Mathematics and Computers in Simulation 225 (2024) 52–65

Contents lists available at ScienceDirect

Mathematics and Computers in Simulation

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

Original articles

A general formulation of reweighted least squares fitting

Carlotta Giannelli [a], Sofia Imperatore [a,*], Lisa Maria Kreusser [b], Estefanía Loayza-Romero [c], Fatemeh Mohammadi [d], Nelly Villamizar [e]

[a] Dipartimento di Matematica e Informatica, Università degli Studi di Firenze, Viale Morgani 67/A, Firenze, 50137, Italy
[b] Department of Mathematical Sciences, University of Bath, Claverton Down, Bath, BA2 7AY, UK
[c] Department of Mathematics, Imperial College London, South Kensington Campus, London, SW7 2AZ, UK
[d] Departments of Mathematics and Computer Science, KU Leuven, Celestijnenlaan 200A, Leuven, B-3001, Belgium
[e] Departments of Mathematics, Swansea University, Fabian Way, Swansea, SA1 8EN, UK

ARTICLE INFO

Keywords:
Weighted least squares
Interpolation
Fitting
Adaptive splines
Hierarchical splines

ABSTRACT

We present a generalized formulation for reweighted least squares approximations. The goal of this article is twofold: firstly, to prove that the solution of such problem can be expressed as a convex combination of certain interpolants when the solution is sought in any finite-dimensional vector space; secondly, to provide a general strategy to iteratively update the weights according to the approximation error and apply it to the spline fitting problem. In the experiments, we provide numerical examples for the case of polynomials and splines spaces. Subsequently, we evaluate the performance of our fitting scheme for spline curve and surface approximation, including adaptive spline constructions.

1. Introduction

We consider the problem of constructing a continuous model $\boldsymbol{v}: \Omega \to \mathbb{R}^D$ on a general domain $\Omega \subseteq \mathbb{R}^N$ in any dimension $N \in \mathbb{N}_{>0}$ from pointwise data $\{(\boldsymbol{x}_1, \boldsymbol{f}_1), \dots, (\boldsymbol{x}_m, \boldsymbol{f}_m)\}$, where $\boldsymbol{f}_i \in \mathbb{R}^D$ for $D \in \mathbb{N}_{>0}$ are observations at points $\boldsymbol{x}_i \in \Omega$, for $i = 1, \dots, m$. Given the ubiquity of this problem across various fields and applications, a number of methods have been developed to address approximation, data fitting, estimation, and prediction. Examples include classical ones, like interpolation and least squares, while more recently, approaches such as compressive sensing or neural networks, have also emerged as viable options for tackling these challenges. In this work, we primarily concentrate on two very well established classical methods: interpolation and weighted least squares, see, e.g., [13]. Even though interpolation is well established, there is still ongoing research in this field, including the development of efficient algorithms with irregularly-spaced data, see e.g., [7,8], among others. Weighted least squares methods [49] can be considered a more general instance of ordinary least squares methods, studied on weakly admissible meshes in [4]. The key difference between weighted and ordinary least squares is that in the former, fixed weight values are associated with the observations to incorporate different weightings in the least squares scheme.

Most of the time, interpolation and least squares methods are regarded as complementary techniques; however, despite this perception they share a strong connection in their polynomial formulation, as noted in [9]. Additionally, as detailed in [16,18] they can also be used in combination to defeat the Runge phenomenon [48]. In the first part of this paper, we propose a general formulation of the weighted least squares approximant as a convex combination of suitable interpolants, for any finite-dimensional function space consisting of real-valued functions defined on a domain $\Omega \subseteq \mathbb{R}^N$ and address its consequences. As a special case, our

* Corresponding author.
E-mail addresses: carlotta.giannelli@unifi.it (C. Giannelli), sofia.imperatore@unifi.it (S. Imperatore), lmk54@bath.ac.uk (L.M. Kreusser), k.loayza-romero@imperial.ac.uk (E. Loayza-Romero), fatemeh.mohammadi@kuleuven.be (F. Mohammadi), n.y.villamizar@swansea.ac.uk (N. Villamizar).

https://doi.org/10.1016/j.matcom.2024.04.029
Received 27 November 2023; Received in revised form 4 April 2024; Accepted 23 April 2024
Available online 7 May 2024

formulation includes the vector space of polynomials up to a certain degree, for which a relation between weighted polynomial least squares and interpolation has been discussed in [9]. In the second part of the paper, we focus on the weights of the problem. Our main aim is to update these weights in the spirit of the Iterative Reweighted Least Squares (IRLS) method [2,46] with convergence guarantees and efficient algorithms [1,52]. IRLS also offers robust regression [31] and is used to smooth the reconstruction. However, the main difference between our approach and the IRLS method is that our updates allow us to preserve sharp features of the final model and are also suitable for adaptive approximations. Our numerical experiments show the performance of the proposed method within the spline framework, from curve to surface fitting with an appropriate underlying mesh, including adaptive spline constructions.

The paper is organized as follows. In Section 2, we provide a brief introduction to the concepts of interpolation and weighted least squares approximation for a given set of observations. In Section 3 we present our theoretical result and address its consequences. In Section 4, we describe the spline models together with their hierarchical extensions used later on in this paper. Moreover, we present a numerical verification of our theoretical result for the space of polynomial splines and comment on its consequences by exploiting hierarchical box-splines. In Section 5, we describe the reweighted least squares spline fitting scheme and thereafter its extension to deal with adaptive spline constructions. Finally, in Section 6, we report three numerical experiments to evaluate the performance of the proposed reweighted least squares fitting schemes.

2. Interpolation and weighted least squares

Throughout this paper, we assume that $\mathbb{R}^N$, $\mathbb{R}^D$, $\mathbb{R}^n$ and $\mathbb{R}^m$ are column vectors. Let $\Omega \subseteq \mathbb{R}^N$ and consider a set of given observations $\{(\boldsymbol{x}_1, \boldsymbol{f}_1), \dots, (\boldsymbol{x}_m, \boldsymbol{f}_m)\}$, where $\boldsymbol{x}_i = (x_i^1, \dots, x_i^N) \in \Omega$ and $\boldsymbol{f}_i = (f_i^1, \dots, f_i^D) \in \mathbb{R}^D$. Moreover, let V be a vector space of functions defined on Ω and taking values in $\mathbb{R}^D$. We denote its finite dimension by $n = \dim V$ and assume that V is generated by a basis $\Gamma = \{\beta_1, \dots, \beta_n\}$, with basis functions $\beta_j : \Omega \to \mathbb{R}^D$, i.e. $V = \operatorname{span}\{\Gamma\}$.

