

# Adaptive scattered data fitting by extension of local approximations to hierarchical splines

Cesare Bracco\*, Carlotta Giannelli†, Alessandra Sestini‡

Dipartimento di Matematica e Informatica “U. Dini”, Università degli Studi di Firenze,  
Viale Morgagni 67/a, 50134 Firenze, Italy

## Abstract

We introduce an adaptive scattered data fitting scheme as extension of local least squares approximations to hierarchical spline spaces. To efficiently deal with non-trivial data configurations, the local solutions are described in terms of (variable degree) polynomial approximations according not only to the number of data points locally available, but also to the smallest singular value of the local collocation matrices. These local approximations are subsequently combined without the need of additional computations with the construction of hierarchical quasi-interpolants described in terms of truncated hierarchical B-splines. A selection of numerical experiments shows the effectivity of our approach for the approximation of real scattered data sets describing different terrain configurations.

**Keywords:** Scattered data fitting, Hierarchical splines, THB-splines, Local least squares, Quasi-Interpolation.

## 1 Introduction

Surface reconstruction of unstructured large data sets requires suitable *adaptive* schemes that facilitate the computation of high-quality approximations with an increased level of resolution only in strictly localized areas. The resulting compact representation automatically identifies the parts of the domain where an increased number of degrees of freedom is needed according to the data distribution (high concentrations of data points usually define local details to be suitably reconstructed). The problem of reconstructing scattered data of high complexity arises in various application areas, ranging from scanner acquisitions to geographic benchmarks, and it is a relevant component for industrial and medical purposes where the visualization and subsequent manipulation of large random data configurations are usually required. Note that scattered data can have significantly different distributions, e.g. data with highly varying density, data with voids, contour data.

Standard computer aided design software tools based on the tensor-product B-spline model do not provide local refinement capabilities. Spline adaptivity may easily be achieved by

---

\*cesare.bracco@unifi.it

†carlotta.giannelli@unifi.it

‡alessandra.sestini@unifi.it

considering multilevel B-spline extensions, where the tensor-product structure is preserved at any level. Hierarchical B-spline constructions of this kind were originally proposed by [9] to define hierarchical spline surfaces in terms of a sequence of overlays. By considering truncated hierarchical B-splines (THB-splines) [11], it is possible to define a strongly stable basis for hierarchical spline spaces that forms a convex partition of unity [12]. THB-splines also guarantee the so-called *preservation of coefficients*: truncated basis functions preserve the coefficients of functions expressed in terms of B-splines of a certain hierarchical level, see again [12]. This allows us to directly extend any quasi-interpolation operator defined in the space of tensor-product B-splines to the hierarchical setting without the need of additional computation [27]. Note that the hierarchical B-spline model may be applied in combination with uniform and non-uniform refinement, different degrees and smoothness, and related constructions easily extend to the general multivariate setting.

We present an adaptive scattered data fitting scheme by extension of local discrete least squares approximations to hierarchical spline spaces. To efficiently deal with non-trivial data configurations, the local solutions are described in terms of variable degree polynomial approximations, according not only to the number of data points locally available, but also to the smallest singular value of the local collocation matrices. Note that the possibility of using higher degree polynomials in each local approximation can be exploited only for sufficiently dense data sets. These local approximations are subsequently converted in local B-spline form and directly combined with the construction of hierarchical quasi-interpolants described in terms of the truncated basis. The choice of defining local approximations in terms of polynomials instead of splines makes our approach more robust when data with highly varying density or voids are considered. Within the hierarchical framework, this local quasi-interpolation operator is combined with an adaptive strategy that at each refinement step identifies a set of basis functions marked for refinement. In order to do this, we developed a marking strategy to identify the subset of hierarchical basis functions to be substituted by finer ones in the spline hierarchy. This allows us to locally improve the accuracy of the approximation according to the scattered data distribution.

The structure of the paper is as follows. Section 2 presents a brief overview of some related works dealing with scattered data fitting, while Section 3 recalls the definition of THB-splines and related properties. Section 4 introduces the developed least squares local data-dependent quasi-interpolation approach, while Section 5 introduces the strategy for the definition of our final adaptive hierarchical approximation. A selection of numerical experiments for the approximation of real data sets describing different terrain configurations is presented in Section 6. Finally, Section 7 concludes the paper.

## 2 Related works

One possible choice for scattered data fitting relies in methods based on radial basis functions (RBFs) since the space naturally depends on the local distribution and density of the input data, see e.g. [8, 30]. Another classic approach to the problem is represented by partition of unity methods, where (suitably chosen) non-negative, compactly supported and linearly independent functions, which form a partition of unity, are combined with local approximants of the data, see, e.g., [1, 8]. Among others, a method that combines adaptive partition of unity with least squares fitting based on radial basis functions was proposed in [22], while [20] presented a scattered data quasi-interpolation scheme based on RBFs.

Focusing on splines, a two-stage method for scattered data fitting by direct extension of local polynomials to bivariate splines was presented in [7] by considering local discrete least squares polynomial approximations [3]. In order to increase the adaptivity of the method, this approach was further developed in [4, 5] by considering hybrid local approximations in terms of polynomials and of linear combinations of radial basis functions in the first stage of the method. In both cases the final approximation is a spline (represented in local Bernstein form) on a regular triangulation (of a suitable extension) of the domain. Scattered data fitting can be naturally addressed with splines on irregular triangulations. Some recent theoretical studies in this context are, for example, [21], where interpolation is proposed for exact data, and [18] where a domain decomposition approach is introduced. In addition to discrete least squares, minimal energy and penalized least squares are considered in the first stage of the method in [18], which can also be listed among the two-stage approaches for scattered data fitting. Furthermore, a quasi-interpolation scheme based on irregular triangulation has also been recently investigated [25]. Multilevel least squares approximation of scattered data over binary triangulations was presented in [14].

Moving to tensor-product splines, a more standard choice in computer aided applications than splines on triangulations, a two-stage method based on extended B-splines with focus on curvilinear domains has been recently introduced in the literature [6]. However, this work provides theoretical results for sufficiently dense data, and, consequently, does not cover real scattered data sets. Interpolation and approximation of scattered 3D data with hierarchical tensor-product B-splines was addressed in [13] by following the hierarchical approach originally proposed in [9]. The method was based on a global least squares minimization with fairing which is also the basic approximation choice adopted in [23, 15]. More precisely, in [23] local tensor-product functions on suitable subdomains are used via repeated knot insertion, while in [15] the THB-spline model is exploited. Interpolation and least square approximation of gridded data with hierarchical splines was also proposed in [10] by taking advantage of the local tensor-product structure of any overlay of the hierarchical spline surface [9]. Note that in the multilevel approach to hierarchical B-splines followed by [9, 10, 13] the use of this kind of global scheme on any refinement level is motivated by the natural assumption of a limited number of degrees of freedom on a single level. When a hierarchical B-spline basis is constructed over the whole domain instead, the solution of the approximation problem is directly computed in the corresponding hierarchical spline space [11, 16]. An adaptive extension of local approximation to hierarchical splines, as the two-stage method here proposed, is then the suitable choice in this case. Partial results related to multilevel B-spline quasi-interpolation of unstructured data were presented in [19]. This refinement construction, however, defines a level-by-level correction in the tensor-product B-spline space that necessarily requires suitable data distributions. Consequently, general scattered data configurations with varying density cannot be covered with this approach. Approximation of large terrain data sets with locally refined B-splines was recently addressed [24].

### 3 THB-splines and hierarchical quasi-interpolation

In this section we briefly introduce hierarchical B-spline spaces and summarize the construction of their truncated basis, which can be conveniently used for the definition of effective hierarchical spline quasi-interpolation operators.

Let

$$V^0 \subset V^1 \subset \dots \subset V^{M-1} \quad \text{and} \quad \{\mathcal{B}_{\mathbf{d}}^0, \mathcal{B}_{\mathbf{d}}^1, \dots, \mathcal{B}_{\mathbf{d}}^{M-1}\}$$

be a sequence of nested  $r$ -variate tensor-product spline spaces  $V^\ell$  defined on a hyper-rectangle  $\Omega \subset \mathbb{R}^r$  and a sequence of corresponding B-spline bases  $\mathcal{B}_{\mathbf{d}}^\ell$  of degree  $\mathbf{d} := (d_1, \dots, d_r)$ , for  $\ell = 0, \dots, M-1$ . In the following we indicate as  $\mathcal{G}^\ell$  the tensor-product grid associated to  $V^\ell$  and as  $\Gamma_{\mathbf{d}}^\ell$  the set of multi-indices with  $r$  components such that

$$\mathcal{B}_{\mathbf{d}}^\ell := \{B_J^\ell, J \in \Gamma_{\mathbf{d}}^\ell\}.$$

We also consider a nested sequence of closed domains

$$\Omega^0 \supseteq \Omega^1 \supseteq \dots \supseteq \Omega^M,$$

where any  $\Omega^\ell \subset \Omega$  is defined as a collection of cells belonging to  $\mathcal{G}^\ell$ , for  $\ell = 0, \dots, M-1$ , and  $\Omega^M = \emptyset$ . We define the hierarchical mesh  $\mathcal{G}_{\mathcal{H}}$  as the set of all the *active* (namely, no further refined) cells belonging to  $\Omega^\ell$ , for each *level*  $\ell = 0, \dots, M-1$ .

