Proceedings of the 17th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2017
4–8 July, 2017.

An efficient technique for the interpolation on compact triangulations

Roberto Cavoretto[1], Alessandra De Rossi[1], Francesco Dell'Accio[2] and Filomena Di Tommaso[2]

[1] *Department of Mathematics "Giuseppe Peano", University of Torino*

[2] *Department of Mathematics and Computer Science, University of Calabria*

emails: roberto.cavoretto@unito.it, alessandra.derossi@unito.it, francesco.dellaccio@unical.it, ditommaso@mat.unical.it

Abstract

In this paper we present an efficient scheme for the computation of triangular Shepard method. More precisely, it is well known that the triangular Shepard method reaches an approximation order better than the Shepard one [4], but it needs to identify useful general triangulation of the node set. Here we propose a searching technique used to detect and select the nearest neighbor points in the interpolation scheme [2, 3]. It consists in determining the closest points belonging to the different neighborhoods and consequently applies to the triangulation-based approach. Numerical results show efficiency of the interpolation procedure.

Key words: scattered data interpolation, triangular Shepard method, fast computation, approximation algorithms

MSC 2000: 65D05, 65D15, 41A05

1 Introduction

Scattered data consists of a set of points $X_n = \{x_1, \ldots, x_n\}$ and corresponding functional values $f_1, \ldots, f_n$, where the points have no structure or order between their relative locations. Among the various approaches to interpolating scattered data, the Shepard method [8] is one of the earliest techniques. It defines an interpolating function as a convex combination of the functional values, that is a linear combination of them with non negative coefficients (or weight functions or basis functions) which are inverse distances to the scattered points and form a partition of unity. The main drawback of the Shepard method is

 ISBN: 978-84-617-8694-7

its low polynomial precision (only constants) that badly affects the reconstructed surface. Several variants of the Shepard method have been considered to overcome this drawback. Among them, the triangular Shepard method [7] is a convex combination of local linear interpolants with triangle-based weight functions, which are the product of inverse distances from the vertices of triangles and form a partition of unity. The main feature of the triangular Shepard method is its linear precision without using derivative data, although, for achieving a good accuracy of approximation, it needs to identify useful general triangulation of the node set.

An efficient organization of the scattered data, when local interpolation is used, turns out to be crucial. To this aim, in literature, techniques known as kd-trees, which are not specifically implemented for a specific interpolation scheme, have already been designed, [1, 6]. In this paper, we propose the use of a versatile partitioning structure, called the block-based partitioning structure, given in [3], which is suitably adapted to triangular Shepard interpolation.

The paper is organized as follows. In Section 2 we consider interpolation using the triangular Shepard method and the selection of the compact triangulation. Section 3 is devoted to present the searching technique used to detect and select the nearest neighbor points in our interpolation scheme. Finally, Section 4 shows some numerical results.

2 Interpolation on compact triangulations

2.1 Triangular Shepard method

In 1983 Little [7] introduced a variant of the Shepard method, called triangular Shepard, which is a Shepard-like convex combination of the linear interpolants of a set of triangles. More precisely, if we denote by $X_n = \{x_1, x_2, \ldots, x_n\}$ a set of nodes of $\mathbb{R}^2$ with associated function data $f_1, \ldots, f_n$ and by $T_m = \{t_1, t_2, \ldots, t_m\}$ a set of triangles which vertices are points of X_n, the triangular Shepard operator is defined by

$$K_\mu[f](x) = \sum_{j=1}^{m} B_{\mu,j}(x) L_j(x), \quad \mu > 0, \tag{1}$$

where $L_j(x)$ is the linear interpolant on the vertices of t_j, $j = 1, \ldots, m$, and the weight functions $B_{\mu,j}(x)$ are defined by

$$B_{\mu,j}(x) = \frac{\prod_{\ell=1}^{3} \frac{1}{\|x - x_{j_\ell}\|^\mu}}{\sum_{k=1}^{m} \prod_{\ell=1}^{3} \frac{1}{\|x - x_{k_\ell}\|^\mu}}, \qquad j = 1, \ldots, m. \tag{2}$$

 ISBN: 978-84-617-8694-7

As noticed by Little himself, the triangular Shepard operator (1) exceeds the Shepard method both in polynomial precision and esthetic behaviour. In fact, it has linear precision while Shepard method achieves only constant precision. The better polynomial precision of the triangular Shepard method reflects on a higher order of approximation. As shown in [4] the triangular Shepard method reaches quadratic approximation order while the Shepard method achieves at most linear approximation order [5]. We remark that the definition of the triangular Shepard operator (1) requires an appropriate list of triangles to be identified. However, these triangles can realize only a general triangulation of the node set X_n, that is a triangulation in which some triangles may overlap or be disjoint.

