Data-driven quasi-interpolant spline surfaces for point cloud approximation

Andrea Raffo[1,2] and Silvia Biasotti[3]

[1]*Department of Mathematics and Cybernetics, SINTEF, Oslo, Norway.*
[2]*Department of Mathematics, University of Oslo, Oslo, Norway.*
[3]*Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes" CNR, Genova, Italy.*

Abstract

In this paper we investigate a local surface approximation, the *Weighted Quasi Interpolant Spline Approximation* (wQISA), specifically designed for large and noisy point clouds. We briefly describe the properties of the wQISA representation and introduce a novel data-driven implementation, which combines prediction capability and complexity efficiency. We provide an extended comparative analysis with other continuous approximations on real data, including different types of surfaces and levels of noise, such as 3D models, terrain data and digital environmental data.
Keywords: Spline methods, quasi-interpolation, point clouds, noise, data-driven model assessment.

1 Introduction

Due to the recent progress in the acquisition by laser scanners, photogrammetry and diagnostic devices and the popularity of remote sensing technologies, there has been an exponential growth in the availability of data and the need of efficient representations. A good representation model must be, at the same time, *efficient*, that is based on the minimum amount of data, yet *effective*, that is able to support conservative conclusions and keep as much information as possible. To handle large data volumes it is often necessary to adopt approximation strategies so that a single point becomes representative of a region or a set of properties [9, 2]. Given a set of measurements (and their spatial locations), its model approximation has to permit deducing information about the process that generated those data, even at locations different from those at which the measurements were obtained. Surface reconstruction, terrain elevation estimation and the approximation of rainfall and pollution fields over Digital Elevation Models (DEMs) are all examples of spatial data approximation. Moreover, an efficient approximation allows the recovery of the digital representation of a physical shape that commonly contains a variety of properties and defects, such as: geometric features; noise and outliers; perturbations introduced when the acquisition conditions are not optimal (low resolution of instruments, motions, etc.) or different acquisition techniques collect data sets of different resolution (laser scans or photogrammetry); incomplete data (e.g, in the acquisition of broken artefacts or scans of objects partially occluded [21]).

Quasi-interpolation schemes [10, 39] are popular for data approximation because, unlike traditional least squares approximations, they do not require solving a linear system. We use B-splines as basis functions because they are computationally convenient, for instance with respect to the common radial basis functions, as (piecewise) polynomial bases require low-order integration schemes to be exactly computed. The use of piecewise algebraic approximations makes our method suitable also to CAD applications, where B-splines and NURBS are *de facto* the standard tools. Moreover, the treatment of essential boundary conditions is more natural for structured approaches like the spline-based ones than in meshless methods, as it relies on the number of repetition of each knot value.

The wQISA implementation proposed in this paper is specifically designed to define powerful prediction methods from low quality points. Starting from the theoretical wQISA definition given in [37] for a single tensor-product mesh, in this work we derive a multi-level approximation algorithm and provide an extended comparative analysis with other methods in the literature. The main contributions of the paper include: a detailed description of the wQISA method when used for surface approximation; a multi-level approximation algorithm based on a data-driven definition of the weight functions; an extensive comparison of the wQISA outcome with other well-known continuous approximation methods.

The remainder of the paper is organized as follows. Section 2 overviews the literature on data approximation focusing on methods related to continuous surface approximation. Section 3 overviews the wQISA method and its properties, together with a multi-level implementation of the method based on a data-driven mesh refinement strategy. Section 5 extensively compares the wQISA outcome on different real use cases, with respect to a number of well-known approximation methods and a set of performance indicators that are detailed in Section 4. Discussions and concluding remarks are provided in Section 6.

2 Previous Work

The literature on data approximation is vast and we cannot do justice to all contributions. We limit our review to methods whose output satisfies some smoothness requirement, either local or global. For a more complete list of methods related to our use cases, we refer to [4] for a recent survey on surface reconstruction methods, to [20] for an overview on methods for modelling terrain data and to [35] for a comparative analysis of methods for approximating rainfall data.

Meshless methods. Kriging is a geo-statistical approach largely used to approximate remote sensing measurements [34]. In particular, (ordinary) Kriging incorporates correlation information into data approximation through a variogram model. The main limitation of ordinary Kriging is the limited scalability; indeed, Kriging's computation scales quadratically with respect to the number of observations. The Moving Least Square (MLS) approach is largely adopted for surface reconstruction [3, 43, 14]. The basic idea behind MLS is to approximate the surface in the neighbour of a point with the tangent plane in that point. Several variations have been introduced, for instance Feng et al. [13] devised a model that uses multiple curves/surfaces approximation, that allows separating mixed scanning points received from a thin-wall object and consists of a second-order extension of the method in [31]. Other implicit approximation techniques express the data as linear combination of basis elements. For instance, radial basis functions (RBFs) are central for scattered data approximation [17]. The quality of the interpolation depends on the choice of the basis function. Originally introduced as a global method [40, 6], RBFs have been adapted to compact support [15]. A combination of MLS and RBF is presented in [36]. Wavelets and wavelet transforms [8] have been applied for data compression and noise reduction by truncating the wavelet decomposition. Among other techniques, we mention Poisson based methods, which have been applied for example to surface reconstruction from point sets [28].

Mesh-based methods. Tensor-product spline surfaces (piecewise polynomial or NURBS) are a well established representation for modelling smooth shapes (see, for example, [41, 12]). Several local refinement methods have been proposed to overcome the limited mesh adaptivity to the data, such as T-splines [42], locally refined (LR) B-splines [11, 27] and (truncated) hierarchical B-splines [22]. We are mainly interested on methods that preserves the tensor product structure, because its simplicity. Among the others, we mention Forsey and Bartels, who introduced hierarchical B-splines for interpolation and least square approximation of gridded data [16]; Lee et al. [30] defined a multi-level quasi-interpolant, the Multilevel B-spline Approximation (MBA), where the coefficients depends on values of tensor product B-splines defined on lattices. Similarly, Greiner and Horman [24] addressed approximation and interpolation of data by global least squares

over hierarchical tensor-product spline spaces. Kiss et al. [29] adapted this approach to handle THB-splines.

3 Weighted Quasi Interpolant Spline Approximations

Quasi-interpolation is a low-cost and accurate procedure in function approximation theory. The term quasi-interpolation has been interpreted differently according to the context. We follow the Cheney [7] definition of a quasi interpolant as any linear operator L of the form

$$Lf := \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} f(x_i, y_j) g_{i,j}, \tag{1}$$

where $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is a function being approximated, $n_x \in \mathbb{N} \cup \{+\infty\} \ni n_y$, (x_i, y_j) are given *nodes* and $g_{i,j} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ are functions at our disposal. Differently from classic Quasi Interpolation methods that focus to function approximation, the Weighted Quasi Intepolant Spline Approximation (wQISA, [37]) aims at point cloud approximation, where the data are assumed to be affected by noise, outliers or partially missing. In this Section we summarise the wQISA concept and list its theoretical properties. Then, we sketch a data-driven implementation of the wQISA algorithm, which allows to deal with both approximation and prediction problems. Finally, we discuss its computational complexity.

3.1 Preliminary concepts

In a general pipeline for approximation, we envisage that a complex surface is decomposed in multi-charts, e.g., [33, 45] and then the approximated patches are stitched together. In this paper we focus on the quality of local approximations, reserving to future investigations the gluing of multiple patches. From the mathematical point of view, every surface can be locally projected onto a plane by using an injective map and, if locally regular, it can be expressed in local coordinates as $(x, y, z(x, y))$. Given a degree p, a knot vector is said to be $(p+1)$-*regular* if no knot occurs more than $p+1$ times and each boundary knot occurs exactly $p + 1$ times.

Definition 3.1. Let $\mathcal{P} \subset \mathbb{R}^3$ be a point cloud and $\mathbf{p} = (p_x, p_y) \in \mathbb{N}^* \times \mathbb{N}^*$ be a bi-degree. Let $\mathbf{x}$ be a $(p_x + 1)$-regular knot vector with boundary knots $x_{p_x} = a_1$ and $x_{n_x} = b_1$ and let $\mathbf{y}$ be a $(p_y + 1)$-regular knot vector with boundary knots $y_{p_y} = a_2$ and $x_{n_y} = b_2$. The *Weighted Quasi Interpolant Spline Approximation* of bi-degree $\mathbf{p}$ to the point cloud $\mathcal{P}$ over the knot vectors $\mathbf{x}$ and $\mathbf{y}$ is defined by

$$f_w(x, y) := \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \hat{z}_w(x_i^*, y_j^*) \cdot B[\mathbf{x}_i, \mathbf{y}_j](x, y), \tag{2}$$

where $x_i^* := (x_i + \ldots + x_{i+p_x})/p_x$ and $y_j^* := (y_j + \ldots + y_{j+p_y})/p_y$ are the *knot averages*, the expression

$$\hat{z}_w(u, v) := \frac{\sum\limits_{(x,y,z) \in \mathcal{P}} z \cdot w(x, y, u, v)}{\sum\limits_{(x,y,z) \in \mathcal{P}} w(x, y, u, v)} \tag{3}$$

is the *control points estimator* with weight function $w : \mathbb{R}^2 \times \mathbb{R}^2 \to [0, +\infty)$ and $B[\mathbf{x}_i, \mathbf{y}_j]$ denotes the tensor product B-spline of bi-degree $\mathbf{p}$ which is uniquely determined by the *local knot vectors* $\mathbf{x}_i = [x_i, \ldots, x_{i+p_x+1}]$ and $\mathbf{y}_j = [y_j, \ldots, y_{j+p_y+1}]$.