The hierarchical B-spline basis is defined by iteratively selecting basis function at different levels, according to the following construction [16, 29].

**Definition 1** *The hierarchical B-spline basis  $\mathcal{H}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}})$  of degree  $\mathbf{d}$  with respect to the mesh  $\mathcal{G}_{\mathcal{H}}$  is defined as*

$$\mathcal{H}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}}) := \{B_J^\ell \in \mathcal{B}_{\mathbf{d}}^\ell : J \in A_{\mathbf{d}}^\ell, \ell = 0, \dots, M-1\},$$

where

$$A_{\mathbf{d}}^\ell := \{J \in \Gamma_{\mathbf{d}}^\ell : \text{supp } B_J^\ell \subseteq \Omega^\ell \wedge \text{supp } B_J^\ell \not\subseteq \Omega^{\ell+1}\}$$

is the active set of multi-indices  $A_{\mathbf{d}}^\ell \subseteq \Gamma_{\mathbf{d}}^\ell$  and  $\text{supp } B_J^\ell$  denotes the intersection of the support of  $B_J^\ell$  with  $\Omega^0$ .

Hierarchical B-splines form a basis for the multilevel space  $S_{\mathcal{H}} := \text{span } \mathcal{H}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}})$  associated to the hierarchical mesh  $\mathcal{G}_{\mathcal{H}}$ .

Let  $s \in V^\ell$  be represented in terms of B-splines of the refined space  $V^{\ell+1}$  as

$$s = \sum_{J \in \Gamma_{\mathbf{d}}^{\ell+1}} \sigma_J^{\ell+1} B_J^{\ell+1}.$$

The truncation operator  $\text{trunc}^{\ell+1} : V^\ell \rightarrow V^{\ell+1}$  with respect to level  $\ell+1$  acts on  $s$  as follows,

$$\text{trunc}^{\ell+1}(s) := \sum_{J \in \Gamma_{\mathbf{d}}^{\ell+1} : \text{supp } B_J^{\ell+1} \not\subseteq \Omega^{\ell+1}} \sigma_J^{\ell+1} B_J^{\ell+1}, \quad \ell = 0, \dots, M-2.$$

The cumulative truncation operator  $\text{Trunc}^{\ell+1} : V^\ell \rightarrow S_{\mathcal{H}} \subseteq V^{M-1}$  with respect to all finer levels in the hierarchy is then defined as

$$\text{Trunc}^{\ell+1}(s) := \text{trunc}^{M-1}(\text{trunc}^{M-2}(\dots(\text{trunc}^{\ell+1}(s))\dots)), \quad \ell = 0, \dots, M-2.$$

Truncated hierarchical B-splines (THB-splines) are defined by exploiting the truncation mechanism [11].

**Definition 2** The truncated hierarchical B-spline basis  $\mathcal{T}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}})$  of degree  $\mathbf{d}$  with respect to the mesh  $\mathcal{G}_{\mathcal{H}}$  is defined as

$$\mathcal{T}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}}) := \{T_J^\ell : J \in A_{\mathbf{d}}^\ell, \ell = 0, \dots, M-1\}, \quad \text{with } T_J^\ell := \text{Trunc}^{\ell+1}(B_J^\ell),$$

and  $B_J^\ell$  is called the mother B-spline of  $T_J^\ell$ .

THB-splines define a strongly stable basis for the hierarchical spline space  $S_{\mathcal{H}}$  with respect to the supremum norm [12]. This means that there exist two constants not dependent on the number of hierarchical levels that can be multiplied to the norm of the coefficients associated to the THB-spline representation of a function  $f$  to bound from below and above the norm of  $f$  itself. This is not the case for the classical hierarchical basis of Definition 1 which is only weakly stable (the associated stability constants have at most a polynomial growth in the number of hierarchical levels) [17].

In addition, THB-splines form a convex partition of unity and preserve the coefficients of their mother functions (they preserve the coefficients of functions represented with respect to one of the bases  $\mathcal{B}_{\mathbf{d}}^\ell$ ) [12]. This last property is indicated as *preservation of coefficients* and enables the construction of *quasi-interpolation* operators in hierarchical spline spaces that do not require additional computations [27]. Let us clarify this important property. We denote by  $V$  a tensor-product spline space of degree  $\mathbf{d}$  defined on  $\Omega^0$  and generated by a B-spline basis  $\mathcal{B}_{\mathbf{d}} := \{B_J, J \in \Gamma_{\mathbf{d}}\}$  defined over a certain grid  $\mathcal{G}$ . Let  $Q : S(\Omega^0) \rightarrow V$  be a quasi-interpolation operator of the form

$$Q(f) := \sum_{J \in \Gamma_{\mathbf{d}}} \lambda_J(f) B_J, \quad (1)$$

for a suitable space  $S$ . Each functional  $\lambda_J : S(\Omega^0) \rightarrow \mathbb{R}$  is locally defined and can be of discrete or continuous type. Thus, if  $Q^\ell(f)$  denotes the instance of  $Q$  when  $V = V^\ell$  and  $\lambda_J^\ell$  the related functionals for  $\ell = 0, \dots, M-1$ , in virtue of the preservation of coefficients, the hierarchical quasi-interpolant  $Q_{\mathcal{H}} : S(\Omega^0) \rightarrow S_{\mathcal{H}}$  is simply defined as

$$Q_{\mathcal{H}}(f) := \sum_{\ell=0}^{M-1} \sum_{J \in A_{\mathbf{d}}^\ell} \lambda_J^\ell(f) T_J^\ell. \quad (2)$$

Note that  $Q_{\mathcal{H}}$  has the capability of reproducing polynomials of degree  $\mathbf{d}$ , whenever  $Q$  has this property. Under mild restrictions on the domain hierarchy, it is also possible to construct operators  $Q_{\mathcal{H}}$  which are projectors onto the hierarchical spline space [27]. Adaptive approximations by different discrete THB-splines quasi-interpolation schemes were recently introduced in the literature [2, 26, 27], by starting from information on  $f$  (and its derivatives in the case of Hermite spline operators) available on a set of gridded points.

The suitable treatment of a *scattered* data set,

$$F = \{(\mathbf{X}_i, f_i), i = 1, \dots, n, \mathbf{X}_i \in \Omega, \text{ with } \mathbf{X}_i \neq \mathbf{X}_j \text{ if } i \neq j\}, \quad (3)$$

requires the design of adaptive schemes able to locally tailor the nature of the fitting method according to the available number of local scattered data points. In order to deal with data of non-trivial complexity, flexible adaptive tools that generate compact representations need to be developed. This requires a versatile definition of local (data dependent) approximations and a low cost construction of the global approximation. We address these issues by: 1) computing local polynomial approximations for defining the operator  $Q$  in (1); 2) extending them to adaptive spline spaces via hierarchical quasi-interpolation. The next two sections detail points 1) and 2).

## 4 Data-dependent local polynomial least squares approximation

In this section we introduce the algorithm for the definition of the quasi-interpolant  $Q$  in (1) by computing the value of any functional  $\lambda_J(f)$ ,  $J \in \Gamma_{\mathbf{d}}$ , associated to the B-spline  $B_J$  for a certain scattered data set  $F$  of the form (3).

We look for a suitable local approximation in the space  $\Pi_J$  of local  $r$ -variate polynomial of total degree  $d_J \leq d$ , with

$$d := \min_{h=1,\dots,r} d_h.$$

For the representation of this local polynomial, we consider the standard (local) power basis suitably translated and scaled in order to map the support of  $B_J$  into the hyper-rectangle  $[0, 1]^r$ . Obviously, this approximation needs to be successively converted in the local spline space  $V_J$  of degree  $\mathbf{d}$  associated with the grid  $\mathcal{G} \cap \text{supp}(B_J)$  in order to set  $\lambda_J(f)$  equal to its  $J$ -th coefficient in the local B-spline basis. By following [7], the polynomial is obtained by least squares approximation of a suitable local subset of data

$$F_J \subset F,$$

initially defined as the set of data  $(\mathbf{X}_i, f_i) \in F$  with  $\mathbf{X}_i$  belonging to the ball of radius  $\rho_J$  equal to the halved diameter of the support of  $B_J$  and with the same center. If  $F_J$  is empty, we enlarge this local subset of data by iteratively replacing  $\rho_J$  with  $k\rho_J$ ,  $k \in \mathbb{N}$ , with  $1 < k \leq K_J$ , where  $K_J$  denotes a positive integer upper bound given in input.<sup>1</sup> Once  $F_J$  is fixed, we initialize  $d_J = d$  and check if the minimal singular value of the collocation matrix associated with the considered local least squares problem is greater or equal than a prescribed threshold  $\sigma$ , with  $0 < \sigma \leq 1$ . If this is not the case,  $d_J$  is decreased by one. Note that, at least when  $d_J$  becomes zero, the minimal singular value is surely greater or equal to  $\sigma$ , since  $\sigma \leq 1$  and the coefficient matrix reduces to a column vector with all unit entries. The use of the threshold  $\sigma$  not only guarantees that the local least squares approximation has a unique solution, but it also ensures that the quality of the polynomial of degree  $d_J$  is comparable with the one of the local best approximation in the same degree polynomial space [3].

