On the approximation order of triangular Shepard interpolation

Francesco Dell’Accio · Filomena Di Tommaso · Kai Hormann

Abstract

Shepard’s method is a well-known technique for interpolating large sets of scattered data. The classical Shepard operator reconstructs an unknown function as a normalized blend of the function values at the scattered points, using the inverse distances to the scattered points as weight functions. Based on the general idea of defining interpolants by convex combinations, Little suggests to extend the bivariate Shepard operator in two ways. On the one hand, he considers a triangulation of the scattered points and substitutes function values with linear polynomials which locally interpolate the given data at the vertices of each triangle. On the other hand, he modifies the classical point-based weight functions and defines instead a normalized blend of the locally interpolating polynomials with triangle-based weight functions which depend on the product of inverse distances to the three vertices of the corresponding triangle. The resulting triangular Shepard operator interpolates all data required for its definition and reproduces polynomials up to degree 1, whereas the classical Shepard operator reproduces only constants. In this paper we show that this interpolation operator consequentially has quadratic approximation order, which is confirmed by our numerical results.

1 Introduction

Shepard’s method [19] is a non-polynomial global interpolation scheme for large sets of scattered data. While this technique works in any dimension, we focus on the case of bivariate interpolation. To this end, let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of $n$ distinct points in $\mathbb{R}^2$, called nodes or sample points, with associated data $f_i$ sampled from some unknown function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, that is, $f_i = f(x_i)$, $i = 1, \ldots, n$. The classical Shepard interpolation formula [19]

$$S_\mu [f](x) = \sum_{i=1}^{n} A_{\mu,i}(x) f_i \quad (1.1)$$

uses point-based basis functions

$$A_{\mu,i}(x) = \frac{1}{\sum_{k=1}^{n} \frac{1}{|x - x_k|^\mu}} \cdot \frac{1}{|x - x_i|^\mu}, \quad i = 1, \ldots, n \quad (1.2)$$

in barycentric form, where $|\cdot|$ denotes the Euclidean norm and $\mu > 0$ is a parameter that controls the range of influence of the data values. A common choice is to take $\mu = 2$, so that the basis functions are rational and infinitely differentiable [4].

Since the basis functions $A_{\mu,i}$ are cardinal, non-negative, and form a partition of unity, the interpolation operator $S_\mu$ is stable [10] in the sense that

$$\min_i f_i \leq S_\mu [f](x) \leq \max_i f_i, \quad x \in \mathbb{R}^2,$$

but for $\mu > 1$ it has flat spots at all nodes. Moreover, the algebraic degree of exactness (abbreviated by “dex” in the following) of the operator $S_\mu$ is 0, that is, it reproduces only constant polynomials, and its approximation order is at most $O(h)$, where $h$ is the mesh size of the set of sample points [10].

To avoid these drawbacks, Shepard [19] suggests a modification of the operator (1.1) where the function values $f_i$ are replaced by the first-order Taylor polynomial of $f$ at $x_i$. Starting from this idea, several other modifications have been proposed (see [5, 21] and the references therein). These so-called modified (or
combined) Shepard operators substitute the function values in (1.1) by local interpolation operators \(P[\cdot, x_t]\) applied to \(f\), giving
\[
S_p[f](x) = \sum_{i=1}^{n} A_{\mu,i}(x) P[f, x_i](x).
\]
Letting \(d_i = \text{dex}(P[\cdot, x_i]) > 0\), a combined Shepard operator increases the algebraic degree of exactness of the original operator (1.1) to \(d = \min_i d_i\). In the bivariate setting, a recent approach [6, 7, 8] is to associate with each sample point \(x_i \in X\) a triangle \(t_i = [x_i, y_i, z_i]\) and to take as \(P[f, x_i]\) a local polynomial interpolant to the data at the vertices of \(t_i\). The vertices \(y_i, z_i \in X\) are close to \(x_i\) and chosen to locally reduce the error of \(P[f, x_i]\). In fact, an explicit expression for the remainder of \(P[f, x_i]\) is known in the case of sufficiently differentiable functions [7]. This expression results from adding to the classical Taylor remainder a polynomial term, which is the difference between \(P[f, x_i]\) and the Taylor expansion \(T[f, x_i]\) of the same degree. Bounding the remainder then leads to practical criteria for associating a triangle to each interpolation point. Several combinations have been introduced depending on the information we have at each node and simplest lacunary cases have been covered. When only function values are given, the polynomial \(P[f, x_i]\) is linear and \(\text{dex}(P[\cdot, x_i]) = 1\), but an asymmetry results, as \(P[f, x_i]\) is weighted only by \(A_{\mu,i}\), which corresponds to \(x_i\), while the weight functions associated with the other vertices of \(t_i\) are not used. Little [14] introduces an extension of the classical Shepard operator which breaks this asymmetry and provides an interpolant with linear precision without using any derivative data.

The goal of this paper is to investigate the approximation order of Little’s triangular Shepard operator and point out other favourable properties. The definition of this operator is based on a triangulation of the nodes and an extension of Shepard’s point-based basis functions (1.2) to triangle-based basis functions (see Section 2). The latter are then used in combination with linear polynomials that locally interpolate the given data at the vertices of each triangle (see Section 3), and we prove that this construction yields an interpolation operator with quadratic approximation order (see Section 4). We further study the cardinal basis functions (see Section 5) and finally provide numerical results (see Section 6), which confirm the order of approximation as well as the good approximation accuracy of this operator. In particular, we consider two kinds of triangulations: the Delaunay triangulation of the nodes and a compact triangulation with about 65% less triangles, which yields a more efficient interpolation operator with comparable approximation accuracy.

2 Triangle-based basis functions

To extend the point-based basis functions in (1.2) to triangle-based basis functions, let us consider a triangulation \(T = \{t_1, t_2, \ldots, t_m\}\) of the nodes \(X\). That is, each \(t_j = [x_{j_1}, x_{j_2}, x_{j_3}]\) is a triangle with vertices in \(X\) and each \(x_j \in X\) is the vertex of at least one triangle, hence
\[
\bigcup_{j=1}^{m} \{j_1, j_2, j_3\} = \{1, 2, \ldots, n\}. \tag{2.1}
\]
For example, \(T\) can be the Delaunay triangulation [9] of \(X\), but we also allow for general triangulations with overlapping or disjoint triangles (see Figure 1).