Remark 3.2. *In Definition 3.1, a wQISA depends on the following inputs: a point cloud, a tensor mesh (uniquely defined by a bi-degree and two knot vectors) and a weight function.*

Remark 3.3. *The weight function w defines a* window *around each point (u, v) and the literature is rich of possible choices, including functions with global and local support. Very popular examples for the function w are:*

3

- *Indicator of radius $r > 0$:*

$$w(x, y, u, v) := \mathbb{1}_{||(x-u, y-v)||_2 \leq r}, \tag{4}$$

- *Gaussian of standard deviation σ:*

$$w(x, y, u, v) := e^{-||(x-u, y-v)||_2 / 2\sigma^2}, \tag{5}$$

- *k-Nearest Neighbors (k-NN):*

$$w(x, y, u, v) := \begin{cases} 1/k, & if \ (x, y) \in N_k(u, v) \\ 0, & otherwise \end{cases}, \tag{6}$$

where $k \in \mathbb{N}^*$ and $N_k(u, v)$ denotes the neighborhood of (u, v) defined by the k closest points of the point cloud.

- *Inverse Distance Weight (IDW):*

$$w(x, y, u, v) := \begin{cases} \dfrac{1}{||(x, y) - (u, v)||_2}, & if \ |C_{(u,v)}| = 0 \\ \begin{cases} \dfrac{1}{|C_{(u,v)}|}, & \forall (x, y) = (u, v) \\ 0, & else \end{cases}, & if \ |C_{(u,v)}| \neq 0 \end{cases} \tag{7}$$

where

$$|C_{(u,v)}| := \{(x, y, z) \in \mathcal{P} \ s.t. \ (x, y) = (u, v)\}.$$

Details on how to fix the free parameters, if any, are provided in Section 3.3.3.

Figure 1 shows the stability of wQISA (until failure) against increasing amount of noise and outliers. In these simulations, a k-NN weight is filtered from outliers by using quartiles.
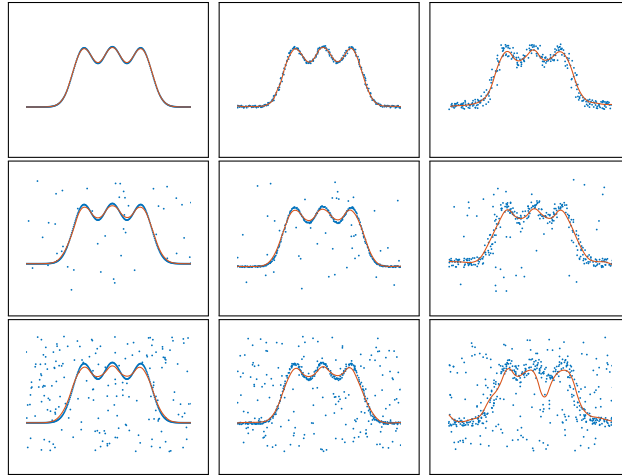

Figure 1: Noise and outliers robustness. The input point cloud is represented in blue, the output approximation is displayed in red. The noise increases from the left to the right. The outliers increase from top to bottom.

3.2 Properties

Similarly to the classical quasi-interpolant schemes [10], the wQISA method satisfies a number of desirable regularity properties, here briefly recalled. For more details and for a probabilistic interpretation of the method, we refer the reader to [37].

Global bounds   Let $\mathcal{P} \subset \mathbb{R}^3$ be a point cloud and $z_{\min}, z_{\max} \in \mathbb{R}$ that satisfy

$$z_{\min} \leq z \leq z_{\max}, \quad \text{for all } (x, y, z) \in \mathcal{P}.$$

Then the weighted quasi interpolant spline approximation to $\mathcal{P}$ from some spline space $\mathbb{S}_{p,[\mathbf{x},\mathbf{y}]}$ and some weight function w has the same (global) bounds

$$z_{min} \leq f_w(x, y) \leq z_{max}, \quad \text{for all } (x, y) \in \mathbb{R}^2. \tag{8}$$

Local bounds   Let $x \in [x_\mu, x_{\mu+1})$ for some μ in the range $p_x + 1 \leq \mu \leq n_x$ and $y \in [y_\nu, y_{\nu+1})$ for some ν in the range $p_y + 1 \leq \nu \leq n_y$. The global bounds of Equation 8 can be refined, by using the B-splines' property of local representation, to

$$\min_{\substack{i=\mu-p_x,\dots,\mu \\ j=\nu-p_y,\dots,\nu}} \hat{z}_w(x_i^*, y_j^*) \leq f_w(x, y) \leq \max_{\substack{i=\mu-p_x,\dots,\mu \\ j=\nu-p_y,\dots,\nu}} \hat{z}_w(x_i^*, y_j^*).$$

We can simplify these bounds to:

$$\min_{(x,y,z)\in\mathcal{P}_{\mu,\nu}} z \leq f_w(x) \leq \max_{(x,y,z)\in\mathcal{P}_{\mu,\nu}} z, \tag{9}$$

where

$$\mathcal{P}_{\mu,\nu} := \bigcup_{\substack{i=\mu-p_x,\dots,\mu \\ j=\nu-p_y,\dots,\nu}} \left\{ \operatorname{supp}\left(w(\cdot, \cdot, x_i^*, y_j^*) \right) \right\} \cap \mathcal{P}.$$

Note that the wQISA bounds depends on the functions w. By choosing weight functions having compact support, one can control these bounds while reducing the number of points required for each control point estimation.

Special configurations, shape preservation and rate of convergence   From basic spline theory, it is straightforward that if $x_{i+1} = \dots = x_{i+p_x} < x_{i+p_x+1}$ and $y_{j+1} = \dots = y_{j+p_y} < y_{j+p_y+1}$ then wQISA interpolates the control point estimate $\hat{z}(x_i^*, y_j^*)$. By assuming (1) any knot in $\mathbf{x}$ and $\mathbf{y}$ to have maximum multiplicity, (2) w to be a 1-NN weight function and (3) the point cloud to consist of samplings of a continuous function $f : [a_1, b_1] \times [a_2, b_2] \to \mathbb{R}$ at the knot averages, we have that f_w corresponds to the *Variation Diminishing Spline Approximation* (VDSA) of f of bi-degree $\mathbf{p}$ to the point cloud $\mathcal{P}$ over the knot vectors $\mathbf{x}$ and $\mathbf{y}$ (see for example [32]). In this perspective, wQISA can be seen as a generalization of VDSA to perturbed data through a wider family of weight functions.

There is also another point of view: wQISA corresponds to applying VDSA to $\hat{z}_w : \mathbb{R}^2 \to \mathbb{R}$. As pointed out in [32], VDSA preserves certain shape properties - such as monotonicity and convexity - of the function being approximated. In our case, wQISA will preserve the monotonicity and convexity of $\hat{z}_w$, i.e., of the average trend of the point cloud, with respect to the chosen weight function. In case of points clouds with defects, the average trend is indeed more important than the position of a point with respect to the others. By considering this analogy between VDSA and wQISA, we conclude that the latter is characterized by a linear convergence (see for example [32] for the rate of convergence of VDSA).

3.3   Data-driven implementation

When dealing with approximation, the quality of a method relies on its prediction capability over independent samples, i.e., on data that has not been used to "train" the model. Whilst for function approximation the accuracy of a method can be quantified by directly comparing the approximation with the original function, in case of point clouds approximation the estimation of the approximation error on new samples is not trivial (for instance, it is often impossible to re-sample the data). Moreover, only a few benchmarks are available. One way to overcome this problem comes from statistical learning, and is here adopted in the form of a learning-based implementation of wQISA. Our implementation consists of four main steps:

Step 1 *Data pre-processing.* The input point cloud is used to generate a *training*, a *validation* and a *test* set, each of which has a specific task in the approximation of the input data. Different strategies are proposed, depending on the size of the input (see Section 3.3.1).

Step 2 *wQISA formulation.* Given a tensor mesh and a weight function, the training set is used to define the control point estimator $\hat{z}_w$. Notice that $\hat{z}_w$ may depend on free parameters to be tuned (see Section 3.3.2).

Step 3 *Parameter settings.* A prediction error is defined on the validation set, in order to fix the possible free parameters of the weight functions 3.3.3).

Step 4 *Model assessment.* Having obtained an approximation model on the basis of the previous steps, the test set is used for estimating the generalization error on new data (see Section 3.3.5).

While the first and the last steps are done only once, the second and the third ones are run in a while-loop, by refining the mesh at each iteration (see Section 3.3.4).