The *interpolation problem* aims to find an element $\boldsymbol{v} \in V$ such that $\boldsymbol{v}(\boldsymbol{x}_i) = \boldsymbol{f}_i$ holds for each $i = 1, \dots, m$. If a solution $\boldsymbol{v}$ exists, it can be expressed as $\boldsymbol{v} = \sum_{j=1}^n \boldsymbol{c}_j \beta_j$ where the coefficients $\boldsymbol{c}_j = (c_j^1, \dots, c_j^D) \in \mathbb{R}^D$ can be determined by solving a linear system for each component $k = 1, \dots, D$. Specifically, each linear system takes the form $B\boldsymbol{c}^k = \boldsymbol{f}^k$, where

$$B = \begin{pmatrix} \beta_1(\boldsymbol{x}_1) & \dots & \beta_n(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \beta_1(\boldsymbol{x}_m) & \dots & \beta_n(\boldsymbol{x}_m) \end{pmatrix} \in \mathbb{R}^{m \times n} \tag{1}$$

is the $m \times n$ collocation matrix, $\boldsymbol{c}^k = (c_1^k, \dots, c_n^k) \in \mathbb{R}^n$ and $\boldsymbol{f}^k = (f_1^k, \dots, f_m^k) \in \mathbb{R}^m$. The solution $\boldsymbol{v} \in V$ is unique, if and only if B is invertible, hence $n = m$ is a necessary condition. However, even if the interpolant $\boldsymbol{v} \in V$ exists, it is well known that it can be affected by poor approximation quality. Several factors can contribute to this, such as the nature of the data (e.g., their quantity or distribution) and the intrinsic properties of the function space V (e.g., the presence of Runge's phenomenon).

As an alternative to interpolation, a common approach is to define the approximant $\boldsymbol{v} \in V$ as the solution to a *weighted least squares problem* when $n \leq m$. In this context, we assign a strictly positive *weight* value $\omega_i \in \mathbb{R}_{>0}$ to each $\boldsymbol{x}_i \in \Omega$ for $i = 1, \dots, m$. Let $W \in \mathbb{R}^{m \times m}$ be the associated diagonal matrix with the ith diagonal entry given by ω_i. Then, the weighted least squares problem involves finding a function $\boldsymbol{v} \in V$, which solves the minimization problem

$$\min_{\boldsymbol{u} \in V} \sum_{i=1}^m \omega_i \left\| \boldsymbol{u}(\boldsymbol{x}_i) - \boldsymbol{f}_i \right\|_2^2. \tag{2}$$

By expanding the solution $\boldsymbol{v} \in V$ of (2) as $\boldsymbol{v} = \sum_{j=1}^n \boldsymbol{c}_j \beta_j$, the matrix $\boldsymbol{c} = (\boldsymbol{c}_1, \dots, \boldsymbol{c}_n)^\top \in \mathbb{R}^{n \times D}$ is the solution to problem (2) in its equivalent *normal form*,

$$\min_{\boldsymbol{c} \in \mathbb{R}^{n \times D}} \| W^{\frac{1}{2}} B\boldsymbol{c} - W^{\frac{1}{2}} \boldsymbol{f} \|_2^2, \tag{3}$$

where, $\boldsymbol{f} = (\boldsymbol{f}_1, \dots, \boldsymbol{f}_m)^\top \in \mathbb{R}^{m \times D}$,

$$W = \operatorname{diag}\{\omega_1, \dots, \omega_m\} = \begin{pmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_m \end{pmatrix} \in \mathbb{R}^{m \times m}$$

and $W^{\frac{1}{2}}$ denotes the element-wise evaluation of the square root of W. The minimization in (3) entails solving a system of linear equations for each column $\boldsymbol{c}^k = (c_1^k, \dots, c_n^k)$, $k = 1, \dots, D$, of $\boldsymbol{c}$. More precisely, for each $k = 1, \dots, D$, $\boldsymbol{c}^k$ is the solution of the normal equation

$$B^\top W B\boldsymbol{c}^k = B^\top W \boldsymbol{f}^k. \tag{4}$$

3. Weighted least squares via interpolation

In this section, to alleviate the notation, we will assume that $D = 1$. Therefore, we avoid the superscript k introduced in Section 2 and reduce the bold writing according to the dimension. Nevertheless, all the results apply in the more general case of $D \geq 1$. Consider $\Omega \subseteq \mathbb{R}^N$, $\{(\boldsymbol{x}_1, f_1), \dots, (\boldsymbol{x}_m, f_m)\}$ a set of given observations with $\boldsymbol{x}_i \in \Omega$ and $f_i \in \mathbb{R}$ and finally $V = \operatorname{span}\{\Gamma\}$ a vector space of dimension n, with basis $\Gamma = \{\beta_1, \dots, \beta_n\}$ and functions $\beta_j : \Omega \to \mathbb{R}$, for $j = 1, \dots, n$. For any set $K \subset \mathbb{N}$ we denote by $\#(K)$ the cardinality of K and we define $\mathcal{P}_n = \{K \subseteq \{1, \dots, m\} : \#(K) = n\}$, the set of all subsets of $\{1, \dots, m\}$ with cardinality n. For each $K \in \mathcal{P}_n$, there exist $k_i \in \{1, \dots, m\}$ for $i = 1, \dots, n$, such that $K = \{k_1, \dots, k_n\}$. We denote by $v_K \in V$ the interpolant of f_{k_i} at points $\boldsymbol{x}_{k_i}$ for $i = 1, \dots, n$. For such K, we define

$$B_K = \begin{pmatrix} \beta_1(\boldsymbol{x}_{k_1}) & \dots & \beta_n(\boldsymbol{x}_{k_1}) \\ \vdots & \ddots & \vdots \\ \beta_1(\boldsymbol{x}_{k_m}) & \dots & \beta_n(\boldsymbol{x}_{k_m}) \end{pmatrix} \in \mathbb{R}^{n \times n} \tag{5}$$

which is called the *collocation matrix* at the points $\boldsymbol{x}_{k_i}$ with respect to the whole basis Γ of V. For $K \in \mathcal{P}_n$ and positive weights $\omega_1, \dots, \omega_m \in \mathbb{R}$, we write

$$\omega_K = \prod_{i \in K} \omega_i. \tag{6}$$

Following the notation in [9], we define $\lambda_K = \omega_K \left|B_K\right|^2$, where $\left|B_K\right|$ denotes the determinant of B_K. Finally, we define $\mathcal{P}_n^\star \subset \mathcal{P}_n$ as the set of all subsets of $\{1, \dots, m\}$ of cardinality n with $|B_K| \neq 0$. In other words, $\mathcal{P}_n^\star = \{K \subseteq \{1, \dots, m\} : \#(K) = n, \text{ and } |B_K| \neq 0\}$.