The triangle-based basis functions with respect to the triangulation \(T\) are then defined by
\[
B_{\mu,j}(x) = \frac{\prod_{j=1}^{3} 1}{\prod_{k=1}^{m} \frac{1}{|x - x_{j_k}|^\mu}}, \quad j = 1, \ldots, m, \tag{2.2}
\]
where $\mu > 0$ is again a control parameter. Like Shepard’s basis functions, the triangle-based basis functions (2.2) are non-negative and form a partition of unity, but instead of being cardinal they satisfy the following properties.

**Proposition 2.1.** The triangle-based basis function in (2.2) and its gradient (that exists for $\mu > 1$) vanish at all nodes $x_i \in X$ that are not a vertex of the corresponding triangle $t_j$. That is,

$$B_{\mu,j}(x_i) = 0, \quad \nabla B_{\mu,j}(x_i) = 0, \quad \mu > 1,$$

for any $j = 1, \ldots, m$ and $i \notin \{j_1, j_2, j_3\}$.

**Proof.** If we multiply both the numerator and the denominator of (2.2) with $|x - x_i|^{\mu}$, then

$$B_{\mu,j}(x) = \frac{C_j(x)}{\sum_{k=1}^{m} C_k(x)},$$

where

$$C_k(x) = |x - x_i|^\mu \prod_{\ell=1}^{3} \frac{1}{|x - x_{i_\ell}|^\mu}, \quad k = 1, \ldots, m.$$  

Let us denote by $J_i$ the set of indices of all triangles which have $x_i$ as a vertex,

$$J_i = \{ k \in \{1, \ldots, m \} : i \in \{k_1, k_2, k_3\} \},$$

and note that $J_i \neq \emptyset$ by (2.1). Then, $C_k(x_i) = 0$ if and only if $k \notin J_i$, and (2.3) follows because $j \notin J_i$. Moreover, if $\mu > 1$, then $C_k(x)$ is differentiable at $x_i$, and (2.4) follows because

$$\nabla B_{\mu,j}(x) = \frac{\nabla C_j(x) \sum_{k=1}^{m} C_k(x) + C_j(x) \sum_{k=1}^{m} \nabla C_k(x)}{(\sum_{k=1}^{m} C_k(x))^2}$$

and $\nabla C_j(x_i) = 0$ for $i \notin \{j_1, j_2, j_3\}$.  

As an immediate consequence of Proposition 2.1 and the partition of unity property we have, for each $i = 1, \ldots, n$,

$$\sum_{j \in J_i} B_{\mu,j}(x_i) = 1, \quad \sum_{j \notin J_i} \nabla B_{\mu,j}(x_i) = 0, \quad \mu > 1$$

where $J_i$ is the index set from (2.5).

Figures 2 and 3 show examples of triangle-based basis functions for $\mu = 2$. Function values close to 0 are marked by red and the ten colours from green via blue to pink correspond to the ten uniform intervals of function values between 0 and 1. As predicted by Proposition 2.1, each basis function has local minima with function value 0 at all nodes, except at the vertices $x_i, i \in \{j_1, j_2, j_3\}$ of the corresponding triangle $t_j$. According to (2.6), if $J_i = \{j\}$, so that $t_j$ is the only triangle adjacent to $x_i$, then the basis function has a local maximum with function value 1 at $x_i$ (see Figure 3).
The second ingredient for Little's extension of the Shepard operator [14] are the linear polynomials that locally interpolate the given data at the vertices of each of the triangles. For \( t_j \in T \), this polynomial \( L_j : \mathbb{R}^2 \to \mathbb{R} \) can be written as

\[
L_j(x) = \sum_{i=1}^{3} \lambda_{j,i}(x)f_{j,i},
\]

where \( \lambda_{j,i}(x), \ell = 1, 2, 3 \) are the barycentric coordinates \([17]\) of \( x \) with respect to the triangle \( t_j = [x_{j_1}, x_{j_2}, x_{j_3}] \), that is,

\[
\lambda_{j,i}(x) = \frac{A(x, x_{j_i}, x_{j_j})}{A(x_{j_1}, x_{j_2}, x_{j_3})}, \quad \lambda_{j,i}(x) = \frac{A(x_{j_i}, x_{j_k}, x_{j_l})}{A(x_{j_1}, x_{j_2}, x_{j_3})}, \quad \lambda_{j,i}(x) = \frac{A(x_{j_i}, x_{j_j}, x)}{A(x_{j_1}, x_{j_2}, x_{j_3})},
\]

with \( A(x, y, z) \) denoting the signed area of the triangle \([x, y, z]\). In general, \( \lambda_{j,i}(x) \) with \( j \in J_i \) and \( i \in \{j_1, j_2, j_3\} \) is the unique linear polynomial with \( \lambda_{j,i}(x_j) = 1 \) and \( \lambda_{j,i}(x) = 0 \) for \( x \) on the line defined by the edge opposite \( x_i \) in the triangle \( t_j \).

To study the approximation order of the interpolating polynomial (3.1), we let \( \Omega \subseteq \mathbb{R}^2 \) be a non-empty, compact, and convex domain with \( X \subseteq \Omega \) (e.g. the convex hull of \( X \)). Following Farwig [10], we further consider the class \( C^{1,1}(\Omega) \) of differentiable functions \( f : \Omega \to \mathbb{R} \) whose partial derivatives are Lipschitz-continuous of order 1, equipped with the seminorm

\[
\|f\|_{1,1} = \sup \left\{ \frac{\frac{\partial^2 f}{\partial x_1^2}(u) - \frac{\partial^2 f}{\partial x_2^2}(v)}{|u - v|} : u, v \in \Omega, u \neq v, \alpha \in \{0, 1\} \right\}.
\]

Moreover, we denote the edge vectors of the triangle \( t_j \) by \( e_{j_1} = x_{j_1} - x_{j_2}, e_{j_2} = x_{j_2} - x_{j_3}, \) and \( e_{j_3} = x_{j_3} - x_{j_1} \).

**Proposition 3.1.** Let \( \Omega \) be a compact, convex domain which contains \( X \) and \( f \in C^{1,1}(\Omega) \). Then,

\[
|f(x) - L_j(x)| \leq \|f\|_{1,1}(2|x - x_j|^2 + 4 h_j C_j |x - x_j|)
\]

for any \( x \in \Omega \), with \( h_j = \max(|e_{j_1}|, |e_{j_2}|, |e_{j_3}|) \) and \( C_j \) a constant which depends only on the shape of \( t_j \).