Remark 3.4. *The use of different point clouds for Steps 1-3 in our data-driven implementation is necessary to avoid data overfitting. This is particularly relevant when looking for a model with a good prediction capability, rather than just aiming at minimizing the error on a finite set of points (see, for instance, [26, 38]).*

We provide a simplified flowchart of the algorithm in Figure 2 and a pseudo-algorithm in Algorithm 1, while referring to Sections 3.3.1-3.3.5 for a detailed description of the implemented procedures.
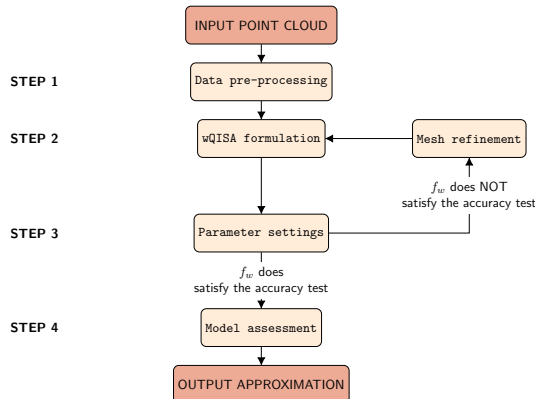

Figure 2: Flowchart of the data-driven wQISA algorithm.

3.3.1 Data pre-processing

The input point cloud $\mathcal{P}$ is used to generate three new sets: a training set $\mathcal{T}$, a validation set $\mathcal{V}$ and a test set $\mathcal{U}$. There is no general rule on how to choose the number of observations in each of the three sets. As suggested in [26], a typical split for large point clouds might be 50% for training set, and 25% each for validation and testing sets. The three sets must have similar data distribution properties, i.e., the noise distribution and the point sampling of the validation and test sets must be comparable to the one of the training set. For instance, the training set $\mathcal{T}$ can be easily obtained in MATLAB from $\mathcal{P}$ by applying the function `pcdownsample` with 50% of sampling rate and then, $\mathcal{V}$ and $\mathcal{U}$ can be obtained with the sample procedure applied to the complement of $\mathcal{T}$. Figure 3 visually shows a possible data partitioning. In order to achieve even more robust results, one can consider several splits of the data into training, validation, and test sets.
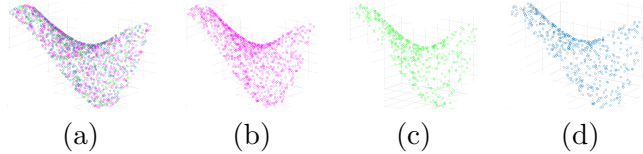
Figure 3: Point cloud partitioning. (a) The original point cloud $\mathcal{P}$; (b) the training set $\mathcal{T}$; (c) the validation set $\mathcal{V}$; (d) the test set $\mathcal{U}$.

In many real-world applications there might be insufficient data to split the set of measurements into three parts. In these cases, we use, the K-fold cross-validation: the input point cloud is split into K roughly equal-sized parts and $K-1$ parts are used to train the model, while the remaining one is used to test it. This process is repeated by changing the part to be used for accuracy evaluation, and then combining the K estimates of the prediction error. In this case, the validation and test sets are the same.

3.3.2 wQISA formulation

The control point estimator is defined by Equation (3). As pointed out in Remark 3.2, the wQISA input consists of:

1. *A point cloud.* We use the training point cloud computed in Section 3.3.1.

2. *A tensor product spline space defined by regular knot vectors.* At the first iteration, one can consider a tensor mesh consisting of a single element with maximum knot multiplicities, with this element being the bounding box of the input data. The mesh is then (globally) refined, iteration-by-iteration, where the approximation shows a lower precision (see Section 3.3.4). Although the spline bi-degree can be considered as an additional parameter, we here choose to focus on C^1 bi-quadratic spline surfaces as they are smooth enough to represent data in a good way [44].

3. *A weight function.* We here decide the weight function to be used on a case-by-case basis. One could nevertheless consider dictionaries of weight functions and then choose the more suitable via an opportune validation error.

Having these inputs fixed, the approximation defined in Equation (2) may still depend on additional parameters to be tuned (see Section 3.3.3).

3.3.3 Parameter settings

The control point estimator depends on the parameters defining a weight function (e.g., k in a k-NN weight), if any. Statistical learning fixes the free parameters by minimizing some prediction error on an independent data set, here represented by the validation set. The reason for not using the training set for parameter settings is the need to avoid overfitting, i.e., a model that has good performances on the input set but has a low prediction capability on new data. We here fix the parameters by minimizing the *Global Mean Squared Error* (GMSE), i.e., the Mean Squared Error (MSE) over the whole validation point cloud. The minimization of the chosen loss function can be undertaken via iterative methods (e.g., stochastic optimization), or by directly evaluating the loss function at a finite number of parameters (e.g., in case of k-NN weight).

3.3.4 Termination criterion

The steps described in the Sections 3.3.2 and 3.3.3 are run in a while loop. The refinement level of a mesh is fixed by considering the last iteration before the GMSE starts increasing (for instance, this occurs when the model starts overfitting the data), with a maximum number of iterations (here we set it to 15). At each iteration of the while loop, we compute a local validation error and use it, together with a user-defined threshold, to decide which

---
Algorithm 1 wQISA pseudocode for k-NN weight

 1: Input: $\mathcal{P}$, $\mathbf{x}$, $\mathbf{y}$, w, ε;
 2: Output: $f_w(\mathbf{x},\mathbf{y})$ (spline approximation), MSE (generalization error)
 3: **return** $\mathcal{T}$, $\mathcal{V}$, $\mathcal{U}$;
 4: **function** WQISA-DEFINITION($\mathcal{T}$, $\mathbf{x}$, $\mathbf{y}$, w)
 5:      Apply Equation 3 and get $f_w(\mathbf{x},\mathbf{y},k)$;
 6:      **return** $f_w(\mathbf{x},\mathbf{y},k)$;
 7: **function** WPARAMETERS($\mathcal{V}$, $f_w(x,y,k)$)
 8:      Find the parameter k that minimizes GMSE;
 9:      Get $f_w(x,y)$ by fixing the optimal parameter k;
10:      **return** $[f_w(x,y)$,GMSE$]$;
11: **function** REFINEMESH($\mathcal{V}$,$\mathbf{x}$, $\mathbf{y}$,$f_w(\mathbf{x},\mathbf{y})$,$\varepsilon$)
12:      Compute LMSE from $\mathcal{V}$ and $f_w(\mathbf{x},\mathbf{y})$;
13:      Check elements where LMSE is above ε;
14:      Split those elements by knot refining both $\mathbf{x}$ and $\mathbf{y}$;
15:      **return** $\mathbf{x}$ and $\mathbf{y}$;
16: **function** GENERALIZATIONERROR($\mathcal{U}$,$f_w(\mathbf{x},\mathbf{y})$)
17:      **return** MSE of $f_w(\mathbf{x},\mathbf{y})$ over $\mathcal{U}$;
18: **function** MAIN$((\mathcal{P}, \mathbf{x}, \mathbf{y}, w, \varepsilon))$
     /*Split the data*/
19: $[\mathcal{T}, \mathcal{V}, \mathcal{U}]$=DataPre-process($\mathcal{P}$);
     /*Define the family of approximations*/
20: $f_w(x,y,k)$=wQISA-Definition($\mathcal{T}$, $\mathbf{x}$, $\mathbf{y}$, w);
     /*Fix the parameters (here, k)*/
21: $[f_w(x,y)$,GMSE$]$=wParameters($\mathcal{V}$, $f_w(x,y,k)$);
     /*Repeat while refining the mesh until the stopping criteria is satisfied*/
22: **while** stopping criteria not satisfied **do**
23:      $[\mathbf{x},\mathbf{y}]$ = RefineMesh($\mathcal{V}$,$\mathbf{x}$, $\mathbf{y}$,$f_w(\mathbf{x},\mathbf{y})$,$\varepsilon$);
24:      $f_w(x,y,k)$=wQISA-Definition($\mathcal{T}$, $\mathbf{x}$, $\mathbf{y}$, w);
25:      $[f_w(x,y)$,GMSE$]$=wParameters($\mathcal{V}$, $f_w(x,y,k)$);
26: MSE=GeneralizationError($\mathcal{U}$,$f_w(\mathbf{x},\mathbf{y})$);
27: **return** $[f_w(\mathbf{x},\mathbf{y}), MSE]$
---

elements should be split by knot insertion. We here insert the knot at the mid-value, in each of the two coordinate directions.

We here consider the *Local Mean Squared Error* (LMSE). Given an element of the mesh, the LMSE on that element is the Mean Squared Error over any validation point whose projection onto the plane (x,y) falls into that element. If no projection lies inside that element, the LMSE for that element is set to zero.

3.3.5   Model assessment

Once the model has been selected, the test set is used to estimate the generalization error. In a data-rich situation, validation and test sets are distinct, and so are in general the validation and test errors. In case the data set was not large enough to be split into three parts, these validation and test sets are equal. This leads the validation and test errors to be the same, and results, in general, in an underestimation of the generalization error (see once more [26]).

3.4   Computational complexity

The computational complexity of a single wQISA iteration depends on the chosen weight function and tensor product spline space. Given the linear space $\mathbb{S}_{\mathbf{p},[\mathbf{x},\mathbf{y}]}$ of tensor product

B-splines of bi-degree $\mathbf{p}$ over the regular knot vectors $\mathbf{x} \in \mathbb{R}^{n_x+p_x+1}$ and $\mathbf{y} \in \mathbb{R}^{n_y+p_y+1}$, there are exactly $\dim(\mathbb{S}_{\mathbf{p},[\mathbf{x},\mathbf{y}]}) = n_x \cdot n_y$ control points to be estimated.