Theorem 1. *The weighted least squares approximant $v \in V$ of the set of points $\{(\boldsymbol{x}_i, f_i)\}_{i=1}^m$ is the weighted sum of the interpolants $v_K \in V$ for $K \in \mathcal{P}_n^\star$ which interpolate the points $(\boldsymbol{x}_i, f_i)$ for $i \in K$, i.e.,*

$$v(\boldsymbol{x}) = \frac{\sum_{K \in \mathcal{P}_n^\star} \lambda_K v_K(\boldsymbol{x})}{\sum_{K \in \mathcal{P}_n^\star} \lambda_K}, \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \tag{7}$$

Proof. By hypothesis, $v \in V$ minimizes (2). If $\Gamma = \{\beta_1, \dots, \beta_n\}$ is a basis of V, then we can write $v(\boldsymbol{x}) = \sum_{j=1}^n c_j \beta_j(\boldsymbol{x})$, for some coefficients $c_j \in \mathbb{R}$ for $j = 1, \dots, n$. We set $\boldsymbol{c} = (c_1, \dots, c_n)^\top$ and $\boldsymbol{f} = (f_1, \dots, f_m)^\top$, and write the normal Eq. (4) as $A\boldsymbol{c} = \boldsymbol{b}$ with

$$A = B^\top W B = \begin{pmatrix} \omega_1\beta_1(\boldsymbol{x}_1) & \dots & \omega_m\beta_1(\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots \\ \omega_1\beta_n(\boldsymbol{x}_1) & \dots & \omega_m\beta_n(\boldsymbol{x}_m) \end{pmatrix} \begin{pmatrix} \beta_1(\boldsymbol{x}_1) & \dots & \beta_n(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \beta_1(\boldsymbol{x}_m) & \dots & \beta_n(\boldsymbol{x}_m) \end{pmatrix},$$
$$\boldsymbol{b} = B^\top W \boldsymbol{f} = \left(\textstyle\sum_{i=1}^m \omega_i\beta_1(\boldsymbol{x}_i) f_i, \quad \dots, \quad \sum_{i=1}^m \omega_i\beta_n(\boldsymbol{x}_i) f_i\right)^\top.$$

By the Cauchy–Binet theorem [5, Section 4.6], $|A| = \sum_{K \in \mathcal{P}_n} \omega_K \left|B_K\right|^2$ and by Cramer's rule, we obtain $c_j = \left|A_j\right| \Big/ |A|$, for each $j = 1, \dots, n$, where A_j is obtained from A replacing its jth column with $\boldsymbol{b}$ for $j = 1, \dots, n$, namely, $A_j = B^\top W B_j$ with

$$B_j = \begin{pmatrix} \beta_1(\boldsymbol{x}_1) & \dots & \beta_{j-1}(\boldsymbol{x}_1) & f_1 & \beta_{j+1}(\boldsymbol{x}_1) & \dots & \beta_n(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_1(\boldsymbol{x}_m) & \dots & \beta_{j-1}(\boldsymbol{x}_m) & f_m & \beta_{j+1}(\boldsymbol{x}_m) & \dots & \beta_n(\boldsymbol{x}_m) \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

Again from the Cauchy–Binet theorem, we obtain $\left|A_j\right| = \sum_{K \in \mathcal{P}_n} \omega_K \left|B_K\right| \left|B_{j,K}\right|$ for $j = 1, \dots, n$, where $B_{j,K}$ is obtained from B_K by replacing its jth column with $(f_1, \dots, f_m)^\top$ for $j = 1, \dots, n$. If $K \in \mathcal{P}_n^\star$, the interpolation conditions satisfy $B_K \boldsymbol{c}_K = \boldsymbol{f}_K$, where $\boldsymbol{f}_K = (f_{k_1}, \dots, f_{k_n})^\top$ with $K = \{k_1, \dots, k_n\}$. Therefore, we write the solution to the linear problem as $\boldsymbol{c}_K = (c_{1,K}, \dots, c_{n,K})^\top$ which is unique for $\left|B_K\right| \neq 0$, and so by Cramer's rule we obtain $c_{j,K} = \left|B_{j,K}\right| \Big/ \left|B_K\right|$. The associated interpolant $v_K(\boldsymbol{x})$ for $K \in \mathcal{P}_n^\star$ can be written as $v_K(\boldsymbol{x}) = \sum_{j=1}^n c_{j,K} \beta_j(\boldsymbol{x})$, hence by letting $\lambda_K = \omega_K \left|B_K\right|^2$, it holds

$$\begin{aligned} v(\boldsymbol{x}) &= \sum_{j=1}^n c_j \beta_j(\boldsymbol{x}) = \sum_{j=1}^n \frac{\left|A_j\right| \beta_j(\boldsymbol{x})}{|A|} = \sum_{j=1}^n \frac{\sum_{K \in \mathcal{P}_n} \omega_K \left|B_K\right| \left|B_{j,K}\right| \beta_j(\boldsymbol{x})}{\sum_{K \in \mathcal{P}_n} \omega_K \left|B_K\right|^2} \\ &= \frac{\sum_{K \in \mathcal{P}_n^\star} \omega_K \left|B_K\right|^2 \sum_{j=1}^n c_{j,K} \beta_j(\boldsymbol{x})}{\sum_{K \in \mathcal{P}_n^\star} \omega_K \left|B_K\right|^2} = \frac{\sum_{K \in \mathcal{P}_n^\star} \lambda_K v_K(\boldsymbol{x})}{\sum_{K \in \mathcal{P}_n^\star} \lambda_K}. \end{aligned}$$

□

Remark 1. Note that Theorem 1 is applicable to any finite-dimensional (multivariate) vector space. This encompasses various function spaces, such as the space of polynomials up to a specific degree, spline spaces with fixed degree and order, spline spaces with varying locations of knot lines or any other real function space of finite dimension equipped with a basis. Moreover, note that Eq. (7) in Theorem 1 is very similar to the formulation of multinode Shepard operators [17] in the case of polynomial spaces V.

