 143–159.

15. Lai, M.-J. and L. L. Schumaker, On the approximation power of splines on triangulated quadrangulations, SIAM J. Numer. Anal. **36** (1999), 143–159.

16. Lai, M.-J. and L. L. Schumaker, Quadrilateral macro-elements, SIAM J. Math. Anal. **33** (2002), 1107–1116.

17. Lovàsz, L., On decomposition of graphs, Studia Sci. Math. Hungar. **1** (1966), 237–238.

18. Mabry, R., Louisiana State University Shreveport, U.S.A, private communication.

19. Nürnberger, G. and Th. Rießinger, Lagrange and Hermite interpolation by bivariate splines, Numer. Func. Anal. Optim. **13** (1992), 75–96.

20. Nürnberger, G. and Th. Rießinger, Bivariate spline interpolation at grid points, Numer. Math. **71** (1995), 91–119.

21. Nürnberger, G., L. L. Schumaker, and F. Zeilfelder, Local Lagrange interpolation by bivariate C^1 cubic splines, in *Mathematical Methods for Curves and Surfaces III, Oslo, 2000,* T. Lyche and L. L. Schumaker (eds), Vanderbilt University Press, Nashville, 2000, 393–404.

22. Nürnberger, G., L. L. Schumaker, and F. Zeilfelder, Lagrange interpolation by C^1 cubic splines on triangulations of separable quadrangulations, in *Approximation Theory X: Splines, Wavelets, and Applications*, C. K. Chui, L. L. Schumaker, and J. Stöckler (eds), Vanderbilt University Press, Nashville, 2002, 405–424.

23. Nürnberger, G. and F. Zeilfelder, Interpolation by spline spaces on classes of triangulations, J. Comput. Appl. Math. **119** (2000), 347–376.

24. Nürnberger, G. and F. Zeilfelder, Developments in bivariate spline interpolation, J. Comput. Appl. Math. **121** (2000), 125–152.

25. Nürnberger, G. and F. Zeilfelder, Local Lagrange interpolation by cubic splines on a class of triangulations, in *Trends in Approximation Theory*, K. Kopotun, T. Lyche, and M. Neamtu (eds), Vanderbilt University Press, Nashville, 2001, 341-350.

26. Nürnberger, G. and F. Zeilfelder, Local Lagrange interpolation on Powell-Sabin triangulations and terrain modelling, in *Recent Progress in Multivariate Approximation*, W. Haußmann, K. Jetter, and M. Reimer (eds), Birkhäuser ISNM 137, Basel, 2001, 227–244.

27. Nürnberger, G. and F. Zeilfelder, Lagrange interpolation by bivariate C^1 splines with optimal approximation order, Advances in Comp. Math., to appear.
28. Sander, G., Bornes supérieures et inférieures dans l'analyse matricielle des plaques en flexion-torsion, Bull. Soc. Royale Sciences Liège **33** (1964), 456–494.
29. Schumaker, L.L., Fitting surfaces to scattered data, in *Approximation Theory II*, G.G. Lorentz, et. al. (eds), 1976, 203–268.
30. Woodall, D., Improper colourings of graphs, in *Graph Colourings*, R. Nelson and R.J. Wilson, (eds), Longman Scientific and Technical (Harlow), 1990, 45–63.