The Gaussian weight is global and thus computes, for a single control point estimation, the linear combination of N addends. Since the weight of any point is at most as expensive as the exponential of an Euclidean norm, the computational complexity of a control point estimate is $O(N)$. To considerably reduce the computational complexity, one can recover local support by composing global weight functions with a k-NN tree (see Equation 6) or with a weight functions with local support, such as the indicator weight function (see Equation 4). Indeed, in case of a k-NN weight function the time needed to compute all the coefficients through k-d trees is proportional to $O(N\log(N))$, where N is the number of points of the cloud [19]. For its efficiency, k-d trees are already adopted for noise point clouds reconstruction, see for instance [23].

The number of iterations to reach the local minimum of the loss function depends on the input point cloud, the chosen approximation method, and the loss function itself (here: the GMSE over the validation point cloud). In these settings, the use of a method with a high convergence rate can cause an overshooting of the minimum. Experimental results on the number of iterations of our method are reported in Section 5 and show that wQISA converges in a reasonably low number of iterations.

n the following, we analyze the CPU times to approximate a point cloud with increasing cardinality, randomly sampled from the mathematical function

$$z = \sqrt{64 - 81((x-0.5)^2 + (y-0.5)^2)}/(9-0.5)$$

over a mesh with an increasing number of knots. We here approximate an IDW weight function by considering, for each knot average, only the K closest points (K is set to 500). We have used, for this test, a prototype implementation developed in Python. These tests run on a 2019 MacBook pro with 2.4 GHz 8-cores Intel®Core$^{\text{TM}}$ i9-processor, resulting in the CPU times shown in the log-log plot of Figure 4.
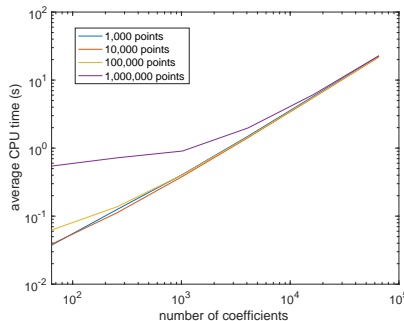

Figure 4: CPU times. Log-log plot of the CPU times when increasing the point cloud cardinality and the number of coefficients.

4 Experimental settings

Despite the popularity of data approximation, we could not find a benchmark able to address all the use cases we are targeting (surface reconstruction, terrain modelling and spatial data measurement approximation); thus we propose a new comparative analysis. In this section we detail the approximation methods and the performance metrics we adopt in our work.

4.1 Approximation methods

We here overview the three popular approximation schemes used as wQISA counterparts throughout this paper: namely, the implicit approximation with Radial Basis Functions

(RBF), Kriging and the Multilevel B-spline Approximation (MBA).

4.1.1 Implicit approximation with radial basis functions

This implicit approximation has the form

$$f(x,y) := \sum_{i=1}^{N} w_i \phi(||(x - x_i, y - y_i)||_2),\tag{10}$$

where the approximating function is represented as a linear combination of *radial basis functions*, each associated with a different center (x_i, y_i) and weighted by an unknown coefficient w_i. The weights w_i can be computed by imposing interpolatory constraints of the data, where the trivial null solution is avoided by adding normal constraints. Depending on the properties of ϕ, RBFs can be locally- or globally-supported. The computational complexity of global and local approximations is respectively $O(N^3)$ and $O(N \log N)$. We here consider the following kernels:

$$\phi(r) := e^{(r/\varepsilon)^2}, \qquad\qquad\qquad \text{(Gaussian)} \tag{11a}$$

$$\phi(r) := \sqrt{1 + (r/\varepsilon)^2}, \qquad\qquad \text{(Multiquadric)} \tag{11b}$$

$$\phi(r) := \frac{1}{\sqrt{1 + (r/\varepsilon)^2}} \qquad\qquad \text{(Inverse multiquadric)} \tag{11c}$$

$$\phi(r) := e^{-\sqrt{r}}, \qquad\qquad\qquad \text{(Modified Gaussian)} \tag{11d}$$

where ε is a shape parameter that here approximates the average distance between nodes. An additional parameter $\alpha \geq 0$ is introduced by performing an L^2-regularization on the interpolation matrix, in order to increase the smoothness of the approximation ($\alpha = 0$ is for interpolation). We consider the implementation from the Python class `interpolate.Rbf`, contained in the `Scipy` library.

4.1.2 Kriging

Kriging owes its importance to its capability to take into account the correlation among input data, which may strongly affect the approximation, e.g., when unevenly distributed. Kriging is defined by the weighted average

$$f(x,y) := \mathbf{w}^t \mathbf{z} = \sum_{i=1}^{N} w_i z_i,\tag{12}$$

where the weights $\mathbf{w} \in \mathbb{R}^N$ are the solution to the linear system $\mathbf{Cw} = \mathbf{c}$, where $\mathbf{C}$ is the covariance matrix of the input point cloud and $\mathbf{c}$ is the vector of covariances between the input points and the prediction point $(x, y) \in \mathbb{R}^2$. Despite its popularity, Kriging suffers several issues when applied to large data sets. For each sample, the ordinary kriging estimator needs to solve a linear system and thus the computational complexity scales quadratically with the number of input points. We use here the Python package `PyKrige`.

4.1.3 Multilevel B-spline approximation

Multilevel B-spline Approximation (MBA) was originally introduced in [30] for approximating scattered data by tensor product B-spline surfaces. A peculiarity of MBA, that makes it analogue to wQISA, is its explicit formulation, which allows it to compute an approximation without solving any equation system. In this respect, MBA can be considered as an example of quasi-interpolation method. At the first step, the tensor product mesh consists of only one element. At each successive iteration, each element is halved in the two coordinate directions. The coefficient c_k of a B-spline B_k is given by

$$c_k = \frac{\sum_i B_k(x_{i,k}, y_{i,k})^2 \phi_i}{\sum_i B_k(x_{i,k}, y_{i,k})^2},$$

where $(x_{i,k}, y_{i,k})_i$ is the set of points within the support of the B-spline B_k and

$$\phi_i := \frac{B_k(x_{i,k}, y_{i,k})z_i}{\sum_l B_l(x_{i,k}, y_{i,k})^2},$$

where the sum in the denominator is taken over all B-splines which contains $(x_{i,k}, y_{i,k})$ in their support. The computational complexity scales linearly with the number of input points. We here use the implementation of the Geometry Group at SINTEF ICT, available at `https://github.com/orochi663/MBA`.

4.2 Evaluation measures

To evaluate the quality of an approximation against the input point cloud, we need to define some error measures. We here present a comparison with a number of measures, each one able to highlight different approximation aspects.

4.2.1 Punctual error and its statistics

Given an approximation defined on the training point cloud, the *absolute punctual error* is computed on the validation point cloud. The absolute punctual error is studied via measures of central tendency (mean) and of statistical dispersion (standard deviation and mean square error).

4.2.2 Hausdorff distance and L^∞ norm

The *Hausdorff distance* (or *two-sided Hausdorff distance*) d_{Haus} between two non-empty subsets A and B in $(\mathbb{R}^d, ||\cdot||_2)$ is given by

$$d_{\text{Haus}}(A, B) := \max\left\{\sup_{a \in A} \inf_{b \in B} ||a - b||_2, \sup_{b \in B} \inf_{a \in A} ||a - b||_2\right\},$$

where sup represents the supremum and inf the infimum. In our case, A is chosen to be the validation point cloud, while B is the image of an approximation f over its domain. In our settings, the sup and inf can be replaced by max and min respectively. We will use this distance for all sets of points embedded in an Euclidean space. For data measurements that associate a scalar value to a point of a grid we will use instead the L^∞ norm.

5 Experimental simulations

In our experiments we consider real data coming from different use cases. We classify the data as affected by different levels of noise: low (e.g., 3D point clouds from high-quality laser scans), average (e.g., terrains from Lidar or sonar acquisition) and high (e.g., air pollution and rainfall measurements from sensors and radars). The data size varies from few dozens to hundreds of thousands of points. All data used in our experiments are provided as additional material. To provide a fair comparative analysis - and avoid any method to overfit - a data-driven implementation is used for all the considered techniques. Hence, the approximation parameters and the model validation are kept distinct and performed on the same training and validation datasets for all methods. In addition, the optimal shaping parameter α for the RBF approximations is found through a constrained trust-region method, while for Kriging we always use the default package settings (linear variogram model).

In all Tables, we use the following naming convention: Gaussian kernel (RBF[1]); multiquadric kernel (RBF[2]); inverse multiquadric kernel (RBF[3]); modified Gaussian kernel (RBF[4]).