Remark 2. In general, we have $\#\left(\mathcal{P}_n^\star\right) \leq \#\left(\mathcal{P}_n\right) = \binom{m}{n}$, due to the non-nullity condition on the determinants $|B_K|$, which guarantees the existence of the interpolants. In particular, in the case of splines, the condition $|B_K| \neq 0$ is equivalent to the Schoenberg–Whitney nesting condition [14]. In addition, the value of λ_K depends on the location of the knot lines.

3.1. Consequences of the interpolatory formulation of weighted least squares approximation

For the univariate polynomial case, results on the upper and lower pointwise error bounds of the approximant derivatives up to a certain order as well as on the influence of the weights have been presented in [9, Section 3] and [9, Section 4], respectively. We extend these results to any vector space V of finite dimension n. This generalization extends the derived conclusions of [9] in a broader setting, beyond the polynomial scenario. More precisely, let $r \in \mathbb{N}$ be given and let $\alpha = (\alpha_1, \dots, \alpha_N)$ be any multi-index with $\sum_{j=1}^{N} \alpha_j = r \geq 0$. If the rth order derivative of $v \in V$ exists, it can be expressed as the weighted average of the rth order derivatives of the interpolants, i.e.,

$$\partial^\alpha v(\boldsymbol{x}) = \frac{\sum_{K \in \mathcal{P}_n^\star} \lambda_K \partial^\alpha v_K(\boldsymbol{x})}{\sum_{K \in \mathcal{P}_n^\star} \lambda_K}. \tag{8}$$

In addition, pointwise upper and lower bounds for the value of the rth order derivative of $v(\boldsymbol{x})$ can be obtained from (8), i.e.,

$$\min_{K \in \mathcal{P}_n^\star} \partial^\alpha v_K(\boldsymbol{x}) \leq \partial^\alpha v(\boldsymbol{x}) \leq \max_{K \in \mathcal{P}_n^\star} \partial^\alpha v_K(\boldsymbol{x}).$$

Likewise, the pointwise approximation error shares the same weighted average, given by

$$f(\boldsymbol{x}) - v(\boldsymbol{x}) = \frac{\sum_{K \in \mathcal{P}_n^\star} \lambda_K \left(f(\boldsymbol{x}) - v_K(\boldsymbol{x})\right)}{\sum_{K \in \mathcal{P}_n^\star} \lambda_K}.$$

Furthermore, the following consequences on the influence of the weights ω_i for $i = 1, \dots, m$ and derivative estimations can be inferred.

Corollary 1. *Let $I \subseteq \{1, \dots, m\}$ be of cardinality $\#(I) = r$, with $1 \leq r \leq n$. Define*

$$u_I(\boldsymbol{x}) = \lim_{\substack{\omega_i \to +\infty \\ \forall i \in I}} v(\boldsymbol{x}) \quad \text{and} \quad \mathcal{P}_n^\star \setminus I = \{K \subseteq \{1, \dots, m\} \setminus I : \#(K) = n, \text{ and } |B_K| \neq 0\}.$$

If $r = n$, then $u_I(\boldsymbol{x}) = v_I(\boldsymbol{x})$, represents the interpolant of the data points indexed by I. If $1 \leq r < n$, then

$$u_I(\boldsymbol{x}) = \frac{\sum_{K \in \mathcal{P}_{n-r}^\star \setminus I} \lambda_{I,K} v_{I \cup K}(\boldsymbol{x})}{\sum_{K \in \mathcal{P}_{n-r}^\star \setminus I} \lambda_{I,K}}, \quad \text{with} \quad \lambda_{I,K} = \omega_K \left|B_{I \cup K}\right|^2.$$

Proof. The proof can be obtained by considering $V = \operatorname{span}\{\Gamma\}$, where $\Gamma = \{\beta_1, \dots, \beta_n\}$ with $\beta_j : \Omega \to \mathbb{R}$ for $j = 1, \dots, n$, and substituting the Vandermonde matrices in [9] with the corresponding collocation matrices B in (1) and B_K in (5). □

Another important implication of Theorem 1 is the possibility of rewriting ℓ^p–approximation problems as suitable convex combination of interpolants as outlined in Remark 3 where we exploit the Iterative Reweighted Least Squares (IRLS) method, see e.g., [2, Section 4.5].

Remark 3. Let us consider the problem

$$\min_{u \in V} \sum_{i=1}^{m} \left\|u(\boldsymbol{x}_i) - f_i\right\|_p^p \tag{9}$$

with $1 < p < 2$, where $V = \operatorname{span}\{\Gamma\}$ is a vector space of dimension n, and $\Gamma = \{\beta_1, \dots, \beta_n\}$ its basis. The IRLS method approximates the exact solution of problem (9) through the iterative computation of weighted least squares problems. In particular, the weights are recursively updated for a maximum number of iterations $K_{\max}$ by

$$\omega_i^{k+1} = \left|u^k(\boldsymbol{x}_i) - f_i\right|^{(p-2)/2},$$

with weights ω_i^k and solution u^k of problem (2), for iterations $k = 1, \dots, K_{\max}$. By direct application of (7) in Theorem 1, the outcome of each iteration can be rewritten as a convex combination of interpolants. Note that the interpolants and the determinant of the matrices B_K do not need to be recomputed. In other words, it is enough to update the weights w_K according to (6), to compute the solution of problem (9).