**Proof.** Let us consider the first-order Taylor expansion of \( f(x_{j_1}) \) and \( f(x_{j_2}) \) at \( x_{j_3} \) with integral remainder [1],

\[
f(x_{j_1}) = f(x_{j_3}) + e_{j_1} \cdot \nabla f(x_{j_3}) + |e_{j_1}|^2 \int_0^1 \frac{\partial^2 f(x_{j_3} + te_{j_1})}{\partial y_j^2} (1-t) \, dt,
\]

\[
f(x_{j_2}) = f(x_{j_3}) + e_{j_2} \cdot \nabla f(x_{j_3}) + |e_{j_2}|^2 \int_0^1 \frac{\partial^2 f(x_{j_3} + te_{j_2})}{\partial y_j^2} (1-t) \, dt,
\]

where \( \partial^2 / \partial y_j^2 \) are the second-order directional derivatives along the unit vectors \( v_{j_i} = e_{j_i}/|e_{j_i}| \). Substitut-
While the first term in (3.5) is bounded as usual \[|f(x) - T_1[f, x_{j_h}](x)| + |T_1[f, x_{j_h}](x) - L_j(x)|.\] By the triangle inequality,

\[|f(x) - L_j(x)| \leq |f(x) - T_1[f, x_{j_h}](x)| + |T_1[f, x_{j_h}](x) - L_j(x)|.
\]

(3.5)

While the first term in (3.5) is bounded as usual [10] by

\[|f(x) - T_1[f, x_{j_h}](x)| \leq 2\|f\|_{1,1} |x - x_{j_h}|^2,
\]

the second term is bounded by

\[|T_1[f, x_{j_h}](x) - L_j(x)| = |\delta_j(x)| \leq 4\|f\|_{1,1} h_j C_j |x - x_{j_h}|.
\]

In fact,

\[|\lambda_{j, j_h}(x)| \leq \frac{h_j}{|A(x_{j_h}, x_j, x_{j_h})|} |x - x_{j_h}|, \quad x \in \Omega
\]

for \(k = 2, 3\), and by writing \(\partial^2 f / \partial \nu^2\) as the limit of difference quotients of the functions \(\partial f / \partial \nu\) as well as (3.2), we get

\[\left|\frac{\partial^2 f}{\partial \nu^2}(u) - \frac{\partial f}{\partial \nu}(v)\right| \leq 2\|f\|_{1,1}, \quad u, v \in \Omega, \quad u \neq v.
\]

Finally, (3.3) follows by setting

\[C_j = \frac{h_j^2}{|A(x_{j_h}, x_j, x_{j_h})|} = \frac{\sin(\alpha_j + \beta_j)}{\sin \alpha_j \sin \beta_j} = \cot(\alpha_j) + \cot(\beta_j),
\]

where \(\alpha_j\) and \(\beta_j\) are the angles adjacent to the longest edge of \(t_j\) (see Figure 4). Note that \(C_j\) is large for small angles \(\alpha_j\) and \(\beta_j\), hence it is advantageous to use as \(T\) the Delaunay triangulation of \(X\), because it maximizes the smallest angle. Moreover, the bound in (3.3) also holds if we replace the reference vertex \(x_{j_h}\) by any of the other two vertices \(x_{j_h}\) and \(x_{j_h}\) of \(t_j\).

4 Triangular Shepard operator

For any \(\mu > 0\) the triangular Shepard operator is defined by

\[K_\mu[f](x) = \sum_{j=1}^m B_{\mu, j}(x) L_j(x),
\]

where \(L_j(x)\) is the linear interpolating polynomial (3.1) over triangle \(t_j\) and \(B_{\mu, j}(x)\) is the corresponding triangle-based basis function from (2.2). For the special case \(\mu = 2\), this operator was proposed by Little [14] and he noticed the following properties.
Proposition 4.1. The operator $K_\mu$ is an interpolation operator, that is,
\[ K_\mu[f](x_i) = f_i, \quad i = 1, \ldots, n, \]
and $\text{dex}(K_\mu) = 1$.

Proof. If $x_i$ is a vertex of triangle $t_j$ (i.e., $i \in \{j_1, j_2, j_3\}$), then $L_j(x_i) = f_i$ by (3.1), otherwise $B_{\mu,j}(x_i) = 0$ by Proposition 2.1. Using (2.6) we then have
\[ K_\mu[f](x_i) = \sum_{j=1}^{m} B_{\mu,j}(x_i) L_j(x_i) = \sum_{j \in \mathcal{J}} B_{\mu,j}(x_i) f_i = f_i. \]
Moreover, $K_\mu$ reproduces polynomials up to degree 1, because $\text{dex}(L_j) = 1$ for $j = 1, \ldots, m$ by construction and because the basis functions $B_{\mu,j}$ are a partition of unity.

Let us now turn to the approximation order of the operator $K_\mu$, which was not studied by Little [14]. To this purpose, we follow Farwig [10] and let $\| \cdot \|$ be the maximum norm and $R_r(y) = \{ x \in \mathbb{R}^2 : \| x - y \| \leq r \}$ be the axis-aligned closed square with centre $y$ and edge length $2r$. With $V(t)$ denoting the set of vertices of a triangle $t \in \mathcal{T}$, we then define
\[ h' = \inf \{ r > 0 : \forall x \in \Omega \exists t \in \mathcal{T} : R_r(x) \cap V(t) \neq \emptyset \} \tag{4.2} \]
and
\[ h'' = \inf \{ r > 0 : \forall t \in \mathcal{T} \exists x \in \Omega : t \subset R_r(x) \}, \]
and finally
\[ h = \max \{ h', h'' \}. \tag{4.3} \]
A small value of $h'$ corresponds to a rather uniform triangle distribution, but does not exclude the presence of large triangles (see Figure 5). The latter cannot occur if $h''$ and then also $h$ are small, because each triangle is contained in a square with edge length $2h$. Note that in the maximum norm, the length of each triangle edge does not exceed $2h$. We further let
\[ M = \sup_{x \in \Omega} \# \{ t \in \mathcal{T} : R_h(x) \cap V(t) \neq \emptyset \}, \tag{4.4} \]
where $\#$ is the cardinality operator, be the maximum number of triangles with at least one vertex in some square with edge length $2h$. Small values of $M$ imply that there are no clusters of triangles.