5.1 Surface reconstruction

As examples of scattered 3D point clouds, we consider three models from the Science and Technology in Archaeology Research Center (STARC, [1]) of The Cyprus Institute, which consist of a digitisation of archaeological fragments. The complete models are shown in Figure 5 (left column): the model labelled as A.1 is part of a vessel; models A.2 and A.3 are parts of votive statues. The models are acquired via high-quality laser scans. For our experiments, we select a region of interest in every model (highlighted in red), which is then up-sampled via a uniform Montecarlo approach; namely, A.1 is sampled with $30,000$ points, A.2 with $47,642$ points and A.3 with $29,843$ points. We here test wQISA with the Gaussian weight functions of Equation 5; the standard deviation is $\sigma = 10^{-4}$ for the model A.1 and $\sigma = 10^{-3}$ for A.2 and A.3.
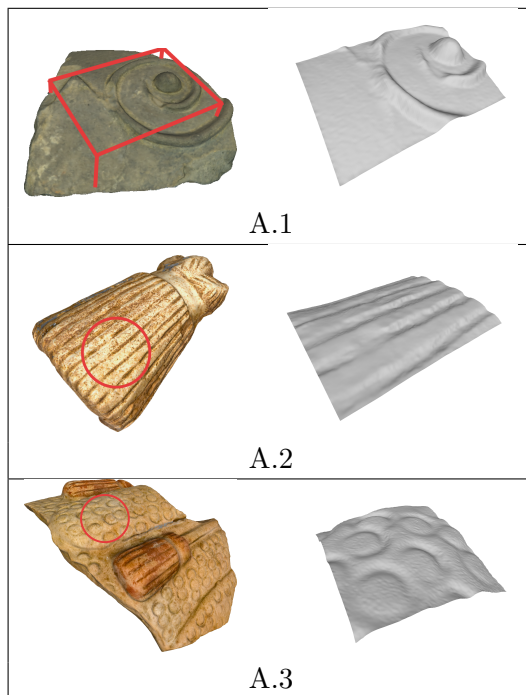

Figure 5: Left sides: the original models (from STARC repository [1]). Right sides: outcomes of the wQISA method via Gaussian weights for the areas highlighted in red.

Figure 5 (right column) depicts the local reconstructions obtained via wQISA. The approximations show a correct recovery of the main details of the artefacts, with an accurate shape preservation: in A.1, the symmetry of the bulge at the center of the spiral is maintained; in A.2, we can notice a faithful reconstruction even in presence of small geometric variations, such as chiselings or local reliefs; finally, A.3 shows a reconstruction characterized by circular patterns. Figure 6 depicts the iterative process that leads to the final wQISA approximations and reports the MSE values computed in Step 4 of Algorithm 1.

We refer to Table 1 for a detailed quantitative comparison of the approximation performances of wQISA, RBFs, Kriging and MBA of the models in Figure 5. We highlight in bold the values with the lowest order of magnitude, which correspond to the best performances. This Table also lists the required number of iterations. From these experiments we can conclude that wQISA performs comparatively well with respect to the other methods when the standard deviation and the Hausdorff distance are considered. This is not surprising because the idea behind wQISA is to fit the overall data trend rather that converging to single values. Tables 2 reports the size of the optimal grid for wQISAs and MBAs. The optimal smoothing parameters for the RBF approximations can be found in Table 3.

The robustness of the data-driven implementation to different choices of the training, validation and test sets (see Section 3.3.1) is here assessed by considering ten different data
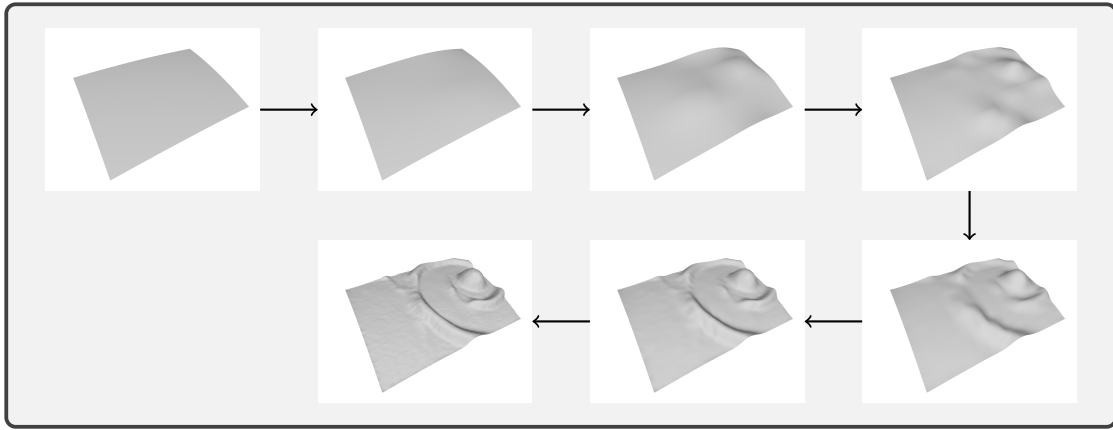
A.1 The MSE over the test set is $1.47 \cdot 10^{-6}$.
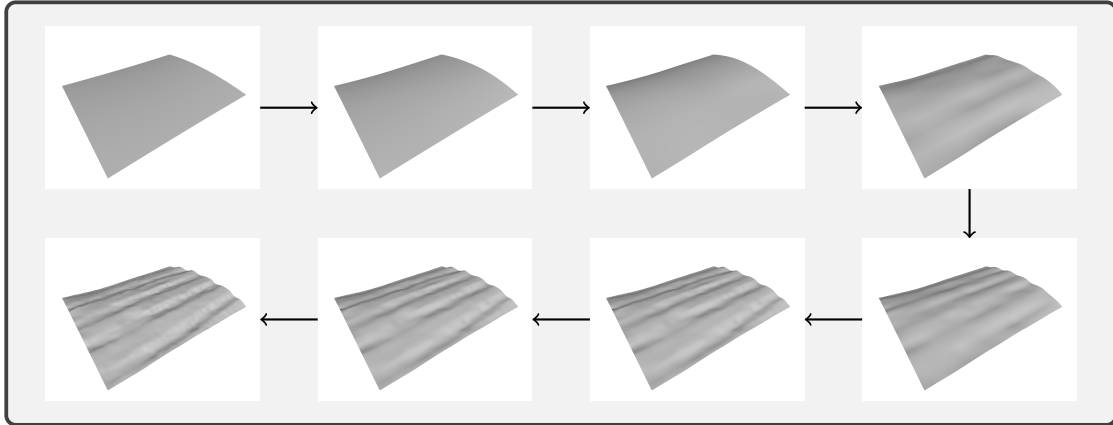
A.2 The MSE over the test set is $2.16 \cdot 10^{-6}$.
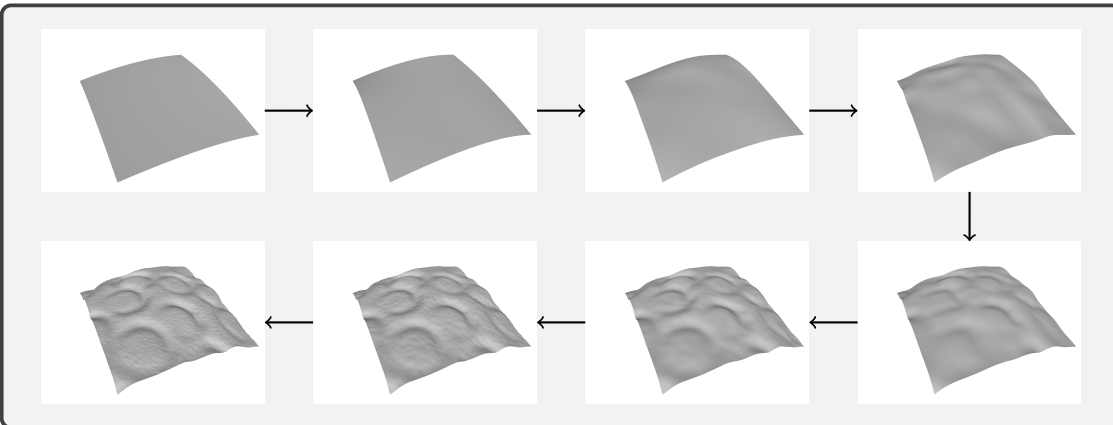
A.3 The MSE over the test set is $6.92 \cdot 10^{-9}$.

Figure 6: Representation step by step of the approximations of the artifacts in Fig. 5 via wQISA with Gaussian weight function.

splits for each 3D model. The resulting MSEs over the respective validation point clouds are graphically represented in the form of boxplots (see Figure 7).

Table 1: Local approximation of 3D models. The labels A.1, A.2 and A.3 refer to the models in Figure 5. In bold, we highlight the errors with the lowest order of magnitude. The single asterisk highlights those cases where the maximum number of iterations, here set to 15, is reached. Kriging runs in a single step and no number is displayed.

