Computers and Mathematics with Applications 71 (2016) 2568–2584

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Efficient computation of partition of unity interpolants through a block-based searching technique

R. Cavoretto *, A. De Rossi, E. Perracchione

Department of Mathematics "G. Peano", University of Torino, via Carlo Alberto 10, I-10123 Torino, Italy

ARTICLE INFO

Article history:
Received 10 October 2015
Received in revised form 12 February 2016
Accepted 13 April 2016
Available online 4 May 2016

Keywords:
Meshfree approximation
Partition of unity method
Fast algorithms
Searching procedures
Radial basis functions
Scattered data interpolation

ABSTRACT

In this paper we propose a new efficient interpolation tool, extremely suitable for large scattered data sets. The partition of unity method is used and performed by blending Radial Basis Functions (RBFs) as local approximants and using locally supported weight functions. In particular we present a new space-partitioning data structure based on a partition of the underlying generic domain in blocks. This approach allows us to examine only a reduced number of blocks in the search process of the nearest neighbour points, leading to an optimized searching routine. Complexity analysis and numerical experiments in two- and three-dimensional interpolation support our findings. Some applications to geometric modelling are also considered. Moreover, the associated software package written in MATLAB is here discussed and made available to the scientific community.

1. Introduction

Meshfree methods are popular tools for solving problems of interpolation and numerical resolution of differential equations. They take advantage of being flexible with respect to geometry, easy to implement in higher dimensions, and can also provide high order convergence. Recently, in approximation theory a specific method has been proved to be effective for interpolation of large scattered data sets, the partition of unity method. Its origin can be found in the context of partial differential equations (PDEs) [1,2]. In scattered data interpolation it is implemented using RBFs as local approximants, since this is the most efficient tool for interpolation of scattered data [3]. The main disadvantage of radial kernel-based method is the computational cost associated with the solution of (usually) large linear systems, therefore recent researches have been directed towards a change of the basis, either rendering them more stable, or considering a local method involving RBFs (see e.g. [4–9]). Here we focus on the localized RBF-based partition of unity approximation. As the name of the partition of unity method suggests, in such local approach, the efficient organization of scattered data is the crucial step. Precisely, in the literature, techniques as *kd-trees*, which allow to partition data in a k-dimensional space, and related searching procedures have already been designed [10–12,3,13]. Even if such techniques enable us to work with high dimensions, they are not specifically implemented for the partition of unity method.

In this paper, starting from the results shown in [14–16], where efficient searching procedures based on the partition of underlying domains in strips or crossed strips are considered, we propose a versatile software for bivariate and trivariate interpolation which makes use of a new partitioning structure, named *block-based partitioning structure*, and a novel related searching procedure. It strictly depends on the size of the partition of unity subdomains. Such technique allows to deal with a truly large number of data with a relatively low computational complexity.

* Corresponding author.
E-mail addresses: roberto.cavoretto@unito.it (R. Cavoretto), alessandra.derossi@unito.it (A. De Rossi), emma.perracchione@unito.it (E. Perracchione).

http://dx.doi.org/10.1016/j.camwa.2016.04.021

More precisely, our procedure for bivariate and trivariate interpolation consists in covering, at first, the reconstruction domain with several non-overlapping small squares or cubes, named *blocks*. Then the usually large scattered data set is distributed among the different blocks by recursive calls to a sorting routine. Once the scattered data are stored in such blocks, an optimized searching procedure is performed enabling us to solve the local interpolation problems arising from the domain decomposition. Specifically, such structure, built ad hoc for the partition of unity method, enables us to run the searching procedure in constant time complexity, independently from the initial number of nodes. An extensive complexity analysis supports our findings and moreover comparisons with other common techniques, as kd-trees, will be carried out. Interpolating large scattered data sets using procedures competitive with the most advanced techniques is thus our main purpose.