Theorem 4.2. Let $\Omega$ be a compact convex domain which contains $X$, $f \in C^{1,1}(\Omega)$, and $\mu > 4/3$. Then,
\[ |f(x) - K_\mu[f](x)| \leq C M \| f \|_{1,1,2} h^2 \]
for any $x \in \Omega$, with $C$ a positive constant which depends on $T$ and $\mu$.

Proof. For $y = (y_1, y_2) \in \mathbb{R}^2$ let
\[ Q_r(y) = \{ x = (x_1, x_2) \in \mathbb{R}^2 : y_k - r < x_k \leq y_k + r, k = 1, 2 \} \]
be the axis-aligned half-open square with centre $y$ and edge length $2r$. Now let $x \in \Omega$ be fixed and consider

Figure 5: The value $h'$ is small for this triangulation, but $h''$ is large because of the indicated large triangle.
the following disjoint covering of $\Omega$. For $k \in \mathbb{N}_0$ we define by

$$U_k = \bigcup_{\nu \in \mathbb{Z}^2, \|\nu\| = k} Q_0(x + 2\nu)$$

the half-open annulus with centre $x$, radius $2k\,h$, and width $h$. For example, $U_0 = Q_0(x)$ and $U_1$ is the union of the eight congruent half-open squares surrounding $U_0$ (see Figure 6). Since $\Omega$ is compact, there exists some $N \in \mathbb{N}$, independent of $x$ and of order $O(1/h)$, such that

$$\Omega \subset \bigcup_{k=0}^N U_k.$$

Noticing that $U_k$ is composed of $8k$ congruent copies of $Q_0(x)$, the number of triangles with at least one vertex in $U_k$ is bounded,

$$\#\{t \in T : U_k \cap V(t) \neq \emptyset\} \leq 8kM, \quad k = 1, \ldots, N.$$  \hfill (4.5)

For any triangle $t$ with at least one vertex in $U_k$ one of the following cases (see Figure 7) holds:

1. $V(t) \cap U_{k-1} \neq \emptyset \Rightarrow (2k-3)h \leq \|x - v\| \leq (2k+1)h \quad \forall v \in V(t);$  
2. $V(t) \subset U_k \Rightarrow (2k-1)h \leq \|x - v\| \leq (2k+1)h \quad \forall v \in V(t);$  
3. $V(t) \cap U_{k+1} \neq \emptyset \Rightarrow (2k-1)h \leq \|x - v\| \leq (2k+3)h \quad \forall v \in V(t).$  \hfill (4.6)

Let us now denote by $T_0$ the set of all triangles with at least one vertex in $U_0$. By the definitions of $h'$ in (4.2) and $M$ in (4.4), this set contains at least one and at most $M$ triangles and for each triangle $t_j = [x_{j_1}, x_{j_2}, x_{j_3}] \in T_0$ we have

$$\|x - x_{j_1}\| \|x - x_{j_2}\| \|x - x_{j_3}\| \leq h \cdot (3h) \cdot (3h) = 9h^3,$$  \hfill (4.7)

because one vertex of $t_j$ is inside $U_0$ and the other two are in $U_0 \cup U_1$. For $k = 1, \ldots, N$ we further denote by $T_k$ the set of all triangles with at least one vertex in $U_k$ and no vertex in $U_{k-1}$. By (4.5), this set contains at
most $8kM$ triangles and by case 3 in (4.6) we have
\[(2k - 1)h^3 \leq \|x - x_{j}\| \|x - x_{j}\| \|x - x_{j}\| \leq (2k + 3)h^3\]  
(4.8)
for each triangle $t_j = [x_{j}, x_{j}, x_{j}] \in T_k$. Further note that by construction,
\[
\bigcup_{k=0}^{N} T_k = T \quad \text{and} \quad \bigcap_{k=0}^{N} T_k = \emptyset.
\]
Let us now turn to the approximation error
\[e(x) = |f(x) - K_n[f](x)|\]
of the triangular Shepard interpolant at $x$. By (4.1) and the fact that the basis function $B_{n,j}$ are non-negative and form a partition of unity,
\[e(x) = \left| \sum_{j=1}^{m} B_{n,j}(x)f(x) - \sum_{j=1}^{m} B_{n,j}(x)L_j(x) \right| \leq \sum_{j=1}^{m} |f(x) - L_j(x)|B_{n,j}(x).
\]
Using Proposition 3.1 and (2.2) we then get
\[e(x) \leq \|f\|_1 \sum_{j=1}^{m} \left\{ \sum_{t=1}^{n} (2\|x - x_{j}\|^2 + 4h_j C_j \|x - x_{j}\|) \right\} \prod_{k=1}^{3} \prod_{t=1}^{3} \|x - x_{k}\|^\mu \frac{1}{\sum_{k=1}^{3} \prod_{t=1}^{3} \|x - x_{k}\|^\mu},
\]
\[\leq C'\|f\|_1 \sum_{j=1}^{m} \left\{ \sum_{t=1}^{n} (2\|x - x_{j}\|^2 + 4h_j C_j \|x - x_{j}\|) \right\} \prod_{k=1}^{3} \prod_{t=1}^{3} \|x - x_{k}\|^\mu \frac{1}{\sum_{k=1}^{3} \prod_{t=1}^{3} \|x - x_{k}\|^\mu},
\]
where $C' = \sqrt[3]{3\mu}$ is a constant which depends on the fact that we bound the Euclidean norm with the maximum norm.