		Punctual error			Hausdorff	Number of Iterations
		mean	st.d.	MSE		
A.1	wQISA	$4.445 \cdot 10^{-5}$	$\mathbf{6.881 \cdot 10^{-5}}$	$\mathbf{6.723 \cdot 10^{-9}}$	$\mathbf{9.229 \cdot 10^{-4}}$	7
	RBF^1	$1.194 \cdot 10^{-5}$	$\mathbf{3.399 \cdot 10^{-5}}$	$\mathbf{1.305 \cdot 10^{-9}}$	$9.303 \cdot 10^{-4}$	15*
	RBF^2	$3.94 \cdot 10^{-3}$	$3.052 \cdot 10^{-3}$	$2.480 \cdot 10^{-5}$	$1.968 \cdot 10^{-2}$	13
	RBF^3	$4.496 \cdot 10^{-3}$	$3.635 \cdot 10^{-3}$	$3.335 \cdot 10^{-5}$	$2.436 \cdot 10^{-2}$	15*
	RBF^4	$2.760 \cdot 10^{-3}$	$2.241 \cdot 10^{-3}$	$1.261 \cdot 10^{-5}$	$1.425 \cdot 10^{-2}$	15*
	Kriging	$2.801 \cdot 10^{-5}$	$\mathbf{5.352 \cdot 10^{-5}}$	$\mathbf{3.650 \cdot 10^{-9}}$	$9.307 \cdot 10^{-4}$	-
	MBA	$\mathbf{9.451 \cdot 10^{-6}}$	$\mathbf{3.137 \cdot 10^{-5}}$	$\mathbf{1.071 \cdot 10^{-9}}$	$9.301 \cdot 10^{-4}$	7
A.2	wQISA	$7.046 \cdot 10^{-4}$	$\mathbf{9.590 \cdot 10^{-4}}$	$1.420 \cdot 10^{-6}$	$\mathbf{6.670 \cdot 10^{-2}}$	8
	RBF^1	9.112	22.504	589.495	$1.351 \cdot 10^3$	15*
	RBF^2	$\mathbf{8.043 \cdot 10^{-5}}$	$\mathbf{2.044 \cdot 10^{-4}}$	$\mathbf{4.826 \cdot 10^{-8}}$	$6.412 \cdot 10^{-2}$	12
	RBF^3	$1.097 \cdot 10^{-2}$	$4.990 \cdot 10^{-2}$	$2.61 \cdot 10^{-3}$	2.264	15*
	RBF^4	$2.199 \cdot 10^{-2}$	$2.973 \cdot 10^{-2}$	$1.361 \cdot 10^{-3}$	1.154	15*
	Kriging	$1.153 \cdot 10^{-3}$	$1.301 \cdot 10^{-3}$	$3.027 \cdot 10^{-6}$	$\mathbf{6.695 \cdot 10^{-2}}$	-
	MBA	$1.214 \cdot 10^{-4}$	$\mathbf{2.503 \cdot 10^{-4}}$	$\mathbf{7.738 \cdot 10^{-8}}$	$\mathbf{6.695 \cdot 10^{-2}}$	13
A.3	wQISA	$1.054 \cdot 10^{-3}$	$\mathbf{9.926 \cdot 10^{-4}}$	$2.081 \cdot 10^{-6}$	$\mathbf{5.640 \cdot 10^{-2}}$	8
	RBF^1	23.964	98.608	$1.030 \cdot 10^4$	$4.342 \cdot 10^3$	15*
	RBF^2	$\mathbf{1.440 \cdot 10^{-4}}$	$\mathbf{2.212 \cdot 10^{-4}}$	$\mathbf{6.958 \cdot 10^{-8}}$	$\mathbf{5.552 \cdot 10^{-2}}$	12
	RBF^3	$3.008 \cdot 10^{-2}$	$4.139 \cdot 10^{-2}$	$2.617 \cdot 10^{-2}$	7.891	15*
	RBF^4	$8.760 \cdot 10^{-2}$	$1.189 \cdot 10^{-1}$	$2.179 \cdot 10^{-2}$	7.066	15*
	Kriging	$\mathbf{1.510 \cdot 10^{-4}}$	$\mathbf{1.601 \cdot 10^{-4}}$	$\mathbf{4.845 \cdot 10^{-8}}$	$\mathbf{5.527 \cdot 10^{-2}}$	-
	MBA	$\mathbf{1.635 \cdot 10^{-4}}$	$\mathbf{2.185 \cdot 10^{-4}}$	$\mathbf{7.446 \cdot 10^{-8}}$	$\mathbf{5.506 \cdot 10^{-2}}$	10

Table 2: Size of the tensor mesh for wQISA and MBA.

Method	A.1	A.2	A.3
wQISA	50×64	128×128	128×128
MBA	128×128	4096×4096	128×128

Table 3: Optimal smoothing parameter α for the RBFs implementations.

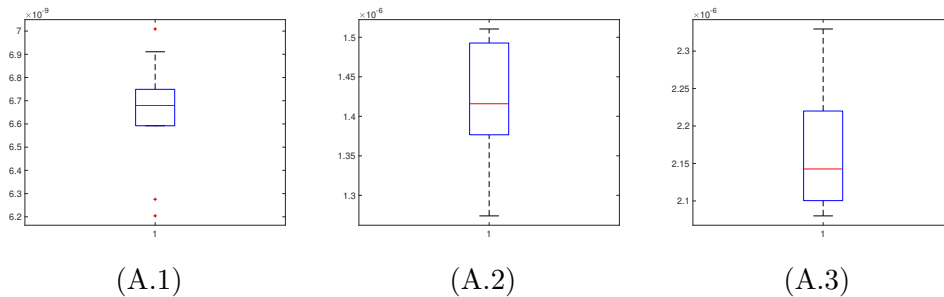| Method	A.1	A.2	A.3
RBF^1	$1.27 \cdot 10^{-3}$	$1.49 \cdot 10^{-1}$	$1.49 \cdot 10^{-1}$
RBF^2	$2.37 \cdot 10^{-1}$	$3.49 \cdot 10^{-2}$	$3.43 \cdot 10^{-2}$
RBF^3	$2.83 \cdot 10^{-1}$	$4.54 \cdot 10^{-2}$	$2.89 \cdot 10^{-1}$
RBF^4	$2.86 \cdot 10^{-1}$	$1.42 \cdot 10^{-1}$	$1.42 \cdot 10^{-1}$

(A.1)   (A.2)   (A.3)

Figure 7: Robustness to different training, validation, and test sets for the models in Figure 5. We show with box-plots the MSE over 10 validation sets.

5.2 Terrain data modelling

Acquisition methods such as Lidar, photogrammetry and sonar produce huge amounts of data that may be perturbed. Methods to efficiently and effectively handle these point clouds are thus required. Since the Kriging and RBFs implementations we are considering are not tailored for big data and present some scalability limitations, here, we limit our attention to the two local methods: wQISA and MBA.

The first data set comes from the island municipality of Værøy and consists of many islands. The data are provided by the Norwegian Mapping Authority Kartverket, and are freely downloadable at `https://hoydedata.no/LaserInnsyn/`. The original point cloud is pre-processed to remove the data outside the mainland, and reduced to 44,529 points. The peculiarity of this terrain is that alternate regions with high variability and almost flat regions.

The second data set corresponds to submarine sand dunes off the coast of France. The data were obtained using bathymetric surveys and are detailed in [18]. The selected part of the data set contains 759,952 points. The difference in sea depth is about 33.2 m. This terrain was already adopted in [44], it is relatively large and presents slight ripples on a wide wavefront.

For both models, we consider a wQISA representation with IDW weight (see Equation 7). Figures 8 shows the wQISA representation of the first terrain, together with the absolute punctual error. The optimal mesh is reached in 11 iteration (1024×1024 elements) for wQISA and in 10 iterations (512×512 elements) for MBA. Similarly, the approximation and the absolute punctual error of the second terrain are shown in Figure 9. For this model, the optimal mesh for wQISA contains 6400×6400 elements while for MBA it contains 1024×1024 elements.
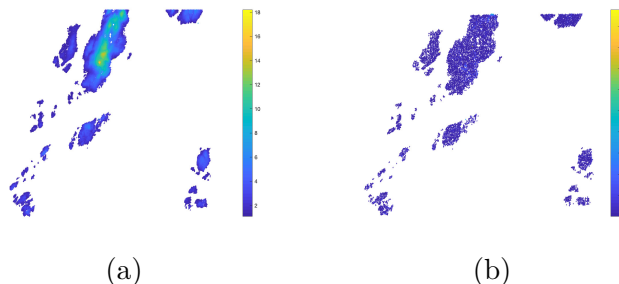

(a)   (b)

Figure 8: Værøy. Figure (a) shows a wQISA of the islands. Figure (b) shows the absolute punctual error.
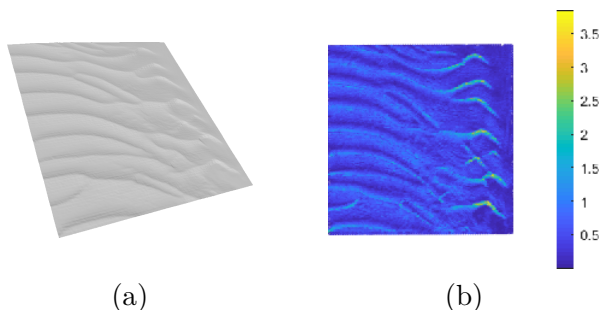

(a)   (b)

Figure 9: Sand dunes. Figure (a) shows a wQISA of the sand dunes (north is pointing upwards to the left). Figure (b) shows the absolute punctual error.

A quantitative comparison of the wQISA and the MBA performances can be found in Table 4. MBA provides a better punctual approximation (highlighted by the MSE measure), but it is again less accurate than wQISA for the overall fit (as highlighted by

Table 4: Terrain data: Værøy (B.1) and sand dunes (B.2).

| | | Punctual error | | | |
		mean	st.d.	MSE	Hausdorff
B.1	wQISA	0.2158	0.2582	0.1132	**9.6266**
	MBA	**0.1513**	**0.2257**	**0.0738**	13.1604
B.2	wQISA	0.0492	0.0641	0.0065	**8.2569**
	MBA	**0.0330**	**0.0543**	**0.0040**	8.2579

the Hausdorff distance).

5.3 Data measurements

We analyse the approximation of strongly perturbed data sets. We consider two precipitation events, with the aim of deepening the comparisons previously presented in [35]. Although focusing here on a specific environmental application, we highlight that these approximation methods do not limit to rainfall field simulations or to a specific sample size. We propose in the additional material the study of a rather small data set (just 11 samples) measuring air pollution. In this case the B-spline approximations rather reduces to linear polynomial approximations, and thus we can consider it as a limiting case.