For facilitating the comprehension of Algorithm 1, which details the procedure described above, we report in the two following lists the considered notation.

### Input parameters

- $F$ : full set of  $n$  scattered data points, defined as in (3);
- $\mathcal{G}$ : tensor-product mesh associated with the considered global spline space  $V$ ;
- $\sigma (\leq 1)$ : positive threshold prescribing a lower bound for the minimal singular value (MSV) of the local least squares matrix;
- $K_J (\geq 1)$ : integer upper bound which controls the maximum size of the local data sample used to construct the local least squares polynomial approximation.

---

<sup>1</sup>The upper bound  $K_J$  for  $k$  could cause a failure of the algorithm for some  $J$  if the data have a very highly varying density and the mesh  $\mathcal{G}$  is not suitably chosen. In this case, a better choice of  $\mathcal{G}$  is suggested. However, this upper bound is reasonable, since we look for a local approximation.

### Further notation used in Algorithm 1

- $F_J$ : local set of scattered data points;
- $I_J$ : local set of indices between 1 and  $n$  such that  $F_J = \{(X_i, f_i) \in F : i \in I_J\}$ ;
- $|F_J|$ : cardinality of  $F_J$ ;
- $d_J$ : total polynomial degree;
- $\Pi_J$ : local  $r$ -variate polynomial space of total degree  $d_J$ ;
- $|\Pi_J|$ : dimension of  $\Pi_J$ ;
- $\mathcal{P}_J$ : local power basis of  $\Pi_J$ ;
- $A_J$ : local least squares matrix collocating  $\mathcal{P}_J$  at the vertices of  $F_J$ ;
- $\text{MSV}(A_J)$ : minimal singular value of  $A_J$ ;
- $V_J$ : local spline space spanned by the B-splines  $B_I, I \in \Gamma_{\mathbf{d}}$ , whose support intersects  $\text{supp}(B_J)$ .

Finally, the short notation  $\lambda_J$  replaces  $\lambda_J(f)$  to denote the  $J$ -th functional in the remaining part of the paper.

### Algorithm 1

**Input:**

- $F, \mathcal{G}, \mathbf{d} = (d_1, \dots, d_r), J \in \Gamma_{\mathbf{d}}, \sigma \leq 1, K_J \geq 1$ ;
1. set  $d := \min_{h=1, \dots, r} d_h$ ;
  2. initialize  $d_J = d$  and correspondingly  $\Pi_J$  and  $\mathcal{P}_J$ ;
  3. set  $\rho_J := (\text{diam}(\text{supp } B_J))/2$ ;
  4. set  $C_J$  equal to the center of  $\text{supp } B_J$ ;
  5. initialize  $I_J = \{1 \leq i \leq n : \|\mathbf{X}_i - C_J\| \leq \rho_J\}$  and consequently  $F_J$ ;
  6. initialize  $k = 2$ ;
  7. while  $|F_J| = 0$  and  $k \leq K_J$ 
    - (a) update  $I_J = \{i : \|\mathbf{X}_i - C_J\| \leq k \rho_J\}$  and consequently  $F_J$ ;
    - (b) increase  $k$  by one;
  8. if  $|F_J| = 0$  and  $k > K_J$  FAILURE
  9. compute the matrix  $A_J = [p(X_i)]_{p \in \mathcal{P}_J, i \in I_J}$ ;

10. if  $|F_J| < |\Pi_J|$  or  $\text{MSV}(A_J) < \sigma$  then
  - (a) decrease  $d_J$  by 1 and update  $\Pi_J$  and  $\mathcal{P}_J$ ;
  - (b) update the matrix  $A_J = [p(X_i)]_{p \in \mathcal{P}_J, i \in I_J}$ ;
11. compute the least squares approximation  $p_J \in \Pi_J$  of the data in  $F_J$ ;
12. represent  $p_J$  as a function of the local spline space  $V_J$ ,  $p_J = \sum_{B_I \in V_J} \alpha_I B_I$ ;
13. set  $\lambda_J = \alpha_J$ .

**Output:**

- the coefficient  $\lambda_J$ .

We should mention that the possibility of using directly  $V_J$  for determining a local spline approximation instead of a polynomial one has been discarded since not feasible in general. The motivation is that  $V_J$  is a suitable choice only if the projection into  $\Omega$  of  $F_J$  is not larger than the support of  $B_J$ . In this case, step 7 of the algorithm should be removed and a FAILURE of the algorithm is more likely. As confirmed by the numerical experiments, for data sets with highly varying density, this can compromise the benefit of the hierarchical formulation in (2) of  $Q$ , since very few levels and relatively coarse meshes can be used. On the other hand, the choice of a polynomial space  $\Pi_J$  with an adaptively selected total degree  $d_J \leq d$ , together with step 7 when a suitable  $K_J$  is considered, guarantees the success of the hierarchical implementation described in the next section (under the mild requirement of a suitable choice of  $\mathcal{G}^0$  — see also Remarks 1 and 2 below). Note that the numerical tests have also confirmed that for some real world data sets, replacing in the algorithm tensor-product polynomial spaces to total degree ones can produce artifacts. Since it would also be more expensive, we consider only polynomial spaces of total degree.

## 5 Adaptive scattered data quasi-interpolation

If the subdomain hierarchy is given, the quasi-interpolation operator  $Q$  introduced in the previous section and based on Algorithm 1 can be easily extended to a hierarchical spline space using the definition in (2). However, in order to get an effective and fully automatic approximation scheme able to control the maximum approximation error at the data points, while reducing the number of degrees of freedom, a strategy for adaptively defining  $M$  and  $\mathcal{G}_{\mathcal{H}}$  is necessary. This is the goal of Algorithm 2 below which successively modifies the number of levels and the hierarchical mesh by comparing the current values of the errors with a given suitable positive tolerance  $\epsilon$ . Whenever  $\mathcal{G}_{\mathcal{H}}$  is updated, the quasi-interpolant  $Q_{\mathcal{H}}(f)$  needs to be updated. Algorithm 1 is then called for the computation of each new functional  $\lambda_J^\ell$  with  $\ell$  varying between 1 and the current number of levels. If  $\mathcal{G}^0$  is suitably chosen and the maximum number of levels  $M_{max}$  is sufficiently high, the algorithm fails only if a too small tolerance  $\epsilon$  is given in input. Before presenting the algorithm, we need to introduce some further notation,

- $\rho_J^\ell := \text{diam}(\text{supp } B_J^\ell)/2$ , for  $J \in \Gamma_{\mathbf{d}}^\ell$ ,  $\ell = 0, \dots, M-1$ ;
- $\delta^\ell := \max_{J \in \Gamma_{\mathbf{d}}^\ell} \rho_J^\ell$ , for  $\ell = 0, \dots, M-1$ .

In order to simplify the notation, we extend the two previous definitions also to  $\ell = -1$ , by simply considering a B-spline set  $\mathcal{B}_{\mathbf{d}}^{-1} := \{B_J^{-1}, J \in \Gamma_{\mathbf{d}}^{-1}\}$  defined on an auxiliary tensor-product mesh whose size is two times the size of  $\mathcal{G}^0$  and having at least  $(d_1 + 1) \cdots (d_r + 1)$  elements.

In order to define the refined hierarchical mesh in the algorithm, we use a refinement by functions: for each data point where the tolerance is not satisfied, we select all the B-splines  $B_J^\ell$  such that  $T_J^\ell \in \mathcal{T}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}})$  and whose support contains the data point. Then, we subdivide all the active cells (that belong to  $\mathcal{G}_{\mathcal{H}}$ ) of the supports of the selected functions (see Figure 1). Note that refining the supports, by Definitions 1 and 2, implies that we are always replacing some functions in the THB-splines basis  $\mathcal{T}_{\mathbf{d}}(\mathcal{G}_{\mathcal{H}})$  with more functions on a finer level. In other words, this approach guarantees to add degrees of freedom in the areas where we need a further error reduction.

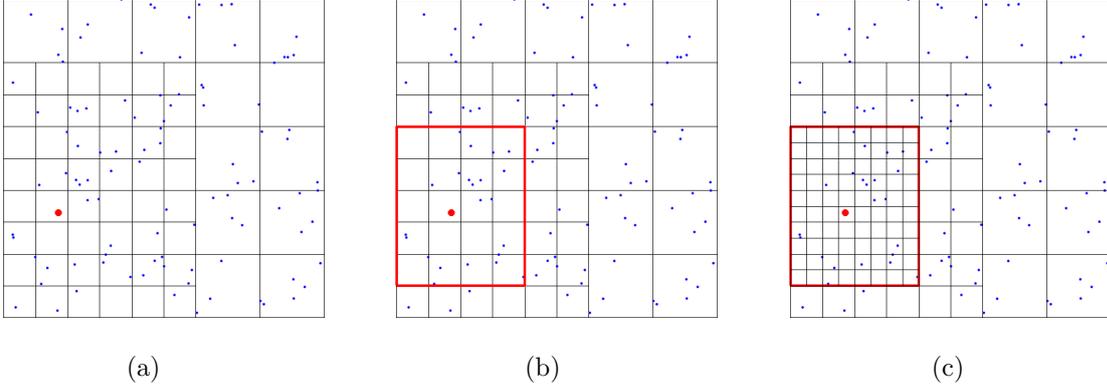


Figure 1: Example of refinement with  $\mathbf{d} = (2, 2)$ : for each data point that does not satisfy the given tolerance (e.g. the red point in (a)), we mark the union of the B-spline supports which contain it (highlighted in (b)), and then subdivide the related cells (c).