Now let $t_j \in T$ be a triangle such that
\[\frac{1}{3} \prod_{t=1}^{3} \|x - x_{t}\| = \min_{j=1,\ldots,m} \prod_{t=1}^{3} \|x - x_{t}\|.
\]
Since at least one triangle of $T$ belongs to $T_0$, we know from (4.7) that
\[\prod_{t=1}^{3} \|x - x_{t}\| \leq 9h^3.
\]
For each $t_j \in T_0$ we then have
\[\prod_{t=1}^{3} \|x - x_{t}\| \leq \frac{9h^3}{(2k - 1)h^3} = \frac{9}{(2k - 1)^3}.
\]
Therefore,
\[\frac{1}{3} \prod_{t=1}^{3} \|x - x_{t}\| \leq \frac{9h^3}{(2k - 1)h^3} \leq \frac{1}{(2k - 1)^3}, \quad \text{if} \quad t_j \in T_0,
\]
\[\frac{1}{3} \prod_{t=1}^{3} \|x - x_{t}\| \leq \frac{9h^3}{(2k - 1)h^3} \leq \frac{1}{(2k - 1)^3}, \quad \text{if} \quad t_j \in T_0.
\]
Further assuming without loss of generality that $x_j = x_k$ for $t_j \in T_k$, so that $\|x - x_{j}\| \leq h$ for each $t_j \in T_0$ and $\|x - x_{j}\| \leq (2k + 1)h$ for each $t_j \in T_k$, $k = 1,\ldots,N$, and taking into account that $h_j \leq \sqrt{3}h \leq 3h$, we get
\[e(x) \leq C'\|f\|_1 \left( \sum_{t_j \in T_0} (2h^2 + 4h_j C_j h) + \sum_{k=1}^{N} \sum_{t_j \in T_k} (2(2k + 1)^2 h^2 + 4h_j C_j (2k + 1)h) \right) \frac{9\mu}{(2k - 1)^3} h^2,
\]
\[\leq C'\|f\|_1 \left( \sum_{t_j \in T_0} (2 + 12C) + 9 \sum_{k=1}^{N} \sum_{t_j \in T_k} \frac{2(2k + 1)^2 h^2 + 12(2k + 1)C'}{(2k - 1)^3} \right) h^2.
\]
where $C'' = \max\{C_1, C_2, \ldots, C_m\}$. Using (4.5) we then have

$$e(x) \leq C'M\|f\|_{1, 1} \left( 2 + 12C'' \right) + 9\mu \sum_{k=1}^{N} \frac{8k^2(2k+1)^2 + 12(2k+1)C''}{(2k-1)^{3\mu}} h^2$$

$$= C'M\|f\|_{1, 1} \left( 2 + 12C'' + 16 \cdot 9\mu \sum_{k=1}^{N} \frac{k(2k+1)^2}{(2k-1)^{3\mu}} + 96 \cdot 9\mu C'' \sum_{k=1}^{N} \frac{k(2k+1)}{(2k-1)^{3\mu}} \right) h^2.$$ 

As the series $\sum_{k=1}^{\infty} \frac{k(2k+1)^2}{(2k-1)^{3\mu}}$ and $\sum_{k=1}^{\infty} \frac{k(2k+1)^2}{(2k-1)^{3\mu}}$ converge for $\mu > 4/3$, we conclude that the approximation order of $K_\mu$ is $O(h^2)$.

5 Cardinal basis functions

Applying $K_\mu$ to the unit data $\delta_i$ with $f_i = 1$ and $f_k = 0$ for $k \neq i$, gives the cardinal basis functions

$$\hat{B}_{\mu, i}(x) = K_\mu[\delta_i](x) = \sum_{j \in I_i} B_{\mu, j}(x)\lambda_{j, i}(x), \quad i = 1, \ldots, n.$$ (5.1)

As $K_\mu$ is linear, they allow us to rewrite the triangular Shepard operator as

$$K_\mu[f](x) = \sum_{i=1}^{n} \hat{B}_{\mu, i}(x)f_i,$$

and as the constant function $f(x) \equiv 1$ is in the precision set of $K_\mu$, it follows that the cardinal basis functions $\hat{B}_{\mu, i}$ form a partition of unity. However, in contrast to their classical counterparts $A_{\mu, i}$, they are not necessarily positive away from the nodes $x_i$.

Figures 9 and 11 show examples of these cardinal basis functions for $\mu = 2$. Negative function values are marked by grey colour, while the small brown regions near $x_i$ indicate function values greater than 1.
Comparing the basis functions to their classical counterparts shown in Figures 8 and 10, we observe that $\hat{B}_{2,i}$ is less “spiky” than $A_{2,i}$ and tends to zero faster with increasing distance from $x_i$. Another notable property is the behaviour of the gradient of $\hat{B}_{\mu,i}$ at the nodes. To this end let

$$I_i = \bigcup_{j \in J_i} \{j_1, j_2, j_3\}$$

be the set of indices of all nodes that share a triangle with $x_i$, that is, which are neighbours of $x_i$ in $T$, including the index $i$ itself.

Proposition 5.1. For any $i = 1, \ldots, n$ and $\mu > 1$, the cardinal basis function in (5.1) has local extrema at almost all nodes,

$$\nabla \hat{B}_{\mu,i}(x_k) = 0, \quad k \notin I_i,$$

and

$$\nabla \hat{B}_{\mu,i}(x_k) = \sum_{j \in J_i} B_{\mu,j}(x_k) c_{j,i}, \quad k \in I_i,$$

where $c_{j,i}$ are some constant vectors which depend only on $t_j$ and $x_i$.

Proof. First note that

$$\nabla \hat{B}_{\mu,i}(x) = \sum_{j \in I_i} \nabla B_{\mu,j}(x) \lambda_j \lambda_i(x) + B_{\mu,j}(x) \nabla \lambda_j \lambda_i(x).$$

If $k \notin I_i$, then it follows from (2.3) and (2.4) that both $B_{\mu,j}(x_k)$ and $\nabla B_{\mu,j}(x_k)$ vanish for any $j \in I_i$, which is sufficient to establish (5.2). To get (5.3), we distinguish two cases. On the one hand, if $k = i$, then $\lambda_j \lambda_i(x_i) = 1$ for any $j \in I_i$ and $\nabla \lambda_j \lambda_i(x_i) = 0$, as mentioned in (2.7). On the other hand, if $k \notin I_i \setminus \{i\}$, then for any $j \in I_i$ we have either $k \in \{j_1, j_2, j_3\}$, implying $\lambda_j \lambda_i(x_k) = 0$, or $k \notin \{j_1, j_2, j_3\}$, so that $\nabla B_{\mu,j}(x_k) = 0$. Overall, this gives (5.3) with $c_{j,i} = \nabla \lambda_j(x)$, which is a constant vector because $\lambda_j \lambda_i(x)$ is linear.