4. Hierarchical spline models

The specification of the basis used to design the continuous model $\boldsymbol{v} : \Omega \to \mathbb{R}^D$ approximating the observations $\{(\boldsymbol{x}_i, \boldsymbol{f}_i)\}_{i=1}^m$, with $\boldsymbol{x}_i \in \Omega \subset \mathbb{R}^N$ and $\boldsymbol{f}_i \in \mathbb{R}^D$ for $i = 1, \dots, m$, plays a fundamental role both for the geometrical and numerical properties of the approximant $\boldsymbol{v}$. Beyond polynomial constructions, splines have been largely adopted both for geometric modelling [22] and isogeometric analysis [33] because of their properties as well as the simplicity of their constructions. In particular, for $N = 1$, let $\Omega = [a, b] \subset \mathbb{R}$, and $\{a \equiv \tau_0 < \tau_1 < \cdots < \tau_{L-2} < \tau_{L-1} \equiv b\}$ a partition of Ω, which defines $\gamma_j = [\tau_j, \tau_{j+1})$ for $j = 0, \dots, L-3$ and $\gamma_{L-2} = [\tau_{L-2}, \tau_{L-1}]$. Moreover, set a certain polynomial degree d and corresponding order $k = d + 1$ and the multiplicities $\{\mu_0, \dots, \mu_{L-1}\}$ associated to each τ_j, with $1 \leq \mu_j \leq k$, for $j = 0, \dots, L-1$. These elements lead to the definition of the so called

knot vector $\boldsymbol{t} = [\tau_0, \tau_1, \dots, \tau_{L-1}]$, whose items τ_j appears according to their multiplicity μ_j, for $j = 0, \dots, L-1$, and finally to the polynomial spline space on Ω, i.e.

$$\left\{ u : \Omega \to \mathbb{R} \quad | \quad u_{|\gamma_j} \in \Pi_k,\ u^{(r)}(\tau_j^-) = u^{(r)}(\tau_j^+) \text{ for } r = 1, \dots, k - \mu_j \text{ and } j = 0, \dots, L-2 \right\},$$

where Π_k indicates the space of polynomials of maximum order k and $u^{(r)}$ indicates the rth order derivative of u. Moreover, univariate splines are non-negative, have local support and form a partition of unity.

When moving to multivariate settings, for $N > 1$, hierarchical splines provide a natural strategy to preserve locality and therefore develop adaptive spline constructions with local refinement capabilities. In the context of data fitting, this allows to suitably design adaptive spline approximation schemes. For $N > 1$, we introduce the polynomial multi-order $\boldsymbol{k}$ and consider a sequence of *nested* linear vector *spline* spaces defined by $V^0 \subset \dots \subset V^\ell \subset V^{\ell+1} \subset \dots \subset V^{L-1}$, where the index ℓ of V^ℓ is called its level. We assume that $V^\ell = \operatorname{span}\left\{\Gamma_{\boldsymbol{k}}^\ell\right\}$ have finite dimension $n_\ell = \dim V^\ell$ where the polynomial spline basis $\Gamma_{\boldsymbol{k}}^\ell = \left\{\beta_1^\ell, \dots, \beta_{n_\ell}^\ell\right\}$ is defined by the polynomial multi-order $\boldsymbol{k}$ and spline basis functions $\beta_j^\ell : \Omega \to \mathbb{R}$. In addition, for each level ℓ, we assume that the boundaries of the supports of all basis functions β_j^ℓ, for $j = 1, \dots, n_\ell$, partition the domain Ω into a certain number of connected *cells* of level ℓ. The collection of all cells among all the levels defines a tessellation $T(\Omega)$ of the domain Ω. In addition, let $\Omega \equiv \Omega^0 \supset \dots \supset \Omega^\ell \supset \Omega^{\ell+1} \supset \Omega^L \equiv \emptyset$ be a sequence of nested open subdomains, where each Ω^ℓ represents a local region of Ω where additional degrees of freedom are required. In particular, we assume that the closure of each subdomain Ω^ℓ coincides with the closure of a collection of cells of level ℓ. The hierarchical spline construction consists of replacing any spline basis function of level ℓ whose support is completely contained in $\Omega^{\ell+1}$ by splines at successively hierarchical levels. More precisely, the hierarchical spline basis is defined as

$$\mathcal{H}_{\boldsymbol{k}} := \left\{ \beta_j^\ell \mid j \in A_{\boldsymbol{k}}^\ell,\ \ell = 0, \dots, L-1 \right\} \tag{10}$$

with the indices

$$A_{\boldsymbol{k}}^\ell := \left\{ j \in \Gamma_{\boldsymbol{k}}^\ell \mid \operatorname{supp}\left(\beta_j^\ell\right) \subseteq \Omega^\ell \wedge \operatorname{supp}\left(\beta_j^\ell\right) \not\subseteq \Omega^{\ell+1} \right\}$$

where $\operatorname{supp}\left(\beta_j^\ell\right)$ denotes the intersection of the support of β_j^ℓ with Ω^0. The corresponding hierarchical space is defined as $\operatorname{span}\left\{\mathcal{H}_{\boldsymbol{k}}\right\}$.

Several families of spline basis functions and their corresponding function spaces admit a hierarchical extension. That is the case of B-splines [14], NURBS [47], as well as box splines [15], among others. Examples of their hierarchical formulations can be found in [24,35,37]. In the following, we provide examples for Theorem 1 and Corollary 1 in the special case of B-spline and hierarchical box-spline models, respectively.

4.1. Verification of Theorem 1 for spline models

In this section, we present the numerical verification of Theorem 1 for two different vector spaces V: the space of polynomials and the space of univariate splines with a specific degree and regularity. We consider a set of $m = 7$ observations in the form (x_i, f_i), for $i = 1, \dots, m$, given by $\{(-4.5, -2), (-3.5, 0), (-2.2, -1), (-1.2, 2.8), (0.8, 2.9), (2.2, 0.5), (4.0, -2)\}$. Furthermore, we assign weights ω_i for $i = 1, \dots, m$ uniformly distributed on $(0, 1)$. In the polynomial case, we choose the polynomial degree $d = 2$, which implies $\#\left(\mathcal{P}_{d+1}\right) = \binom{m}{d+1} = \binom{7}{3} = 35$. Thus, we have a total of 35 interpolation problems to be solved. Moving on to polynomial spline spaces, in addition to the degree, we consider the spline order $k = d + 1 = 3$ and set the knot vector $\boldsymbol{t} = [-5, -5, -5, -5/3, 5/3, 5, 5, 5]$ of length 8, implying that the spline space has dimension $n = 8 - 3 = 5 \le m$. Since $n \le m$, there are at most $\#\left(\mathcal{P}_n\right) = \binom{m}{n} = \binom{7}{5} = 21$ interpolation problems required to fully reconstruct the spline least squares approximant. In the case of spline spaces, there is no guarantee that each point subsequence satisfies the Schoenberg–Whitney nesting conditions. To illustrate this, consider the data set identified by $K = \{1, 2, 3, 4, 5\} \subset \{1, \dots, m\}$, which does not satisfy these conditions. This is due to the absence of data points within the support of the last basis function, specifically $x_i \notin [5/3, 5]$ for $i = 1, \dots, 5$. Consequently, the interpolant v_K does not exist, and no interpolation problem needs to be solved for this particular subset. To fully reconstruct the final global least squares approximation in this example, we need to compute 20 interpolation problems instead of the original 21. Fig. 1 illustrates the interpolants and the global least squares approximation involved in (7) for the polynomial and spline spaces introduced for this numerical example. Note that we choose to show an example related to the univariate polynomial and spline space, but our results hold in general for any (multivariate) vector space of finite dimension.