The first event occurred on September 29, 2013, and was characterized by light rain over Liguria with two different thunderstorms that caused local flooding and landslides. This data set counts only 143 measurements, all acquired by the rain gauge network maintained by Regione Liguria, which is spread over the whole region. A quantitative comparison is provided in Table 5 and is computed by performing a LOO cross-validation on each method. The optimal parameters are here listed:

- *wQISA with k-NN weight.* The optimal number of neighbors k is $k = 2$, where we have checked for any $k = 1, \ldots, 10$. The optimal mesh has 8×8 elements.

- *RBF.* The optimal smoothing parameter α for the RBF approximations is computed by minimizing the mean squared error: $\alpha = 1.00$ for the Gaussian kernel (RBF[1]); $\alpha = 1.22$ for the RBF with multiquadric kernel (RBF[2]); $\alpha = 1.00$ for the RBF with inverse kernel (RBF[3]); $\alpha = 34.31$ for the RBF with modified Gaussian kernel (RBF[4]).

- *MBA.* The mean squared error start increasing after the at iteration number 6, with an optimal tensor mesh containing 16×16 elements.

wQISA has the best generalization performances over the data set with respect to standard deviation, mean squared error and L^∞ norm. This means that wQISA offers more accurate predictions on independent data.

Table 5: Rainfall fields, statistics for the error distribution for the LOO cross-validation.

Method	Mean [mm]	Std [mm]	MSE [mm^2]	L^∞ [mm^2]	Numb. of iterations
wQISA	0.405	**0.565**	**0.483**	**4.952**	4
RBF[1]	4.058	4.720	38.747	39.957	114
RBF[2]	**0.390**	0.587	0.497	5.321	20
RBF[3]	5.472	4.326	48.659	42.285	132
RBF[4]	0.790	0.936	1.500	6.400	26
Kriging	0.474	0.613	0.600	5.668	-
MBA	0.412	0.629	0.562	4.983	5

The second event occurred on January, 2014. The data set contains $19,187$ measurements, gathered from different devices: rain gauges and weather radar. Besides the 143 rain gauges by Regione Liguria, another 25 measure stations by Genoa municipality are considered. The remaining points come from raw radar acquisitions, at first as reflectivity measurements with a range of 400 km. The frequency of mountains over the whole Ligurian territory affects the quality of radar acquisitions and a pre-processing step is needed to remove ground clutter effects; processed data are then combined with observations gathered from rain gauges, which are more reliable measurements but do not cover the whole region. The integration of radar data in the interpolation of the precipitation field makes it possible to extend rainfall fields also to areas surrounding Liguria, and therefore to have a clearer picture about the temporal evolution of precipitation events. Since the temporal interval is different for each acquisition device, rainfall measurements have been cumulated. In this study, a 30 minutes cumulative step has been used (240 time samples). This procedure of integrating gauge and radar data was made to alleviate the well-known error and uncertainty that characterize radar estimates. Spurious signals may be caused, for example, by radar failure or by shielding of the radar beam by mountain ranges [25]. However, various outliers still perturbs the data. The simulation over five time intervals is given in Table 6.

Again, wQISA provides the best generalization performances with respect to standard deviations and mean squared error. The difficulty of handling these datasets is reflected by the variability of the L^∞ norm; because of the presence of outliers, the maximum of the absolute differences is largely unstable and, for each dataset, there is a different winner, even if the wQISA is generally one of the best performing methods also in with respect to this measure. The optimal parameters for the RBF approximations are provided in Table 7. The optimal parameters k for the wQISA with k-NN weight function are: $k = 1$ (00:30), $k = 2$ (01:00), $k = 5$ (01:30), $k = 1$ (00:30), $k = 5$ (02:00) and $k = 7$ (02:30).

6   Concluding remarks

The weighted quasi interpolant spline approximation (wQISA) is a simple and robust procedure to obtain a spline approximation of point clouds. This paper presents a data-driven implementation of wQISA [37], inspired to the supervised learning paradigm. The method has been tested on several real-world data from different application domains and different levels of noise. Experiments have shown that wQISA is able to accurately generalize to data different from those used to define it, i.e. it can effectively describe geometric shapes and measurements not only at those points where it was defined. In other words, wQISA can be successfully used also in prediction tasks, as a linear regression model in supervised learning.

When dealing with data affected by low level of noise, the wQISA method offers approximations that are comparable with other well-known methods. However, the experiments show a better adherence of the wQISA surfaces to point clouds, as reflected by the smaller Hausdorff distance. On these relatively error-free models, we have experimentally noticed a convergence with less iterations than the other methods considered. On the one hand, the wQISA method has linear convergence. On the other hand, this becomes an advantage when dealing with strongly perturbed data, as it allows to adapt better to the underlying distribution. Indeed, on very noisy data wQISA shows a very good stability with respect to standard deviation and mean squared error. In the simulation of precipitation events, wQISA outperforms meshless methods (RBF and Kriging), which in [35] showed the best approximation capability. Experiments confirm that wQISA is able to successfully handle hundreds of thousands of points.

To sum up, two conclusions can be drawn from the comparative analysis. First: it is difficult and possibly ill-advised to search for an *absolute* approximation method, because different approximation techniques can provide better results in different contexts and using different evaluation metrics. Second: for the studied point clouds, the experimental

Table 6: Rainfall fields, statistics for the error distribution.

	Method	Mean [mm]	Std [mm]	MSE [mm^2]	L^∞ [mm]	Numb. of iterations
00:30	wQISA	**0.321**	**0.597**	**0.460**	14.578	7
	RBF[1]	6.193	89.241	$8 \cdot 10^3$	$4.16 \cdot 10^3$	121
	RBF[2]	0.625	0.924	1.244	14.356	80
	RBF[3]	1.549	30.776	949.591	$1.55 \cdot 10^3$	6
	RBF[4]	1.135	1.570	3.753	15.600	23
	Kriging	0.958	1.248	2.476	14.473	-
	MBA	0.675	1.053	1.563	**14.238**	5
01:00	wQISA	**0.270**	**0.589**	**0.419**	11.125	10
	RBF[1]	130.886	225.675	$6.80 \cdot 10^4$	$6.94 \cdot 10^3$	122
	RBF[2]	0.362	2.03	4.985	91.068	122
	RBF[3]	144.857	124.710	$3.65 \cdot 10^4$	903.952	122
	RBF[4]	0.790	0.936	1.500	**6.400**	728
	Kriging	0.964	1.233	2.245	12.449	-
	MBA	0.595	1.009	1.373	12.815	5
01:30	wQISA	0.305	0.647	0.512	10.792	8
	RBF[1]	1.297	3.062	11.055	64.532	6
	RBF[2]	0.302	0.645	**0.509**	10.822	35
	RBF[3]	0.503	1.254	1.826	20.700	7
	RBF[4]	1.109	1.641	3.363	13.334	24
	Kriging	0.474	**0.613**	0.600	**5.668**	-
	MBA	**0.299**	0.653	0.516	10.810	8
02:00	wQISA	**0.457**	**1.078**	**1.370**	28.484	7
	RBF[1]	266.258	246.256	$1.32 \cdot 10^5$	$2.69 \cdot 10^3$	122
	RBF[2]	0.749	4.389	19.822	101.409	122
	RBF[3]	369.316	326.457	$2.43 \cdot 10^5$	**18.775**	137
	RBF[4]	1.283	2.222	6.582	34.145	12
	Kriging	1.197	1.892	5.014	33.140	-
	MBA	0.916	1.560	3.271	31.4356	5
02:30	wQISA	0.423	**0.974**	**1.127**	**16.069**	9
	RBF[1]	36.954	42.373	$3.16 \cdot 10^3$	470.072	124
	RBF[2]	0.432	0.991	1.169	$3.22 \cdot 10^3$	90
	RBF[3]	0.850	3.112	10.408	128.381	11
	RBF[4]	1.544	2.933	10.984	23.973	12
	Kriging	1.776	2.334	8.600	22.421	-
	MBA	**0.391**	1.058	1.271	31.9741	8

Table 7: Optimal smoothing parameter α for the RBFs implementations.

Method	00:30	01:00	01:30	02:00	02:30
RBF[1]	0.00	1.00	0.68	1.00	0.00
RBF[2]	439.64	1.00	0.19	0.99	0.99
RBF[3]	0.00	1.00	0.00	0.59	0.00
RBF[4]	104.84	1.00	0.98	0.24	0.23

results show that data-driven wQISA exhibits a good prediction capability when it comes to strongly perturbed data.

As a further development of the method, we think it is possible to consider adaptive refinement schemes, such as in the case of LR B-splines or THB-splines [11, 5]. This is particularly relevant because these locally refining schemes naturally deal with isogeometric computations and simulation and offers the valuable perspective to practically adopt this work for Computer Aided Design and Manufacturing (CAD/CAM), Finite Element

Analysis and IsoGeometric Analysis [27, 22]. Finally, we intend to extend the comparative analysis to include other real-world applications, methods, and high dimensional data.

Acknowledgments

References

[1] STARC repository. `http://public.cyi.ac.cy/starcRepo/`.

[2] J. Ackermann and M. Goesele. A survey of photometric stereo techniques. *Found. Trends Comput. Graph. Vis.*, 9(3–4):149–254, 2015.