It follows from Proposition 5.1 that the interpolant $K_\mu[f]$ does not necessarily have flat spots at the nodes.
6 Numerical results

To verify the quadratic approximation order of the triangular Shepard operator $K_2$ in (4.1) predicted by Theorem 4.2, we carried out various numerical experiments with different sets of nodes and 12 test functions (see Figure 12), including those introduced by McLain [15], Franke [12, 13], and Renka and Brown [18]. We report the results of some of these experiments for different kinds of triangulations in Sections 6.1 and 6.2. In Section 6.3 we further present a comparison of the approximation accuracy of $K_2$ and some alternative state-of-the-art interpolation methods.

6.1 Approximation order for Delaunay triangulations

Our first series of experiments is based on six different Delaunay triangulations with increasing resolution (see Figure 13). These triangulations were generated by prescribing $N$ uniformly distributed nodes along the boundary of the unit square $R = [0, 1] \times [0, 1]$ and then using Shewchuk’s TRIANGLE program [20] to create a conforming Delaunay triangulation of $R$ with no angle smaller than 20 degrees and no triangle area greater than $a_{\text{max}} = 4\sqrt{3}/N^2$, by inserting Steiner points. Note that $a_{\text{max}}$ is the area of an equilateral triangle with edge length $4/N$, the spacing of the prescribed boundary nodes. Table 1 lists the number of vertices and triangles, as well as the maximum edge length $h_T = \max\{h_1, \ldots, h_m\}$ for the six triangulations. Note that for Delaunay triangulations $h_T$ is of the same order as $h$ in Theorem 4.2.

For each of the 12 test functions $f_i$ we constructed the triangular Shepard interpolant $K_2[f_i]$ and determined the maximum approximation error $e_{\text{max}}$ by evaluating $|f_i(x) - K_2[f_i](x)|$ at 100,000 random points $x \in R$ and recording the maximum value. The results are shown in Figure 14 and clearly demonstrate the quadratic approximation order of the operator $K_2$, except in the case of $f_{10}$, which is not in $C^{1,1}(R)$. For other values of $\mu > 4/3$ we obtained similar results.

Figure 15 shows a comparison between the reconstruction of test function $f_1$ using the classical Shepard interpolant $S_0[f_1]$ and the triangular Shepard interpolant $K_2[f_1]$, both based on samples taken at the nodes of the first four of our six Delaunay triangulations. While it is well-known that the Shepard interpolant behaves rather poorly, which has led to various improvements (see [21] and references therein), we include this comparison because the construction of the triangular Shepard operator is as simple as that of the classical Shepard operator, and we believe that it can be further improved significantly by following ideas similar to the ones used for extending the Shepard interpolant. Note that the superior aesthetic behaviour of the triangular Shepard interpolants $K_2$, compared to the Shepard interpolants, had also been observed by Little [14].

6.2 Approximation order for general triangulations

In a second series of experiments, we tested the approximation order of $K_2$ when used with respect to general triangulations, constructed in the following way. For each node $x_i \in X$ we choose among the 15 triangles...
Figure 13: Three of the six Delaunay triangulations used in our numerical experiments with $N = 40$ (left), $N = 80$ (middle), and $N = 160$ (right) boundary nodes (compare Table 1).

<table>
<thead>
<tr>
<th>$N$</th>
<th>$n$</th>
<th>$m$</th>
<th>$h_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>202</td>
<td>362</td>
<td>0.122196</td>
</tr>
<tr>
<td>80</td>
<td>777</td>
<td>1472</td>
<td>0.068061</td>
</tr>
<tr>
<td>160</td>
<td>2991</td>
<td>5820</td>
<td>0.034871</td>
</tr>
<tr>
<td>320</td>
<td>11638</td>
<td>22954</td>
<td>0.018046</td>
</tr>
<tr>
<td>640</td>
<td>46176</td>
<td>91710</td>
<td>0.008993</td>
</tr>
<tr>
<td>1280</td>
<td>195323</td>
<td>389364</td>
<td>0.004494</td>
</tr>
</tbody>
</table>

Table 1: Starting from $N$ uniformly distributed nodes on the boundary of the unit square, we generated Delaunay triangulations with $n$ vertices, $m$ triangles, and maximum edge length $h_T$ (compare Figure 13).

Figure 14: Log-log plot of the approximation error $e_{\text{max}}$ over the maximum edge length $h_T$ for the 12 test functions in Figure 12 and the Delaunay triangulations in Figure 13 and Table 1. As reference, the dotted line indicates a perfect quadratic trend.

that connect $x_i$ with 2 of its 6 nearest neighbours in $X$ the one which locally reduces the error bound of the associated linear interpolant in (3.3). After omitting duplicate triangles we get a triangulation $T$ of the nodes with $m \leq n$ triangles, where some of the triangles may overlap each other. As the number of triangles is only about $1/3$ the number of triangles in the Delaunay triangulation of $X$, we call $T$ a compact triangulation of $X$. For our numerical experiments we created compact triangulations of the sets of nodes given by the six Delaunay triangulations in Figure 13 (see Figure 16 and Table 2) and determined the approximation errors for the triangular Shepard interpolant as described in Section 6.1. Figure 17 summarizes the results and confirms that the approximation order is again quadratic (the plots are again over the maximum edge length $h_T$, which is of the same order as $h$), except in the case of $f_{10}$, for which Theorem 4.2 does not apply.

The bottom row in Figure 15 shows the reconstruction of test function $f_1$ using the triangular Shepard interpolant $K_2[f_1]$ with respect to the first four of our six compact triangulations. Comparing the results to those obtained by using Delaunay triangulations, we can see that both the visual quality and the maximum approximation error are similar if the compact triangulation has about twice as many triangle as the Delaunay triangulation.
Figure 15: Reconstruction of Franke’s principal test function $f_1$ from $n$ samples using classical Shepard interpolation (top) and triangular Shepard interpolation, based on Delaunay (middle) and compact triangulations (bottom).