4.2. Weighted least squares with hierarchical box splines

In this section, we perform a numerical investigation of the role of the weights addressed in Corollary 1 in Section 3.1 for a weighted least squares problem with hierarchical box splines in the bivariate case. We consider a point cloud of 64×64, i.e. $m = 4096$, uniformly gridded data, obtained by sampling the function

$$f(\boldsymbol{x}) = f(x, y) = \frac{2}{3\exp\left(\sqrt{(10x-3)^2 + (10y-3)^2}\right)} + \frac{2}{3\exp\left(\sqrt{(10x+3)^2 + (10y+3)^2}\right)} + \frac{2}{3\exp\left(\sqrt{(10x)^2 + (10y)^2}\right)} \tag{11}$$

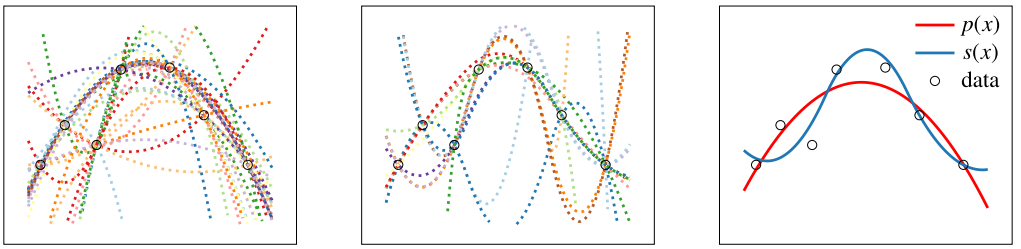


Fig. 1. Numerical verification of (7), Theorem 1: $v_K(\boldsymbol{x})$ interpolants for polynomials (left) and splines (centre) and the final weighted least squares approximations as sum of interpolants (right) for polynomials (red) and splines (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

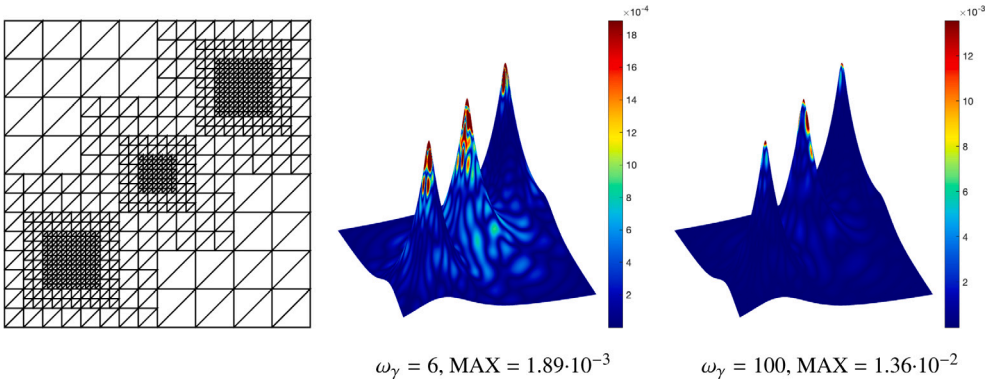


Fig. 2. Hierarchical box spline mesh (left); weighted least squares surfaces for $\omega_\gamma = 6$ (centre) and 100 (right).

for $\boldsymbol{x} = (x_1, x_2) \in [-1, 1]^2$ and we associate to each item of the point cloud a weight $\omega_i > 0$ for $i = 1, \dots, m$. For different weight values, we compute the weighted least squares approximation $v(\boldsymbol{x})$ of the input data in terms of 521 C^2 quartic box splines, in their hierarchical extension, see again [35]. In order to investigate the role of the weights in the approximation process, we execute the following two steps.

(a) Firstly, we perform a standard least squares approximation and individuate the data sites whose approximation error is above a certain threshold ϵ, thus define $K := \{i \in \{1, \dots, m\} : \left|v(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\right| > \epsilon\}$.
(b) Subsequently, for $i \in \{1, \dots, m\} \setminus K$, we set the corresponding weights values $\omega_i = \omega_0$ and similarly for $i \in K$ we set $\omega_i = \omega_\gamma$.

Therefore, we analyse the accuracy behaviour of the resulting approximant $v(\boldsymbol{x})$ for different choices of $(\omega_0, \omega_\gamma)$ in terms of maximum approximation error (MAX). Setting $\epsilon = 5\text{e}{-4}$, then $\#(K) = 124$ and for $\omega_0 = \omega_\gamma = 1$, the resulting ordinary least squares approximation is characterized by a MAX error of $2.15 \cdot 10^{-3}$. We then keep $\omega_0 = 1$ and vary $\omega_\gamma = 2, 3, 4, \dots, 10, 50, 100$. We note that increasing the value of the weights ω_γ may help the final accuracy of the approximant. In particular, we obtain MAX = $2.15 \cdot 10^{-3}$, $2.06 \cdot 10^{-3}$, $2.00 \cdot 10^{-3}$ for $\omega_\gamma = 2, 3, 4$ and MAX decreases further when increasing ω_γ until achieving its minimum, with MAX = $1.89 \cdot 10^{-3}$, for $\omega_\gamma = 6$. However, if the weight values ω_γ are too large, the final approximant deteriorates. For instance, for $\omega_\gamma = 7$, MAX = $1.91 \cdot 10^{-3}$, for $\omega_\gamma = 10$, MAX = $2.63 \cdot 10^{-3}$ and for $\omega_\gamma = 100$, MAX = $1.36 \cdot 10^{-2}$. These results can also be observed in Fig. 2, which depicts the hierarchical box spline mesh (left) together with two weighted least squares approximation resulting from two different choices of weights, namely $\omega_\gamma = 6$ (centre) and $\omega_\gamma = 100$ (right). The pointwise error colour map, ranging from the minimum to the maximum approximation error, is plotted for each surface separately. The corresponding colour bars are also reported.