## Algorithm 2

### Input:

- $F, \mathcal{G}^0, \mathbf{d} = (d_1, \dots, d_r)$ ;
- maximum number of levels  $M_{max} \geq 2$  and tolerance  $\epsilon > 0$  for adaptive refinement;
- positive threshold  $\sigma \leq 1$  governing the degree selection for each local polynomial approximation.

1. set  $\mathcal{G}_{\mathcal{H}} = \mathcal{G}^0, \mathcal{S}_{\mathcal{H}} = V^0$  and  $M = 1$ ;

2. for each  $J \in \Gamma_{\mathbf{d}}^0$ , compute  $\lambda_J^0$  with Algorithm 1 using as input  $F, \mathcal{G} = \mathcal{G}^0, \mathbf{d}, J \in \Gamma_{\mathbf{d}}^0, \sigma$  and

$$K_J := \left( \left\lceil \frac{2\delta^{-1}}{\rho_J^0} \right\rceil + 1 \right);$$

3. if for some  $\lambda_J^0$  Algorithm 1 ends with a FAILURE, end with a FAILURE;

4. compute the errors

$$e_i := |Q_{\mathcal{H}}(f)(\mathbf{X}_i) - f_i|, \quad i = 1, \dots, n;$$

5. while  $\max_{i=1, \dots, n} e_i > \epsilon$  and  $M \leq M_{max}$

- (a) for  $i = 1, \dots, n$ , if  $e_i > \epsilon$ , mark the functions  $B_J^\ell$ ,  $J \in A_{\mathbf{d}}^\ell$ ,  $\ell = 0, \dots, M - 1$  such that  $\mathbf{X}_i \in \text{supp } B_J^\ell$ ;
- (b) define the new mesh  $\mathcal{G}_{\mathcal{H}}$  and the corresponding space  $S_{\mathcal{H}}$  by performing dyadic refinement of the cells belonging to the support of the marked functions;
- (c) if  $\mathcal{G}_{\mathcal{H}}$  includes cells belonging to  $\mathcal{G}^{M+1}$ , increase  $M$  by one;
- (d) if  $M > M_{max}$ , FAILURE;
- (e) for  $\ell = 0, \dots, M - 1$  and for each new  $J \in A_{\mathbf{d}}^\ell$ , compute  $\lambda_J^\ell$  with Algorithm 1 using as input  $F$ ,  $\mathcal{G} = \mathcal{G}^\ell$ ,  $\mathbf{d}$ ,  $J \in A_{\mathbf{d}}^\ell$ ,  $\sigma$  and

$$K_J := \left( \left\lceil \frac{2\delta^{\ell-1}}{\rho_J^\ell} \right\rceil + 1 \right);$$

(f) update the errors

$$e_i := |Q_{\mathcal{H}}(f)(\mathbf{X}_i) - f_i|, \quad i = 1, \dots, n.$$

**Output:**

- $M \leq M_{max}$  and the hierarchical mesh  $\mathcal{G}_{\mathcal{H}}$ ;
- all the coefficients  $\lambda_J^\ell$ ,  $J \in A_{\mathbf{d}}^\ell$ ,  $\ell = 0, \dots, M - 1$  of the hierarchical quasi-interpolant  $Q_{\mathcal{H}}(f)$ .

The second possible FAILURE of the algorithm in step 5(d) could sometimes be removed by simply increasing  $M_{max}$ . However it could also persist if  $\epsilon$  is too small (but clearly this tolerance can not be arbitrarily small). The following two remarks discuss two important aspects of the algorithm. Remark 1 focuses on the first possible FAILURE of the algorithm and on how it is possible to avoid it. Remark 2 explains why, if step 2 is overcome, then in the “while” cycle the required new functionals can always be computed.

**Remark 1** *Because of the initialization of  $K_J$  in step 2, we cannot guarantee, in general, that it is possible to compute all the coefficients  $\lambda_J^0$  in this step. This motivates step 3 of the algorithm. This initial FAILURE could be avoided by initializing  $K_J$  with a sufficiently high integer value. However, this is not the right solution, since this would require to consider data  $(\mathbf{X}_i, f_i)$  with  $\mathbf{X}_i$  quite far from the supports of the THB-splines. Our experiments confirm that this can significantly worsen (and sometimes also ruin) the accuracy of the quasi-interpolant. In order to avoid this problem, we suggest a more careful choice of  $\mathcal{G}^0$ , which takes into account the chosen degree as well as the data distribution.*

**Remark 2** *The choice of  $K_J$  addresses several issues. First, it was designed to keep it strictly related to the size of the mesh of the level  $\ell$  of the considered  $T_J^\ell$ . In fact, as mentioned in the previous remark, our tests suggests that considering data points too far from the supports of the THB-splines often worsen the accuracy of the quasi-interpolant. Moreover, our setting in step 5(e) guarantees that, at each “while” cycle we have locally enough data points to compute the coefficients associated with the new active  $T_J^\ell$ . First, consider that the new active THB-splines are of levels  $\ell > 0$ . Note then that for the computation of the new  $\lambda_J^\ell$ ,  $\ell > 0$ , we look for data points at most in a ball which includes the supports of the marked functions intersecting the support of  $B_J^\ell$ . Since by Step 4(a) the support of the previously marked functions contains at least one data point, this guarantees that we have locally at least one data point to compute  $\lambda_J^\ell$  (see Figure 2).*

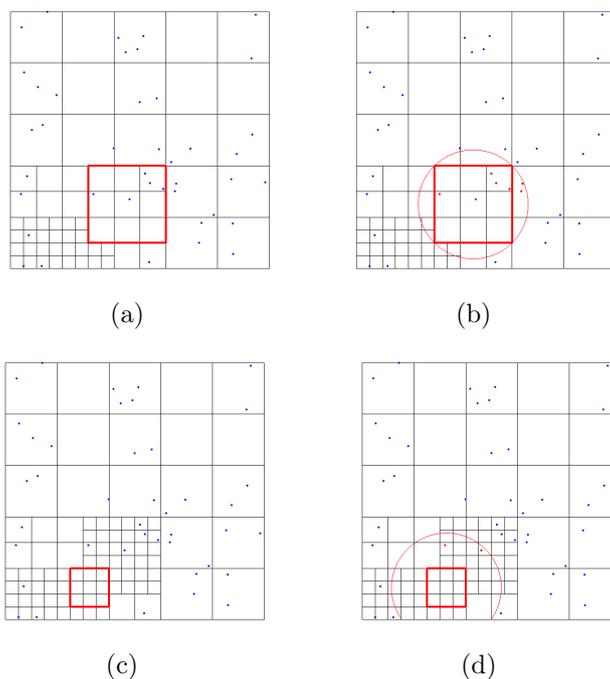


Figure 2: For  $\mathbf{d} = (2, 2)$ , the support of a B-spline  $B_J^\ell$  to be refined is highlighted in (a) while the circle and the points used to determine  $\lambda_J^\ell$  are shown (red) in (b). The support of a new B-spline  $B_I^{\ell+1}$  intersecting the support of  $B_J^\ell$  is highlighted in (c) and, according to step 5(e) of Algorithm 2 and to step 7 of Algorithm 1, the related circle and points used to determine  $\lambda_I^{\ell+1}$  are shown in (d).

## 6 Numerical results

In this section we present the results obtained by applying the hierarchical approximation algorithm to different types of data, including comparisons with other spline approximation methods, when available.

For the tests presented in Examples 1-4, we always use  $\mathbf{d} = (2, 2)$ . If not stated otherwise, we always choose  $M_{max}$  sufficiently large to be able to meet the given tolerance  $\epsilon$ , which is

specified in the corresponding tables (Tables 1,3,5,6).

In the following tests, we report the number of degrees of freedom (NDOF) and the two errors

$$e_{max} := \max_{1 \leq i \leq n} e_i, \quad e_{RMS} := \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}.$$

In all the cases, the results show that the algorithm succeeds at providing accurate approximations of the data satisfying the given tolerance on the maximum error. In Examples 1-2, we also showed that the method favourably compares with respect to previous two-stage methods that consider spline spaces over uniform triangulations. In particular, with our adaptive scheme we get essentially the same maximum error with significantly less degrees of freedom. We note that the  $e_{RMS}$  is instead slightly higher, which is probably natural since the refinement in Algorithm 2 is completely driven by a control based only on the maximum error  $e_{max}$ .