6.3 Approximation accuracy

We finally carried out a series of experiments to compare the approximation accuracies of the triangular Shepard operator $K_2$ and

1. the global version of the classical Shepard operator $S_2$ given in (1.1);
2. the global version of the first-order Shepard–Taylor operator $S_{T_1}$, which is obtained by substituting in (1.1) functional evaluation at $x_i$ with the first-order Taylor polynomial of $f$ at $x_i$;
3. the local version of the classical Shepard operator $\tilde{S}_2$, where the Shepard basis functions (1.2) are multiplied by Franke–Little weights [3] in order to make them compactly supported;
4. the local version of the first-order Shepard–Taylor operator $\tilde{S}_{T_1}$;
5. the linear Shepard operator LSHEP [21], which substitutes in $\tilde{S}_{T_1}$ the first-order Taylor polynomial of $f$ at $x_i$ with the linear polynomial that interpolates $f$ at $x_i$ and fits the data at the 4 nodes closest to $x_i$ best in the least squares sense.

Note that the global and the local version of the classical Shepard operator, $S_2$ and $S_{T_1}$, as well as the LSHEP operator use the same data as $K_2$, that is, they rely on function values only. Instead, the global and local version of the first-order Shepard–Taylor operator, $\tilde{S}_2$ and $\tilde{S}_{T_1}$, require the first-order derivatives at each
Three of the six compact triangulations used in our numerical experiments with $m = 133$ (left), $m = 490$ (middle), and $m = 1860$ (right) triangles (compare Table 2).

<table>
<thead>
<tr>
<th>$n$</th>
<th>$m$</th>
<th>$h_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>202</td>
<td>133</td>
<td>0.130966</td>
</tr>
<tr>
<td>777</td>
<td>490</td>
<td>0.065158</td>
</tr>
<tr>
<td>2991</td>
<td>1860</td>
<td>0.034552</td>
</tr>
<tr>
<td>11638</td>
<td>7151</td>
<td>0.016804</td>
</tr>
<tr>
<td>46176</td>
<td>28582</td>
<td>0.008769</td>
</tr>
<tr>
<td>195323</td>
<td>120350</td>
<td>0.004651</td>
</tr>
</tbody>
</table>

Table 2: Starting from the set of nodes given by the six Delaunay triangulations in Figure 13, we generated compact triangulations with $n$ vertices, $m$ triangles and maximum edge length $h_T$ (compare Figure 16).

![Figure 17: Log-log plot of the approximation error $e_{max}$ over the maximum edge length $h_T$ for the 12 test functions in Figure 12 and the triangulations in Figure 16 and Table 2. As reference, the dotted line indicates a perfect quadratic trend.](image)