As shown in this example, the choice of weights can lead to either better or worse spline fitting results in terms of the MAX error compared to ordinary least squares, where all the weights are set to ones. To the best of our knowledge, only a few deterministic methods address the use of weights associated with data points in the spline fitting literature. However, the topic is widely discussed in the statistics community, going back to locally weighted scatterplot smoothing (LOWESS) [11] for smoothing scatterplots by robust locally weighted regression [10] and multivariate adaptive regression splines (MARS) [25], with a more recent method proposed in [6] and the references therein, for example. In this work, we propose a deterministic strategy to suitably take advantage of using weights associated with data points within the spline fitting process which also includes adaptive spline fitting constructions.

5. Reweighted least squares spline fitting

In this section, we propose a strategy to take effective advantage of the weights associated with each input data in the context of fitting problems. This approach is inspired by the notion of landmarks used in shape analysis [3], but we introduce a more general

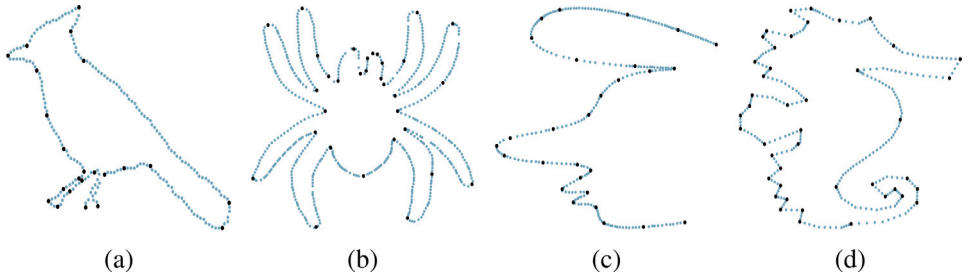


Fig. 3. Point cloud (blue dots) and set of type I markers (black dots). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

concept of *markers* and use them within the framework of curve and surface fitting. More precisely, landmarks can be understood as a set of labelled points which represent some physically identifiable parts of an object, as well as important features of the input data, which need to be encoded and reproduced in the final continuous approximation model. Fig. 3 shows four point clouds (in blue) and the chosen landmarks (in black), which represent the features desirable to be preserved by the fitted curve. However, depending on the acquisition process, data can be affected by noise and outliers whilst the final approximation models should avoid the reproduction of corrupted data. For the given set of observations $\{\boldsymbol{x}_i, \boldsymbol{f}_i\}_{i=1}^m$, we generalize the concept of landmarks to *markers* of two types. More precisely, we define the index set $K_I \subseteq \{1, \dots, m\}$ as *markers of type I* if the associated points $\{\boldsymbol{x}_i, \boldsymbol{f}_i\}$ for $i \in K_I$ represent data features to be preserved, while we define the index set $K_{II} \subseteq \{1, \dots, m\}$ as *markers of type II* if the index indicates noisy data or outliers which should not be reproduced. In particular, we have $K_I \cap K_{II} = \emptyset$ and $K_I \cup K_{II} \subseteq \{1, \dots, m\}$. Note that the choice of type I and type II markers and their identification depends considerably on the problem at hand and it is still an open research topic, see e.g., for automatic features selection [30,55] or for outliers recognition [20,21,44]. The markers identification is out of the scope of the present article and we assume the markers (of both types) to be known a priori. Nevertheless, for synthetic data, we devise an error-driven detection of markers second to point cloud pre-processing. Finally, we provide fitting schemes which can address simultaneously both types of markers, by leveraging the weight values associated with them depending on the approximation error, also within an adaptive approximation framework.

The fitting problem with markers consist of finding an element $\boldsymbol{v} \in V$ which approximate the points $\{\boldsymbol{x}_i, \boldsymbol{f}_i\}_{i=1}^m$, i.e. $\boldsymbol{v}(\boldsymbol{x}_i) \approx \boldsymbol{f}_i$, with $\|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2 < \mathrm{tol}_I$ for $i \in K_I$ and $\|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2 < \mathrm{tol}_{II}$ for $i \notin K_{II}$. Note that the usual formulation of a fitting problem can be interpreted as a special instance of the present one. In particular, it is equivalent to setting $K_I = \{1, \dots, m\}$, $K_{II} = \emptyset$ and choosing $\mathrm{tol}_I = \mathrm{tol}_{II} = \epsilon$. Specifically, the reweighted least squares algorithm with markers is described in Algorithm 1 and it consists of the following steps. For an initial choice of the weight values,

1. Solve the weighted least squares approximation problem (2).
2. Update the point-wise error $e_i = \|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2$ for each $i = 1, \dots, m$.
3. Check whether the current fitting $\boldsymbol{v} \in V$ meets the requirements prescribed by the markers K_I and K_{II} and the respective error tolerances. In particular, the fitting process is completed if $e_i < \mathrm{tol}_I$ for $i \in K_I$ and $e_i < \mathrm{tol}_{II}$ for $i \notin K_{II}$.
4. If the accuracy requirements are satisfied, return the approximation computed in Step 1, otherwise update the weight values corresponding to the markers. In particular, $\omega_i = \omega_i \cdot \alpha$ with $\alpha > 1$ for each $i \in K_I$ and $\alpha < 1$ for each $i \in K_{II}$.
5. Start again from Step 1 with the new weight values.

Note that the update choice of the weights in Step 4 increases the values of the weights related to markers of type I, whereas it decreases them for markers of type II. In particular, we suggest an error-driven update of the weights with $\alpha = (1 + e_i)$ for $i \in K_I$ and $\alpha = 1/(1 + e_i)$ for $i \in K_{II}$. Finally, it is worth highlighting that the IRLS procedure [2,46] falls within this general fitting scheme for the special choice of $\alpha = 1/\max\{\delta, e_i\}$ for each $i \in K_{II}$, with $\delta > 0$ necessary for the stability of the IRLS method.

The reweighted fitting scheme from Algorithm 1 is presented in its more general formulation, namely, the approximant is sought in any *fixed* finite-dimensional vector space V. However, in the incoming numerical experiments, we apply it to spline models as the ones defined in Section 4.