**Example 1** *In the first example we approximate the black forest elevation data set (15885 points) already used for testing scattered data approximation methods in [7] and [5], where two-stage methods with splines over uniform triangulations were considered. This is a case of scattered data with highly varying density, as evident from Figure 3(a). Figure 3(b) clearly shows the adaptive nature of our method, which produces a very refined mesh in the areas of the domain where the data density is high and it is harder to get the required accuracy (they correspond to mountain regions, as it can be seen in Figures 4(a)-(c)). In Table 1, we report the step-by-step results obtained with our method.*

*In Table 2 we compare the performance with two approximants on scattered data defined in spline spaces over triangulations: the first, denoted by  $P$ , is obtained in [7] by extending local polynomial approximation, while the second, denoted by  $H_{MQ}$  is constructed in [5] by using local hybrid polynomial/radial basis approximations.*

$M$	elements of $\mathcal{G}^{M-1}$	NDOF	$e_{max}$	$e_{RMS}$
1	$32 \times 32$	1156	2.680e+02	6.839e+01
2	$64 \times 64$	4058	1.803e+02	3.630e+01
3	$128 \times 128$	12705	7.165e+01	1.240e+01
4	$256 \times 256$	20564	5.349e+01	8.855e+00
5	$512 \times 512$	22759	3.438e+01	8.637e+00
6	$1024 \times 1024$	23092	2.979e+01	8.620e+00

Table 1: Numerical results for Example 1 (black forest data set) with tolerance  $\epsilon = 30.0$  and  $\sigma = 5 \cdot 10^{-2}$ .

method	NDOF	$e_{max}$	$e_{RMS}$
$Q_{\mathcal{H}}$	23092	2.979e+01	8.620e+00
$P$	91526	3.060e+01	3.270e+00
$H_{MQ}$	91526	3.200e+01	2.170e+00

Table 2: Comparison of the performances of  $Q_{\mathcal{H}}$  with the scattered data approximants  $P$  and  $H_{MQ}$ , constructed in [7] and [5], respectively, for Example 1.

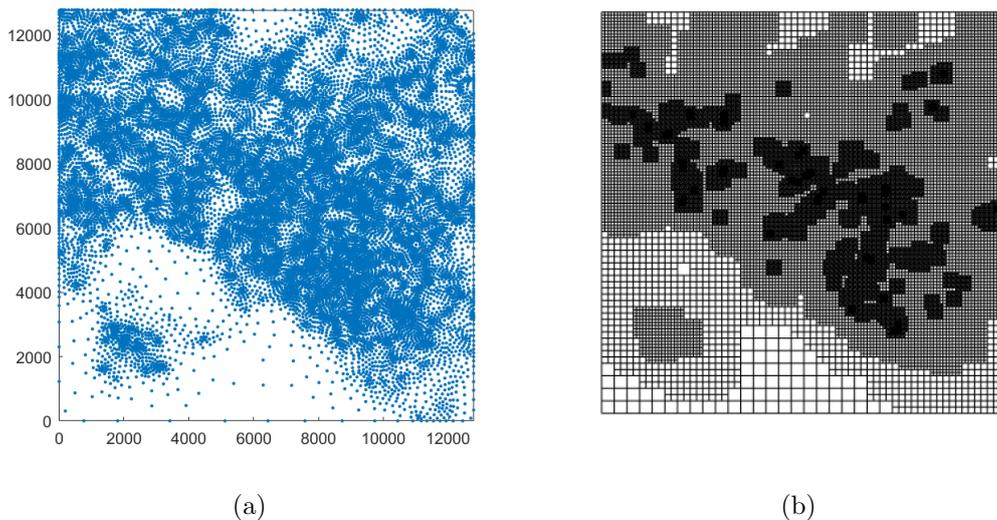


Figure 3: The data points of Example 1 (a) and the hierarchical mesh generated by our approximation algorithm (b).

**Example 2** We consider the glacier data set (8345 points) used for testing scattered data approximation methods in [7] and [5]. We here deal with a quite different type of scattered data, namely contour track data, where the data comes from the sampling along curves of equal elevation, see Figure 6(b). Since in this case the approximation algorithm produces a surface which, while well approximating the data, shows some unwanted oscillations, we consider a slightly different version of the scheme. In particular, we decrease the degree of the local approximations also if their evaluation at the vertices of a suitable local grid returns values too far from the range of the scattered data locally considered. Moreover, this is a case where the data points do not cover a rectangular domain: as evident from Figure 6(b), there are two areas (bottom left and top right corners) without any data points. We then consider a domain obtained by removing two small areas from the corners of the smallest rectangle containing all the data points, see Figure 5. The approximation algorithm is then applied by simply considering as  $V^\ell$ ,  $\ell = 0, \dots, M - 1$  the spline spaces defined on the rectangle containing the domain, and then discarding the B-spline functions whose support does not intersect the domain. Consequently, the hierarchical mesh in Figure 5 shows the mesh limited to the domain. In Table 3 and Table 4, we report the step-by-step results of the test and the comparison with the two approximants on scattered data defined in spline spaces over uniform triangulations,  $P$  and  $H_{MQ}$ , provided in [7] and [5], respectively.

$M$	elements of $\mathcal{G}^{M-1}$	NDOF	$e_{max}$	$e_{RMS}$
1	$16 \times 16$	309	5.784e+01	1.313e+01
2	$32 \times 32$	1002	3.436e+01	6.381e+00
3	$64 \times 64$	2113	2.217e+01	4.174e+00
4	$128 \times 128$	2626	1.647e+01	3.949e+00
5	$256 \times 256$	2736	1.583e+01	3.934e+00

Table 3: Numerical results for Example 2 (glacier data set), with tolerance  $\epsilon = 16.0$  and  $\sigma = 2 \cdot 10^{-1}$ .

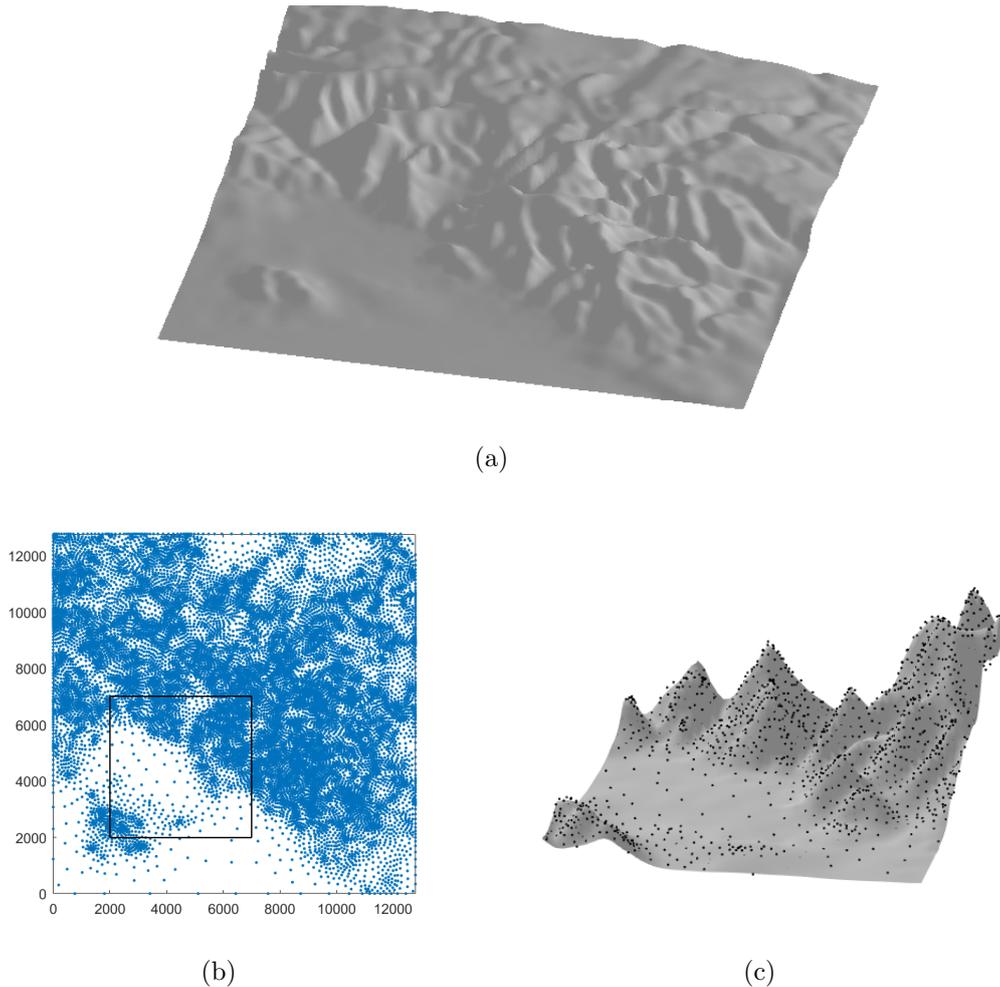


Figure 4: The resulting surface approximating the data set of Example 1 (a); (c) is a zoom of the surface in (a), with the original data represented as black dots, corresponding to the area highlighted in (b).

method	NDOF	$e_{max}$	$e_{RMS}$
$Q_{\mathcal{H}}$	2736	1.583e+01	3.934e+00
$P$	7254	1.870e+01	2.780e+00
$H_{MQ}$	7254	1.560e+01	2.260e+00

Table 4: Comparison of the performances of  $Q_{\mathcal{H}}$  with the scattered data approximants  $P$  and  $H_{MQ}$ , constructed in [7] and [5], respectively, for Example 2.