2.2 Selection of the compact triangulation

In order to identify useful general triangulation of the node set X_n we take into account theoretical results achieved in [4], which link the bound for the remainder term of the linear interpolant $L_j(x)$ with the bound for the remainder of the triangular Shepard operator (1). In particular, all triangles should have a rather regular form (as near as possible to equilateral triangles) and two quantities, denoted by h' and h'', play a key role in the choice of these triangles: the first one is the fill distance in the maximum norm and controls the uniformity of the triangle distribution, the second one excludes the presence of large triangles.

For each node x_i of the node set X_n we choose, among the 15 triangles with a vertex in x_i and other 2 vertices among its 6 nearest neighbors in X_n, the one which locally reduces the bound

$$2\|x - x_{j_1}\|^2 + 4h_j C_j \|x - x_{j_1}\| \tag{3}$$

for the error of the local linear interpolant, where x_{j_1} denotes the first vertex of t_j, h_j denotes the maximum edge length of t_j and C_j is a constant which depends only on the shape of t_j. We omit duplicate triangles and we get a triangulation T_m of the nodes with $m \leq n$ triangles, in which some of the triangles may overlap or be disjoint.

3 Localizing searching technique

In this section we present the searching technique used to detect and select the nearest neighbor points in our interpolation scheme [2, 3]. Though this procedure can be applied on a generic domain $R \subseteq \mathbb{R}^2$ and in higher dimensions, for our purpose we here pursue this description focusing on the unit square, i.e. $R = [0, 1] \times [0, 1]$.

Firstly, we define a circular neighborhood of radius

$$\delta = \frac{2}{d}, \tag{4}$$

 ISBN: 978-84-617-8694-7

with

$$d = \left\lfloor \frac{\sqrt{n}}{2} \right\rfloor, \tag{5}$$

where each neighborhood is centred at a data point belonging to R. To localize points in our method, we set $n/d^2 = 4$. Note that the larger (smaller) the value of d is, the more (less) localizing the scheme is.

In order to determine the closest points belonging to the different neighborhoods and consequently apply our triangulation-based approach, we propose a new structure that partitions the domain in blocks of square shape. Such technique results in an effective searching procedure which is quite efficient from a computational viewpoint. For this scope we partition the domain R with b^2 square blocks, b being the number of blocks along one side of the unit square defined as

$$b = \left\lceil \frac{1}{\delta} \right\rceil. \tag{6}$$

It follows that the side of each square block turns out to be equal to the neighborhood radius. Hence, such a choice (seemingly trivial) allows us to examine in the searching process only a small number of blocks, significantly reducing the computational effort compared to standard or more advanced searching procedures such kd-trees [1, 9]. In fact, our searching routine is performed in a constant time, independently from the initial number of nodes considered.

In our partitioning technique square blocks are numbered from 1 to b^2, following the lexicographic order "bottom to top, left to right". By a repeated use of a quicksort routine the set X_n is thus partitioned by the block-based partitioning structure into b^2 subsets X_{n_k}, $k = 1, \ldots, b^2$, where X_{n_k} are the points belonging to nine blocks: the k-th block and its eight neighboring blocks, see Figure 1. In such framework, we are able to get an optimal procedure to find the interpolation nodes closest to each of points.

4 Numerical results

In our numerical experiments we report results obtained by using the triangular Shepard interpolant (1). All the experiments are carried out by means of a grid of $n_e = 21 \times 21$ evaluation points on $R = [0, 1] \times [0, 1]$ by using several sets of Halton points [6]. The used test function is the Franke's one,

$$\begin{aligned} f(x,y) = 0.75 \exp\left(-\frac{(9x-2)^2 + (9y-2)^2}{4}\right) + 0.50 \exp\left(-\frac{(9x-7)^2 + (9y-3)^2}{4}\right) \\ +0.75 \exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{4}\right) - 0.20 \exp\left(-(9x-4)^2 - (9y-7)^2\right). \end{aligned}$$

 ISBN: 978-84-617-8694-7

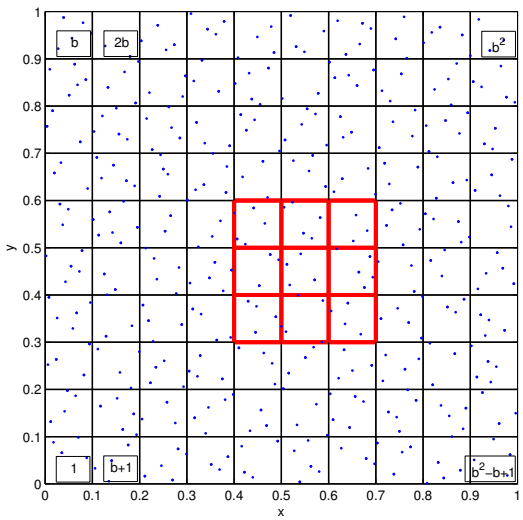

Figure 1: Example of 2D block-based partitioning structure with a data point set: in red the k-th block and its eight neighboring blocks, in blue a set of scattered points contained in the unit square R.

n	RMSE	t_{old}	t_{new}
10000	2.87e-4	27.3056	7.0784
20000	1.53e-4	242.5568	18.1962
40000	7.51e-5	–	49.3938
80000	3.59e-5	–	245.2871

Table 1: CPU times (in seconds) and RMSEs on a grid of $n_e = 21 \times 21$ evaluation points for the Franke's test function using different sets of n Halton points.