The results show that the global triangular Shepard operator $K_2$ is comparable to the local Shepard interpolation methods. This encourages us to develop and analyse a local version of $K_2$ in future work and to study...
<table>
<thead>
<tr>
<th>term</th>
<th>$S_2$</th>
<th>$S_{11}$</th>
<th>$K_0$ Delaunay</th>
<th>$K_2$ compact</th>
<th>$\bar{S}_2$</th>
<th>$\bar{S}_{11}$</th>
<th>LSHEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$e_{\text{max}}$</td>
<td>$4.34 \times 10^{-1}$</td>
<td>$1.06 \times 10^{-1}$</td>
<td>$5.16 \times 10^{-2}$</td>
<td>$1.00 \times 10^{-1}$</td>
<td>$2.15 \times 10^{-1}$</td>
<td>$1.19 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$5.93 \times 10^{-2}$</td>
<td>$2.31 \times 10^{-2}$</td>
<td>$7.79 \times 10^{-3}$</td>
<td>$1.47 \times 10^{-2}$</td>
<td>$2.55 \times 10^{-2}$</td>
<td>$1.14 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$9.07 \times 10^{-2}$</td>
<td>$3.07 \times 10^{-2}$</td>
<td>$1.21 \times 10^{-2}$</td>
<td>$2.35 \times 10^{-2}$</td>
<td>$4.29 \times 10^{-2}$</td>
<td>$1.94 \times 10^{-2}$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$e_{\text{max}}$</td>
<td>$6.17 \times 10^{-4}$</td>
<td>$2.65 \times 10^{-4}$</td>
<td>$2.66 \times 10^{-4}$</td>
<td>$4.20 \times 10^{-4}$</td>
<td>$6.79 \times 10^{-4}$</td>
<td>$2.28 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$2.27 \times 10^{-2}$</td>
<td>$7.15 \times 10^{-3}$</td>
<td>$2.64 \times 10^{-3}$</td>
<td>$4.53 \times 10^{-3}$</td>
<td>$5.95 \times 10^{-3}$</td>
<td>$3.03 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$2.62 \times 10^{-2}$</td>
<td>$8.78 \times 10^{-3}$</td>
<td>$4.76 \times 10^{-3}$</td>
<td>$7.76 \times 10^{-3}$</td>
<td>$1.14 \times 10^{-2}$</td>
<td>$5.04 \times 10^{-3}$</td>
</tr>
<tr>
<td>$f_3$</td>
<td>$e_{\text{max}}$</td>
<td>$9.48 \times 10^{-2}$</td>
<td>$4.16 \times 10^{-2}$</td>
<td>$9.46 \times 10^{-2}$</td>
<td>$2.31 \times 10^{-2}$</td>
<td>$4.45 \times 10^{-2}$</td>
<td>$3.31 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$2.12 \times 10^{-2}$</td>
<td>$7.16 \times 10^{-3}$</td>
<td>$2.71 \times 10^{-3}$</td>
<td>$3.39 \times 10^{-3}$</td>
<td>$7.44 \times 10^{-3}$</td>
<td>$3.35 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$2.78 \times 10^{-2}$</td>
<td>$1.02 \times 10^{-2}$</td>
<td>$8.04 \times 10^{-3}$</td>
<td>$5.05 \times 10^{-3}$</td>
<td>$1.03 \times 10^{-2}$</td>
<td>$5.32 \times 10^{-3}$</td>
</tr>
<tr>
<td>$f_4$</td>
<td>$e_{\text{max}}$</td>
<td>$9.24 \times 10^{-2}$</td>
<td>$6.32 \times 10^{-2}$</td>
<td>$3.07 \times 10^{-2}$</td>
<td>$2.43 \times 10^{-2}$</td>
<td>$3.32 \times 10^{-2}$</td>
<td>$2.49 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$2.06 \times 10^{-2}$</td>
<td>$3.12 \times 10^{-2}$</td>
<td>$1.47 \times 10^{-3}$</td>
<td>$2.73 \times 10^{-3}$</td>
<td>$8.28 \times 10^{-3}$</td>
<td>$2.40 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$2.67 \times 10^{-2}$</td>
<td>$3.39 \times 10^{-2}$</td>
<td>$3.06 \times 10^{-3}$</td>
<td>$4.09 \times 10^{-3}$</td>
<td>$1.05 \times 10^{-2}$</td>
<td>$4.15 \times 10^{-3}$</td>
</tr>
<tr>
<td>$f_5$</td>
<td>$e_{\text{max}}$</td>
<td>$1.30 \times 10^{-1}$</td>
<td>$1.25 \times 10^{-1}$</td>
<td>$8.57 \times 10^{-2}$</td>
<td>$2.18 \times 10^{-2}$</td>
<td>$7.29 \times 10^{-2}$</td>
<td>$2.07 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$1.87 \times 10^{-2}$</td>
<td>$4.41 \times 10^{-2}$</td>
<td>$2.51 \times 10^{-3}$</td>
<td>$2.96 \times 10^{-3}$</td>
<td>$9.64 \times 10^{-3}$</td>
<td>$5.32 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$2.44 \times 10^{-2}$</td>
<td>$4.78 \times 10^{-2}$</td>
<td>$7.59 \times 10^{-3}$</td>
<td>$4.35 \times 10^{-3}$</td>
<td>$1.27 \times 10^{-2}$</td>
<td>$6.42 \times 10^{-3}$</td>
</tr>
<tr>
<td>$f_6$</td>
<td>$e_{\text{max}}$</td>
<td>$9.30 \times 10^{-4}$</td>
<td>$3.08 \times 10^{-4}$</td>
<td>$1.93 \times 10^{-4}$</td>
<td>$2.67 \times 10^{-4}$</td>
<td>$5.01 \times 10^{-4}$</td>
<td>$2.84 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$1.13 \times 10^{-1}$</td>
<td>$7.35 \times 10^{-3}$</td>
<td>$1.78 \times 10^{-3}$</td>
<td>$3.36 \times 10^{-3}$</td>
<td>$5.04 \times 10^{-3}$</td>
<td>$2.51 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$1.77 \times 10^{-1}$</td>
<td>$9.15 \times 10^{-2}$</td>
<td>$3.13 \times 10^{-2}$</td>
<td>$5.82 \times 10^{-3}$</td>
<td>$9.02 \times 10^{-3}$</td>
<td>$4.52 \times 10^{-3}$</td>
</tr>
<tr>
<td>$f_7$</td>
<td>$e_{\text{max}}$</td>
<td>$1.05 \times 10^{0}$</td>
<td>$7.12 \times 10^{-1}$</td>
<td>$5.01 \times 10^{-1}$</td>
<td>$8.10 \times 10^{-1}$</td>
<td>$9.41 \times 10^{-1}$</td>
<td>$1.95 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$1.09 \times 10^{0}$</td>
<td>$4.86 \times 10^{-1}$</td>
<td>$2.80 \times 10^{-1}$</td>
<td>$6.13 \times 10^{-1}$</td>
<td>$6.98 \times 10^{-1}$</td>
<td>$2.75 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$1.58 \times 10^{0}$</td>
<td>$8.03 \times 10^{-2}$</td>
<td>$5.65 \times 10^{-2}$</td>
<td>$1.05 \times 10^{-1}$</td>
<td>$1.19 \times 10^{-1}$</td>
<td>$3.79 \times 10^{-2}$</td>
</tr>
<tr>
<td>$f_8$</td>
<td>$e_{\text{max}}$</td>
<td>$2.15 \times 10^{-1}$</td>
<td>$9.66 \times 10^{-2}$</td>
<td>$1.32 \times 10^{-2}$</td>
<td>$1.63 \times 10^{-2}$</td>
<td>$9.02 \times 10^{-2}$</td>
<td>$5.83 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$2.78 \times 10^{-2}$</td>
<td>$1.17 \times 10^{-3}$</td>
<td>$1.33 \times 10^{-3}$</td>
<td>$3.00 \times 10^{-3}$</td>
<td>$1.15 \times 10^{-2}$</td>
<td>$9.73 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$4.32 \times 10^{-2}$</td>
<td>$1.77 \times 10^{-3}$</td>
<td>$1.92 \times 10^{-3}$</td>
<td>$3.90 \times 10^{-3}$</td>
<td>$1.55 \times 10^{-2}$</td>
<td>$1.35 \times 10^{-2}$</td>
</tr>
<tr>
<td>$f_9$</td>
<td>$e_{\text{max}}$</td>
<td>$3.17 \times 10^{-1}$</td>
<td>$1.20 \times 10^{-1}$</td>
<td>$2.68 \times 10^{-1}$</td>
<td>$1.68 \times 10^{-1}$</td>
<td>$1.63 \times 10^{-1}$</td>
<td>$9.87 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{mean}}$</td>
<td>$7.68 \times 10^{-2}$</td>
<td>$2.96 \times 10^{-2}$</td>
<td>$1.19 \times 10^{-2}$</td>
<td>$3.30 \times 10^{-2}$</td>
<td>$3.77 \times 10^{-2}$</td>
<td>$2.12 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$e_{\text{MS}}$</td>
<td>$1.03 \times 10^{-1}$</td>
<td>$3.72 \times 10^{-2}$</td>
<td>$2.15 \times 10^{-2}$</td>
<td>$3.10 \times 10^{-2}$</td>
<td>$4.77 \times 10^{-2}$</td>
<td>$2.89 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Table 3: Comparison of six interpolation operators applied to the 12 test functions in Figure 12 using the interpolation nodes of the Franke data set (see Figure 18). The smallest error in each row is marked in boldface.
combined operators, based on $K_2$, with higher degrees of exactness. It is worth noting that local versions of $K_2$ can be interpreted as partition of unity methods, which were suggested and deeply studied by Babuška and Melenk [2, 16] in the context of meshfree Galerkin methods for solving partial differential equations (see also [11, 22]). Hence, the quadratic approximation order for local versions of $K_2$ can be deduced from the general theory in the $L_2$-norm [16] as well as the $L_{\infty}$-norm [22, Theorem 15.17].

Acknowledgements

This paper was supported by the INdAM-GNCS 2014 Research Project of the Italian National Group for Scientific Computation of the National Institute for Advanced Mathematics “F. Severi”.

References