**Example 3** We consider a subset of the Rotterdam harbour data set used in [7] and [5]. More precisely, we select the data whose location is inside an L-shaped domain, to give a further example of application of the algorithm to non-rectangular domains. In this case the data are affected by noise and by the presence of several outliers. In order to clean the data, we performed a preliminary step, as suggested in [7]. First, we ran our approximation method on the data set using  $M_{max} = 6$  and tolerance  $\epsilon = 6 \cdot 10^{-2}$ . Then, we removed from the data set all the points where the error of the approximation exceeds  $e_{RMS}$ . The resulting data set

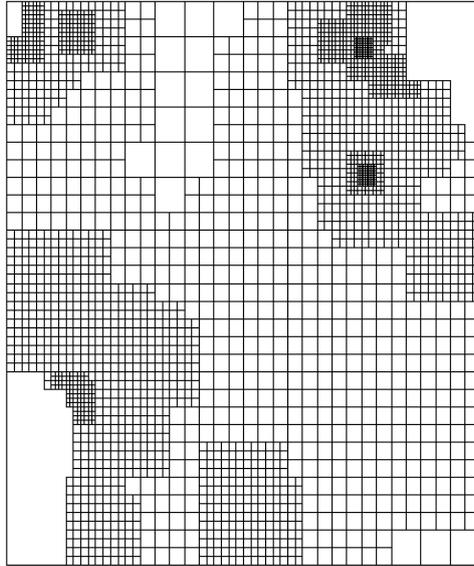


Figure 5: The hierarchical mesh generated by our approximation algorithm for Example 2.

(12250 points) is shown in Figure 7(a). Table 5 presents the results obtained at the different refinement steps and Figure 8 shows the hierarchical spline surface that approximates the given data.

$M$	elements of $\mathcal{G}^{M-1}$	NDOF	$e_{max}$	$e_{RMS}$
1	$8 \times 8$	91	7.742e-01	1.631e-01
2	$16 \times 16$	278	5.054e-01	1.031e-01
3	$32 \times 32$	941	5.258e-01	6.677e-02
4	$64 \times 64$	3451	3.992e-01	4.326e-02
5	$128 \times 128$	11247	3.607e-01	2.867e-02
6	$256 \times 256$	17018	1.614e-01	2.498e-02
7	$512 \times 512$	18095	9.574e-02	2.469e-02
8	$1024 \times 1024$	18218	6.994e-02	2.465e-02

Table 5: Numerical results of Example 3 (Rotterdam harbour data set), with tolerance  $\epsilon = 7 \cdot 10^{-2}$  and  $\sigma = 5 \cdot 10^{-2}$ .

**Example 4** We present this test to assess the behaviour of our method also on gridded data. We consider a set of  $129 \times 129$  elevation data of a mountain region of the Hawaii Islands (available at [28]). In Table 6 we report the step-by-step results of the test, while Figure 9 shows the hierarchical mesh obtained with the adaptive scheme. The original data surface is illustrated in Figure 10 together with the hierarchical spline approximation and the corresponding contour plots. Note that for exact gridded data sets other quasi-interpolation operators may be more effective.

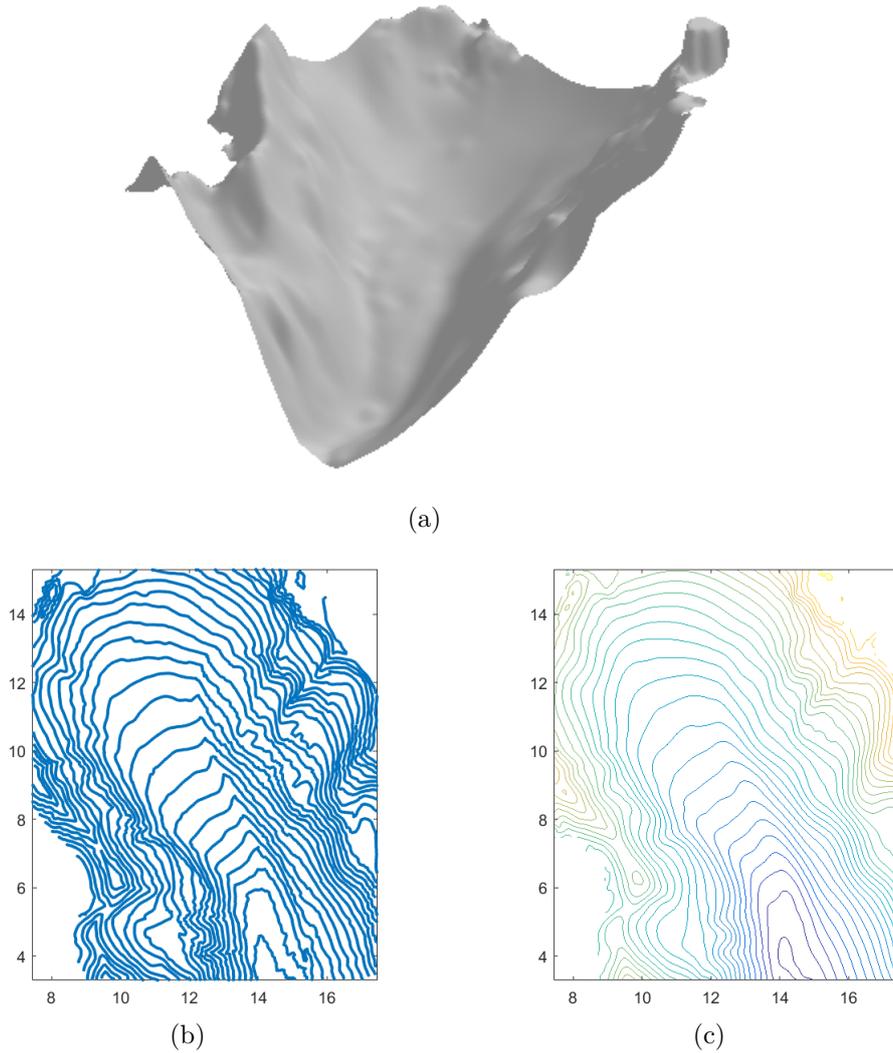


Figure 6: The resulting surface approximating the data set of Example 2 (a); (b) and (c) are the original contour track data and the contour plot of the approximating surface, respectively.

$M$	elements of $\mathcal{G}^{M-1}$	NDOF	$e_{max}$	$e_{RMS}$
1	$64 \times 64$	4356	1.415e+02	1.051e+01
2	$128 \times 128$	16065	7.869e+01	6.109e+00
3	$256 \times 256$	36933	5.262e+01	4.400e+00
4	$512 \times 512$	59706	1.000e+01	2.875e+00

Table 6: Numerical results for Example 4 (Hawaii data set), with tolerance  $\epsilon = 10.0$  and  $\sigma = 10^{-4}$ .

The adaptive nature of Algorithm 1 is outlined in Table 7 where the information concerning the degrees of local polynomials for Examples 1–4 based on biquadratic splines is reported.

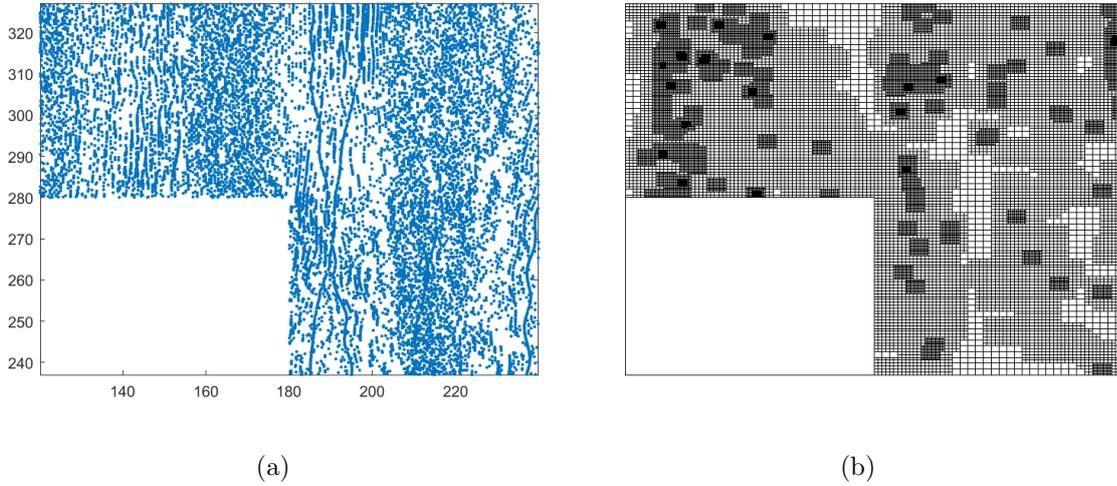


Figure 7: The data points of Example 3 (a) and the hierarchical mesh generated by our approximation algorithm (b).

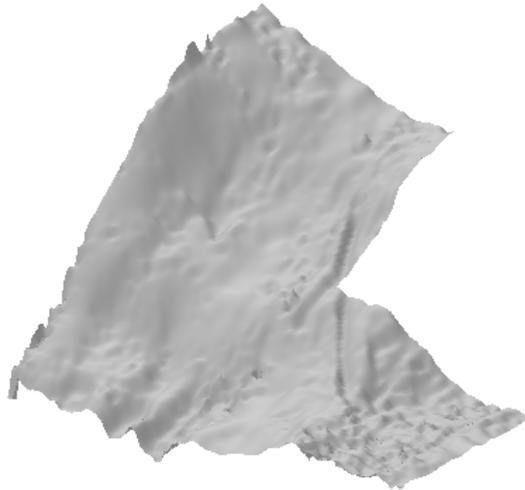


Figure 8: The resulting surface approximating the data set of Example 3.