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Point set surfaces. In *Proceedings of the conference on visualization 01, VIS 01, IEEE Computer Society*, pages 21–28, 2001.

[4] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, 36(1):301–329, 2017.

[5] Cesare Bracco, Carlotta Giannelli, David Großmann, and Alessandra Sestini. Adaptive fitting with thb-splines: Error analysis and industrial applications. *Comput. Aided Geom. Des.*, 62:239 – 252, 2018.

[6] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. ACM SIGGRAPH*, pages 67–76, 2001.

[7] E. W. Cheney. *Approximation Theory, Wavelets and Applications*, volume 454, chapter Quasi-interpolation, pages 37–45. NATO Science Series, 1995.

[8] Charles K Chui. *An introduction to wavelets*. Elsevier, 2016.

[9] M. DÆhlen and M. Floater. Iterative polynomial interpolation and data compression. *Numerical Algorithms*, 5(3):165–177, Mar 1993.

[10] C de Boor and G.J Fix. Spline approximation by quasi-interpolants. *Journal of Approximation Theory*, 8(1):19 – 45, 1973.

[11] T. Dokken, T. Lyche, and K. F. Pettersen. Polynomial splines over locally refined box-partitions. *Comput. Aided Geom. Des.*, 30(3):331–356, 2013.

[12] Gerald E. Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code.* Academic Press, Inc., USA, 4th edition, 1996.

[13] W. Feng, Z. Yang, and J. Deng. Moving multiple curves/surfaces approximation of mixed point clouds. *Commun Math Stat*, 2(1):107–124, 2014.

[14] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, July 2005.

[15] Michael S. Floater and Armin Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73(1):65 – 78, 1996.

[16] David R. Forsey and Richard H. Bartels. Surface fitting with hierarchical splines. *ACM Trans. Graph.*, 14(2):134–161, April 1995.

[17] Richard Franke. Scattered data interpolation: Tests of some method. *Mathematics of Computation*, 38(157):181–200, 1982.

[18] Marcaurelio Franzetti, Pascal Le Roy, Christophe Delacourt, Thierry Garlan, Romain Cancouët, Alexey Sukhovich, and Anne Deschamps. Giant dune morphologies and dynamics in a deep continental shelf environment: Example of the banc du four (Western Brittany, France). *Marine Geology*, 346:17 – 30, 2013.

[19] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, September 1977.

[20] Gabriele Garnero. Comparisons between different interpolation techniques. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5:W3, 2013.

[21] A. Georgopoulos. *Heritage and Archaeology in the Digital Age.* Springer, 2017.

[22] C. Giannelli, B. J uttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon, and J. Špeh. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Comput Methods Appl Mech Eng*, 299:337 – 365, 2016.

[23] Simon Giraudot, David Cohen-Steiner, and Pierre Alliez. Noise-adaptive shape reconstruction from raw point sets. *Computer Graphics Forum*, 32(5):229–238, 2013.

[24] Gnther Greiner and Kai Hormann. Interpolating and approximating scattered 3D data with hierarchical tensor product B-splines, surface fitting and multiresolution methods, 1997.

[25] D L Harrison, S J Driscoll, and M Kitchen. Improving precipitation estimates from weather radar using quality control and correction techniques. *Meteorological Applications*, 7(2):135–144, 2000.

[26] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, second edition, 2009.

[27] K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Comput Methods Appl Mech Eng*, 2014.

[28] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In 4th EG Symp. on Geometry Processing, pages 61–70, 2006.

[29] Gábor Kiss, Carlotta Giannelli, Urška Zore, Bert Jüttler, David Großmann, and Johannes Barner. Adaptive CAD model (re-)construction with THB-splines. *Graphical Models*, 76(5):273 – 288, 2014.

[30] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE T Vis Comput Gr*, 3(3):228–244, 1997.

[31] D. Levin. *Geometric modeling for scientific visualization*, chapter Mesh-independent surface interpolation, pages 37–49. Springer-Verlag, 2003.

[32] T. Lyche and K. Mørken. *Spline Methods Draft*, chapter Tensor Product Spline Surfaces, pages 149–166. University of Oslo, April 2011.

[33] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, July 2003.

[34] M. A. Oliver and R. Webster. Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information Systems*, 4(3):313–332, 1990.

[35] G. Patané, A. Cerri, V. Skytt, S. Pittaluga, S. Biasotti, D. Sobrero, T. Dokken, and M. Spagnuolo. Comparing methods for the approximation of rainfall fields in environmental applications. *ISPRS J. Photogramm. and Remote Sens.*, 127:57–72, 2017.

[36] Giuseppe Patané and Michela Spagnuolo. Local approximation of scalar functions on 3D shapes and volumetric data. *Comput Graph-UK*, 36(5):387 – 397, 2012.

[37] A. Raffo and S. Biasotti. Weighted Quasi Interpolant Spline Approximations: Properties and Applications, 2019.

[38] Z Reitermanova. Data splitting. In *WDS*, volume 10, pages 31–36, 2010.

[39] Paul Sablonnière. Recent progress on univariate and multivariate polynomial and spline quasi-interpolants. In Detlef H. Mache, József Szabados, and Marcel G. de Bruin, editors, *Trends and Applications in Constructive Approximation*, pages 229–245. Birkhäuser Basel, 2005.

[40] V. V. Savchenko, A. A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14(4):181–188, 1995.

[41] L. L. Schumaker. *Spline Functions: Basic Theory*. Cambridge University Press, 2007.

[42] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-Splines and T-NURCCS. *ACM Trans. Graph.*, 22(3):477–484, 2003.

[43] C. Shen, J.F. O'Brien, and J.R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph.*, 23(3):896–904, August 2004.

[44] V. Skytt, O. Barrowclough, and T. Dokken. Locally refined spline surfaces for representation of terrain data. *Computer & Graphics*, 49:58–68, 2015.

[45] T. Sorgente, S. Biasotti, M. Livesu, and M. Spagnuolo. Topology-driven shape chartification. *Comput. Aided Geom. Des.*, 65:13 – 28, 2018.

A   Additional material

When dealing with particularly small data sets, wQISA can still be used for single polynomial approximation. As an example, we consider the 2017 Particulate Matter (PM10) report for the city of Oslo (Norway), which is freely available at the municipal data-bank[1]. Particulates, also known as suspended particulate matter, are microscopic particles of solid or liquid matter suspended in the air. They are classified by size, such as for the particles smaller than 10 μm, called PM10, and the particles less than $2.5\mu m$, called PM2.5. PM10 contains particles that originate from combustion (e.g., heating) and road dust (e.g., car

---

[1]http://statistikkbanken.oslo.kommune.no/webview/

tyres and brakes). As particulates are among the most harmful form of air pollution, their approximation is of great interest.

The data set contains only 11 samples; see in Figure 10 an Oslo map with their position. The evaluation results give insight into the approximation performance in real condition of sparsity and provide an example of a situation where there is insufficient data to split into three parts.
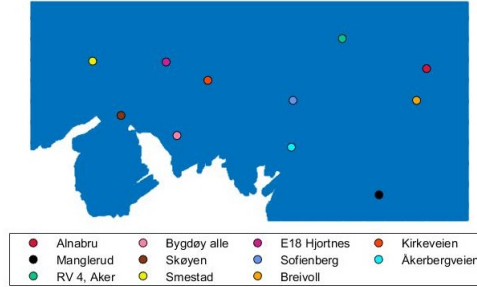

Figure 10: The 11 stations monitoring the PM10 concentration, owned by *Oslo kommune* and *Statens vegvesen*.

Given the small size of the data set, a Leave-One-Out cross-validation was performed to fix the free parameters and compute the statistics for the punctual error. The optimal parameters were computed by minimizing the mean squared error:

- *wQISA with k-NN weight.* Given the low number of points, we fix the restrict our attention to a degenerate tensor mesh containing just one element. The optimal number of neighbors k is $k = 5$, where we have checked for any $k = 1, \ldots, 10$.

- *RBF.* The optimal smoothing parameter α for RBF approximations is computed by minimizing the mean squared error: $\alpha = 679.69 \cdot 10^{-3}$ for Gaussian kernel (RBF^1) in 53 iterations; $\alpha = 190.06 \cdot 10^{-3}$ for RBF with multiquadric kernel (RBF^2) in 51 iterations; $\alpha = 25.00 \cdot 10^{-6}$ for RBF with inverse kernel (RBF^3) in 7 iterations; $\alpha = 983.44 \cdot 10^{-3}$ for RBF with modified Gaussian kernel (RBF^4) in 51 iterations.

- *MBA.* The mean squared error start increasing after the first iteration, thus we adopt a tensor mesh containing just one element as for wQISA.

The results are presented in Table 8. The three best performances with respect to the MSE are given by MBA, wQISA and RBF with Gaussian kernel.

Table 8: Air pollution, statistics for the error distribution for the LOO cross-validation.

Method	Mean [mm]	Std [mm]	MSE [mm^2]	L^∞ [mm]
wQISA	3.005	**1.984**	12.967	7.245
Kriging	3.365	2.020	15.401	7.068
RBF^1	7.626	3.346	77.042	16.525
RBF^2	3.320	2.260	16.130	7.921
RBF^3	3.501	2.531	18.660	7.704
RBF^4	14.731	3.714	230.801	21.193
MBA	**2.576**	2.378	**11.775**	**6.784**