In Table 1 we show the CPU times computed in seconds and the root mean square error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_e} {e_i}^2}{n_e}}$$

with

$$e_i = |f(P_i) - K_2[f](P_i)|, \tag{7}$$

P_i being an evaluation point in R. In particular, we compare performance of the procedure which computes all the distances between the scattered points (t_{old}) with the one using the block-based searching technique (t_{new}).

 ISBN: 978-84-617-8694-7

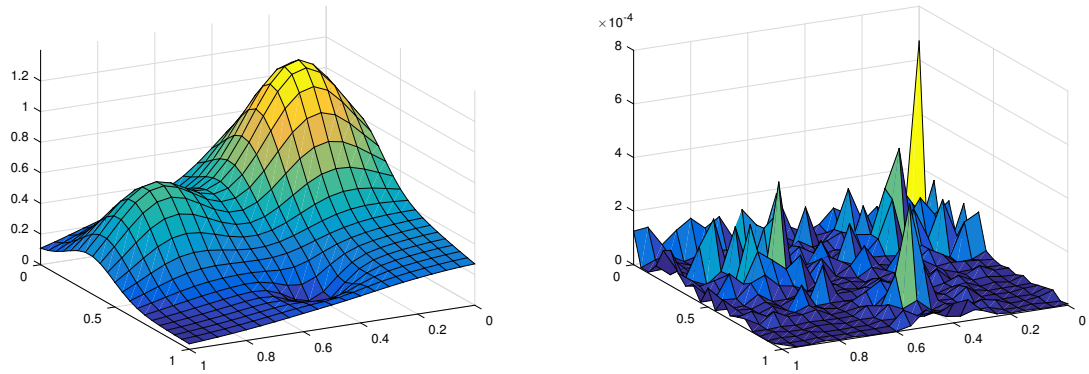

Figure 2: Approximation of Franke's test function (left) and the absolute errors (right) on a grid of $n_e = 21 \times 21$ evaluation points using $n = 40000$ Halton points.

Figure 2 shows the reconstruction of Franke's test function and the absolute values e_i on $n = 40000$ Halton points.

Acknowledgements

This research has been accomplished within the RITA "Research ITalian network on Approximation". This work was partially supported by the projects "Metodi e modelli numerici per le scienze applicate" and "Approssimazione multivariata e algoritmi efficienti con applicazioni a problemi algebrici, differenziali e integrali" of the Department of Mathematics of the University of Torino. Moreover, this research was supported by GNCS–INdAM 2016-2017 project and by a research fellow of the Centro Universitario Cattolico.

References

[1] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, *An optimal algorithm for approximate nearest neighbor searching in fixed dimensions*, J. ACM **45** (1998) 891–923.

[2] R. Cavoretto, A. De Rossi, *A meshless interpolation algorithm using a cell-based searching procedure*, Comput. Math. Appl. **67** (2014) 1024–1038.

[3] R. Cavoretto, A. De Rossi, E. Perracchione, *Efficient computation of partition of unity interpolants through a block-based searching technique*, Comput. Math. Appl. **71** (2016) 2568–2584.

[4] F. Dell'Accio, F. Di Tommaso, K. Hormann, *On the approximation order of the triangular Shepard interpolation*, IMA J. Numer. Anal. **36** (2016) 359–379.

[5] R. Farwig, *Rate of convergence of Shepard's global interpolation formula*, Math. Comp **46** (1986) 577–590.

[6] G.E. Fasshauer, *Meshfree approximation methods with MATLAB*, World Scientific, Singapore, 2007.

[7] F. Little, *Convex combination surfaces*, in R.E. Barnhill, W. Boehm (eds.), *Surfaces in Computer Aided Geometric Design*, North-Holland, Amsterdam, 1983.

[8] D. Shepard, *A two-dimensional interpolation function for irregularly-spaced data.*, in *Proceedings of the 23rd ACM National Conference*, New York: ACM Press, pp. 517524.

[9] H. Wendland, *Scattered data approximation*, Cambridge University Press, Cambridge, 2005.

 ISBN: 978-84-617-8694-7