	NDOF	degree 0	degree 1	degree 2
example 1	23092	12.186%	24.043%	63.771%
example 2	2736	18.469%	22.348%	59.183%
example 3	18218	23.234%	29.793%	46.973%
example 4	59706	47.459%	35.413%	17.127%

Table 7: Percentage of polynomials of degrees 0, 1, 2 computed with Algorithm 1 for Examples 1-4.

In these experiments, biquadratic splines proved to be the best compromise between the need to keep the approximation local and the possibility of considering polynomials of a suitable (high) degree.

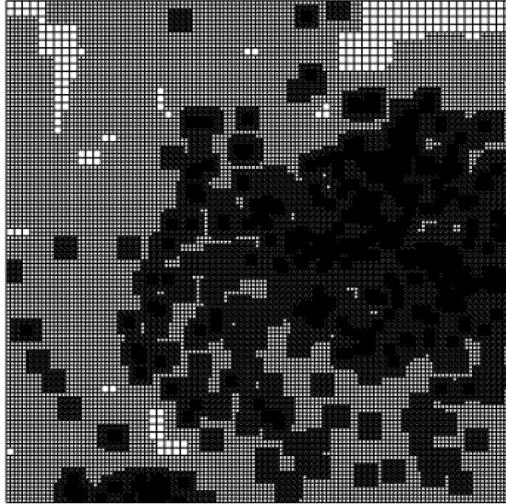


Figure 9: Hierarchical mesh generated by our approximation algorithm in Example 4.

**Example 5** *In the last example, the set  $F$  defined in (3) is obtained by sampling the peak function  $f(x, y) = 2 / (3 \exp((10x - 3)^2 + (10y + 3)^2))$  for  $(x, y) \in [-1, 1]^2$  on the set of 16000 scattered data locations shown in Figure 11 (left). When considering  $\sigma = 10^{-6}$ ,  $\epsilon = 2 \cdot 10^{-3}$ ,  $M_{\max} = 7$ , and an initial uniform mesh with  $15 \times 15$  elements, with  $\mathbf{d} = (2, 2)$  the scheme does not meet the given tolerance  $\epsilon$ . This problem is resolved by considering  $\mathbf{d} = (4, 4)$  which allows us to obtain the final approximation with 2390 degrees of freedom distributed on four levels. Polynomials of degree 4 are always selected in this case. The corresponding hierarchical mesh is shown in Figure 11 (right). Clearly, the situation can remarkably vary according to the initialization of the input parameters.*

**Remark 3** *Both the theoretical framework of (truncated) hierarchical B-splines and the corresponding algorithms here presented are not restricted to uniform knot configurations but also cover the case of arbitrary knot distributions. Since the adaptive nature of the mesh is already realized with the hierarchical construction, the common practice working with (T)HB-splines relies on starting with a uniform tensor-product grid, for then successively identifying the different levels of resolution via dyadic refinement. Nevertheless, it is possible to start with a non-uniform initial grid and refine every marked cell by performing an arbitrary splitting into subcells. Obviously, the resulting implementation can become unnecessarily complicated, also due to the arbitrary choice of the splitting. As a compromise, we consider a test where we start with a non-uniform mesh for then performing dyadic refinement. Figure 12 (b) shows the mesh obtained with  $\mathbf{d} = (2, 2)$  for the data set considered in Example 5 starting with the non-uniform mesh with  $15 \times 15$  elements reported on Figure 12 (a). In this case, by considering  $\sigma = 10^{-6}$  and  $\epsilon = 5 \cdot 10^{-2}$ , the tolerance  $\epsilon$  is met in two levels with 370 degrees of freedom. Repeating the same experiment starting with a uniform mesh with the same number of elements, we need three levels and 553 degrees of freedom.*

## 7 Conclusion

An adaptive scattered data fitting scheme based on hierarchical spline spaces has been introduced by defining a quasi-interpolant whose coefficients are obtained from local polynomial

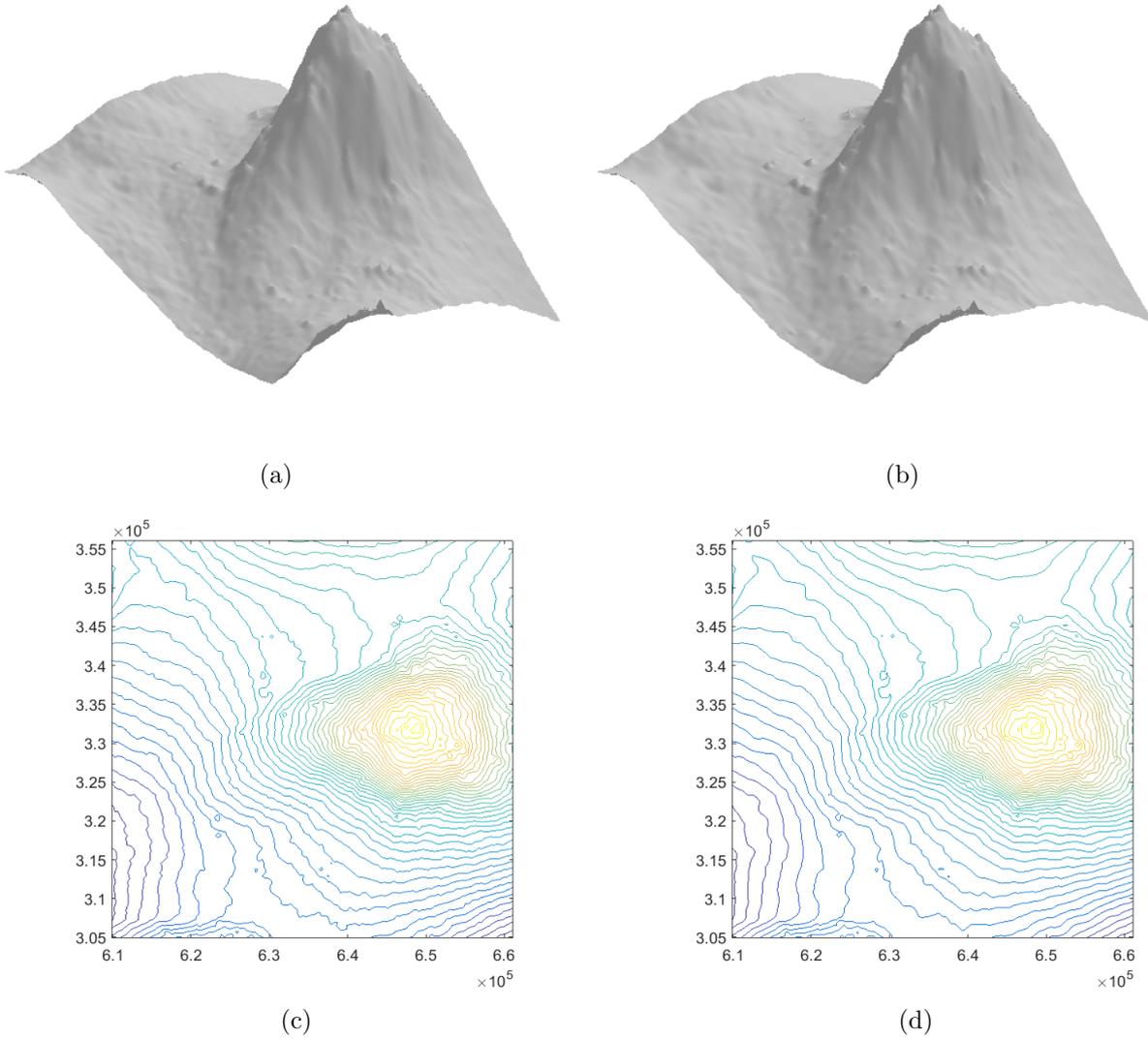


Figure 10: The original data surface of Example 4 (a) and the hierarchical spline approximation (b). The contour plots of the original data (c) and of our approximation (d) are also shown.

approximations of the data. To this aim, local least squares polynomial approximations of variable degree are suitably combined with hierarchical quasi-interpolation. By exploiting the characterization of THB-spline constructions, the coefficients obtained as solution of the local problems can be used to define the global spline approximation without the need of additional computations. The resulting scheme has been validated on several scattered data sets of different nature including structured data, data with highly varying density, data with holes, and also configurations along contour lines. The results confirm that high quality ap-