A second meaningful feature of our procedures is the flexibility with respect to the problem geometry. In general, in literature the scattered data interpolation problem is considered in very simple and regular domains, such as squares or cubes [3,17]. This approach is limiting in the context of meshfree methods because of the versatility of the meshless technique with respect to domains having different shapes. Instead in this work, our aim is to provide an automatic software that allows to solve scattered data interpolation problems in generic domains. Specifically, here we focus on convex domains. This choice is due to the fact that our scope consists in solving interpolation problems in domains which are, in general, a priori unknown, i.e. problems arising from applications [18,19].

In what follows, in order to point out the versatility of the software, we will investigate several applications of such algorithm. For 2D data sets we stress the importance of having such versatile tool in biomathematics, presenting a short sketch about the reconstruction of the attraction basins [18]. The same approach can also be employed in the approximation of the so-called *sensitivity surfaces* [19]. Then, for 3D data sets, we analyse the problem of modelling implicit surfaces via partition of unity interpolation [20,21]. It is known that the reconstruction of 3D objects is computationally expensive because of the large amount of data. Thus, the importance of having an efficient partitioning structure in such framework follows.

The paper is organized as follows. In Section 2 we recall theoretical preliminaries on local RBF-based partition of unity approximation. In Section 3, we describe in detail the block-based partition of unity algorithms for bivariate and trivariate interpolations, which are based on the use of the new block-based partitioning and searching procedures. Computational complexity of these interpolation algorithms is then analysed in Section 4. In Section 5 we report numerical experiments devoted to point out the accuracy of our algorithms. Section 6 contains some applications in biomathematics and CAGD. Section 7 deals with conclusions and future work. We point out that the algorithms are made available to the scientific community in a downloadable free software package:

http://hdl.handle.net/2318/158790.

2. Preliminaries

In this section we briefly review the partition of unity approximation based on a localized use of RBF interpolants. This computational technique is meshfree and effectively works with large sets of scattered data points [3,13].

2.1. RBF interpolation

Given a set $\mathcal{X}_N = \{\boldsymbol{x}_i \in \mathbb{R}^M, i = 1, \ldots, N\}$ of N distinct *data points*, also called *data sites* or *nodes*, in a domain $\Omega \subseteq \mathbb{R}^M$, and a corresponding set $\mathcal{F}_N = \{f_i = f(\boldsymbol{x}_i), i = 1, \ldots, N\}$ of *data values* or *function values* obtained by possibly sampling any (unknown) function $f : \Omega \longrightarrow \mathbb{R}$, the standard RBF interpolation problem consists in finding an interpolant $R : \Omega \longrightarrow \mathbb{R}$ of the form

$$R(\boldsymbol{x}) = \sum_{i=1}^{N} c_i \phi(\|\boldsymbol{x} - \boldsymbol{x}_i\|_2), \quad \boldsymbol{x} \in \Omega, \tag{1}$$

where $\|\cdot\|_2$ is the Euclidean norm, and $\phi : [0, \infty) \longrightarrow \mathbb{R}$ is a RBF [22,23]. The coefficients $\{c_i\}_{i=1}^N$ are determined by enforcing the interpolation conditions

$$R(\boldsymbol{x}_i) = f_i, \quad i = 1, \ldots, N. \tag{2}$$

Imposing the conditions (2) leads to a symmetric linear system of equations

$$\Phi \boldsymbol{c} = \boldsymbol{f}, \tag{3}$$

where $\Phi_{ki} = \phi(\|\boldsymbol{x}_k - \boldsymbol{x}_i\|_2)$, $k, i = 1, \ldots, N$, $\boldsymbol{c} = [c_1, \ldots, c_N]^T$, and $\boldsymbol{f} = [f_1, \ldots, f_N]^T$. When $\boldsymbol{c}$ is found by solving the system (3), we can evaluate the RBF interpolant at a point $\boldsymbol{x}$ as

$$R(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{c},$$

where $\boldsymbol{\phi}^T(\boldsymbol{x}) = [\phi(\|\boldsymbol{x} - \boldsymbol{x}_1\|_2), \ldots, \phi(\|\boldsymbol{x} - \boldsymbol{x}_N\|_2)]$.

The interpolation problem is well-posed, i.e. a solution to the problem exists uniquely, if and only if the matrix Φ is nonsingular. A sufficient condition to have nonsingularity is that Φ is positive definite.