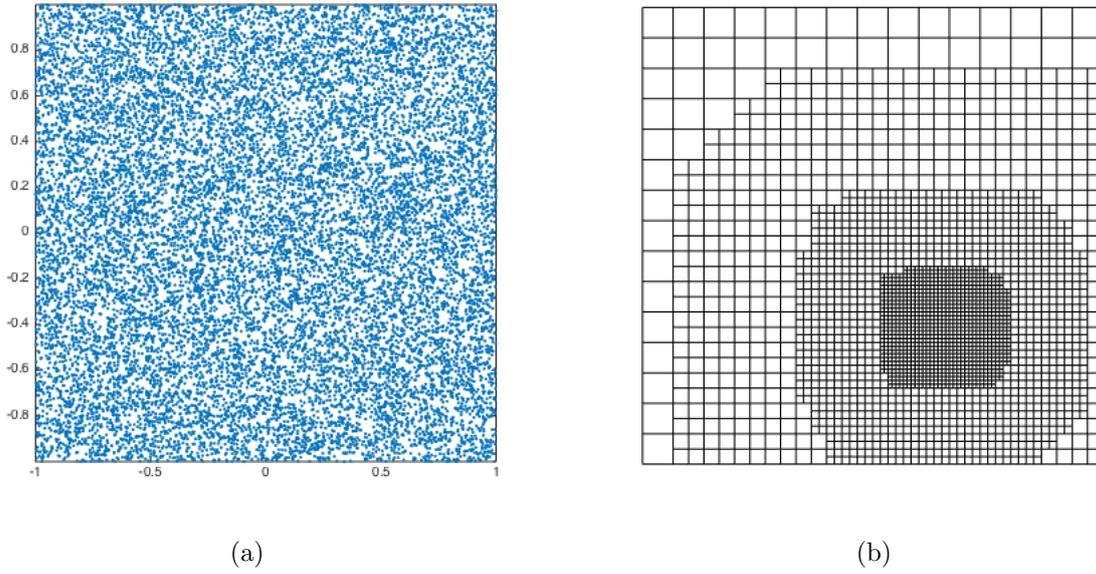


Figure 11: The data point (a) and the hierarchical mesh (b) generated by our approximation algorithm in Example 5 for  $\mathbf{d} = (4, 4)$ .

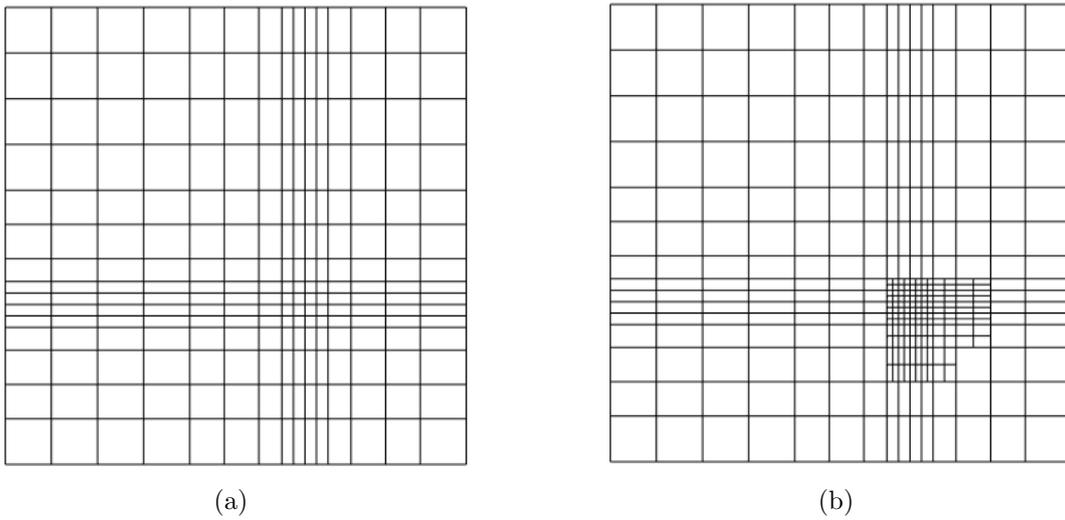


Figure 12: Initial non-uniform mesh (a) and hierarchical mesh on two levels (b) in Remark 3.

proximations can be directly computed via *compact* hierarchical representations which need a reduced number of degrees of freedom compared to previous two-stage methods.

## Acknowledgements

The support by MIUR “Futuro in Ricerca” programme through the project DREAMS (RBFR13FBI3) and by the Istituto Nazionale di Alta Matematica (INdAM) through Gruppo Nazionale per il Calcolo Scientifico (GNCS) are gratefully acknowledged.

## References

- [1] G. Allasia and C. Bracco. Multivariate Hermite-Birkhoff interpolation by a class of cardinal basis functions. *Appl. Math. Comput.*, 218:9248–9260, 2012.
- [2] C. Bracco, C. Giannelli, F. Mazzia, and A. Sestini. Bivariate hierarchical Hermite spline quasi-interpolation. *BIT*, 56:1165–1188, 2016.
- [3] O. Davydov. On the approximation power of local least squares polynomials. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 346–353. University of Huddersfield, UK, 2002.
- [4] O. Davydov, R. Morandi, and A. Sestini. Local RBF approximation for scattered data fitting with bivariate splines. In M. De Bruin, D. H. Mache, and J. Szabados, editors, *Trends and Applications in Constructive Approximation*, volume 151 of *International Series of Numerical Mathematics*, pages 91–102. Birkhäuser / Basel, 2005.
- [5] O. Davydov, R. Morandi, and A. Sestini. Local hybrid approximation for scattered data fitting with bivariate splines. *Comput. Aided Geom. Design*, 23:703–721, 2006.
- [6] O. Davydov, J. Prasiswa, and U. Reif. Two-stage approximation methods with extended b-splines. *Math. Comp.*, 83:809–833, 2014.
- [7] O. Davydov and F. Zeilfelder. Scattered data fitting by direct extension of local polynomials to bivariate splines. *Adv. Comp. Math.*, 21:223–271, 2004.
- [8] G. F. Fasshauer. *Meshfree Approximation Methods with MATLAB*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.
- [9] D. R. Forsey and R. H. Bartels. Hierarchical B-spline refinement. *Comput. Graphics*, 22:205–212, 1988.
- [10] D. R. Forsey and R. H. Bartels. Surface fitting with hierarchical splines. *ACM Trans. Graphics*, 14:134–161, 1995.
- [11] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Comput. Aided Geom. Design*, 29:485–498, 2012.
- [12] C. Giannelli, B. Jüttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comp. Math.*, 40:459–490, 2014.
- [13] G. Greiner and K. Hormann. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In A. L. Méhauté, C. Rabut, and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 163–172. Vanderbilt University Press, Nashville, TN, 1997.
- [14] Ø. Hjelle and M. Dæhlen. Multilevel least squares approximation of scattered data over binary triangulations. *Computing and Visualization in Science*, 8(2):83–91, 2005.
- [15] G. Kiss, C. Giannelli, U. Zore, B. Jüttler, D. Großmann, and J. Barner. Adaptive CAD model (re-)construction with THB-splines. *Graph. Models*, 76:273–288, 2014.

- [16] R. Kraft. Adaptive and linearly independent multilevel B-splines. In A. Le Méhauté, C. Rabut, and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 209–218. Vanderbilt University Press, Nashville, 1997.
- [17] R. Kraft. *Adaptive und linear unabhängige Multilevel B-Splines und ihre Anwendungen*. PhD thesis, Universität Stuttgart, 1998.
- [18] M.-J. Lai and L. L. Schumaker. A domain decomposition method for computing bivariate spline fits of scattered data. *SIAM J. Numer. Anal.*, 47:911–928, 2009.
- [19] B.-G. Lee, J.-J. Lee, and K.-R. Kwon. Quasi-interpolants based multilevel B-spline surface reconstruction from scattered data. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, *Computational Science and Its Applications – ICCSA 2005: International Conference, Singapore, May 9-12, 2005, Proceedings, Part III*, pages 1209–1218. Springer Berlin Heidelberg, 2005.
- [20] S. Liu and C. L. Wang. Quasi-interpolation for surface reconstruction from scattered data with radial basis functions. *Comput. Aided Geom. Design*, 29:435–447, 2012.
- [21] G. Nürnberger, V. Rayevskaya, L. L. Schumaker, and F. Zeilfelder. Local lagrange interpolation with bivariate splines of arbitrary smoothness. *Constr. Approx.*, 23:33–59, 2005.
- [22] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graph. Models*, 68:15–24, 2006.
- [23] C. Rabut. Locally tensor product functions. *Numer. Algor.*, 39:329–348, 2005.
- [24] V. Skytt, O. Barrowclough, and T. Dokken. Locally refined spline surfaces for representation of terrain data. *Comput. Graphics*, 49:58–68, 2015.
- [25] H. Speleers. A family of smooth quasi-interpolants defined over Powell-Sabin triangulations. *Constr. Approx.*, 41:297–324, 2015.
- [26] H. Speleers. Hierarchical spline spaces: quasi-interpolants and local approximation estimates. *Adv. Comp. Math.*, 43:235–255, 2017.
- [27] H. Speleers and C. Manni. Effortless quasi-interpolation in hierarchical spaces. *Numer. Math.*, 132:155–184, 2016.
- [28] U. S. G. Survey. <https://www.usgs.gov/>, [http://dds.cr.usgs.gov/pub/data/nationalatlas/el\\_usa\\_hawaii.bil\\_nt00924.tar.gz](http://dds.cr.usgs.gov/pub/data/nationalatlas/el_usa_hawaii.bil_nt00924.tar.gz).
- [29] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 200:3554–3567, 2011.
- [30] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.