As far as spline constructions are concerned, the definition of a fixed spline space $V = \mathrm{span}\{\beta_1, \dots, \beta_n\}$ relies on the choice of the polynomial degree and order as well as a suitable tessellation $T(\Omega)$ of the domain Ω. For spline fitting problems, in addition to the definition of the spline space V also the parameterization of the data needs to be addressed, namely to associate at each $\boldsymbol{f}_i \in \mathbb{R}^D$ a suitable parametric value $\boldsymbol{x}_i \in \Omega$. Both *knot placement* to define $T(\Omega)$ and *parameterization* to define the sites $\boldsymbol{x}_i \in \Omega$ for $i = 1, \dots, m$, play fundamental roles in the accuracy of the final spline fitting model, they are crucial open research topic and several methods have been provided to properly address them, see e.g. [19,23,32,39,51,53,54] and the references therein. Their resolution is out of the scope of the present paper, thus we assume both the parameterization and the spline space to be constructed with established techniques. Nevertheless, special attention needs to be dedicated to adaptive spline construction, where the space V is not fixed but enlarged and updated iteratively.

Algorithm 1: General formulation of the reweighted least squares fitting.

Data: Point cloud $\{\boldsymbol{x}_i, \boldsymbol{f}_i\}_{i=1}^m$, the set of markers $K_I, K_{II} \subseteq \{1, \dots, m\}$, the tolerances tol_I, tol_{II}, a fixed vector space $V = \mathrm{span}\{\beta_1, \dots, \beta_n\}$ and a maximum number of iterations $\mathrm{M}_{\max}$;

Result: $\boldsymbol{v} \in V$ the reweighted least squares approximant.

Initialize the weights $\omega_i = 1$ and the point-wise errors $e_i = 1$ for each $i = 1, \dots, m$ and loop = 0

while $\max_{i \in K_I} e_i > \mathrm{tol}_I$ *and* $\max_{i \in I \setminus K_{II}} e_i > \mathrm{tol}_{II}$ *and* loop $< \mathrm{M}_{\max}$ **do**

  Solve the weighted least squares problem (2);

  compute the errors $e_i = \|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2$;

  update the weights associated to the landmarks K_I and K_{II}, as $w_i = w_i \cdot \alpha(e_i)$;

  set loop = loop+1;

end

return $\boldsymbol{v} \in V$ *which solves the weighted least squares problem.*

5.1. Reweighted least squares adaptive spline fitting

In this section, we show how to suitably combine the update of the weights with adaptive spline approximation schemes. We revisit the adaptive least squares fitting scheme proposed in [26] by suitably assigning weight values to the data observation within the adaptive routine. The main idea which drives an adaptive fitting algorithm consists of adding iteratively degrees of freedom in regions of the domain Ω where the approximation error is too high. Similarly to Algorithm 1, if the points with a too high error belong to K_I, then their weights will be augmented, otherwise if they belong to K_{II} their weights will be diminished. In addition, at each iteration of the adaptive loop, not only the updates of the weight values take place, but also of the sets K_I and K_{II}. This strategy is effective since, thanks to the adaptive refinement, in some regions of the domain the accuracy requirements $\mathrm{tol}_I, \mathrm{tol}_{II}$ are already locally achieved and it is useless or even harmful to keep on modifying the weight values. More precisely, once an initial parameterization and tessellation Ω are chosen, i.e. for a fixed hierarchical spline space V, and for an initial choice of the weights we perform the following steps.

1. Compute the (penalized) weighted least squares problem

$$\boldsymbol{v}(x) = \underset{\boldsymbol{u} \in V}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{m} \omega_i \left\| \boldsymbol{u}\left(\boldsymbol{x}_i\right) - \boldsymbol{f}_i \right\|_2^2 + \lambda J\left(\boldsymbol{u}\right), \tag{12}$$

   where the penalization term J is the thin-plate energy functional, whose influence is controlled by $\lambda \geq 0$, i.e. for $\boldsymbol{x} = (s, t) \in \Omega$,

$$J\left(\boldsymbol{u}\right) = \int_{\Omega} \left\| \frac{\partial^2 \boldsymbol{u}}{\partial s \partial s} \right\|_2^2 + 2 \left\| \frac{\partial^2 \boldsymbol{u}}{\partial s \partial t} \right\|_2^2 + \left\| \frac{\partial^2 \boldsymbol{u}}{\partial t \partial t} \right\|_2^2 \mathrm{d}s\mathrm{d}t. \tag{13}$$

2. Evaluate the error indicator as the point-wise error distance $\|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2$ for each $i = 1, \dots, m$. By individuating the sites $\boldsymbol{x}_i \in \Omega$ where $\|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2 \geq \epsilon$, the error indicator identifies the region of the domain Ω where potentially additional degrees of freedom are needed to meet the prescribed accuracy ϵ.
3. If $\|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2 < \epsilon$ for each $i = 1, \dots, m$ return the approximation computed in Step 1, otherwise update the weights valued and the markers K_I and K_{II}. In particular, for each $i \in K_I$ if $e_i < \mathrm{tol}_I$, then the ith data satisfies the accuracy requirements and it should not belong to the markers of type I any more, namely $K_I = K_I \setminus \{i\}$. On the contrary, if $e_i > \mathrm{tol}_I$, then its corresponding weight needs to be enlarged, i.e. $\omega_i = \alpha \cdot \omega_i$, with $\alpha > 1$. Similar considerations hold for $i \in K_{II}$, with inverted inequalities and $\alpha < 1$.
4. Mark the cells of the current hierarchical spline space which contain the parameters $\boldsymbol{x}_i$ identified by the error indicator.
5. Refine the marked cells by suitably splitting them to effectively enlarge the hierarchical spline space.
6. Update the data parameterization by suitably moving the data sited $\boldsymbol{x}_i$ for each $i = 1, \dots, m$.
7. Start again from Step 1 with the new weight values.

Note that in Step 1 we introduced the penalization term (13), usually addressed as *thin-plate energy*, whose influence is ruled by a weight $\lambda \geq 0$. The introduction of such a functional is a common practice in spline geometric modelling, in particular when reconstructing a spline geometry from a set of unorganized data. More precisely, a regularization term is commonly introduced to smoothen the solution and therefore avoid the presence of spurious oscillations and artefacts, which may affect the final geometric model. According to [36], we set $\lambda \geq 0$ to be a constant small ($\sim 10^{-6}$) value and we keep it fixed during the entire fitting process. Non-constant regularization weight functions for data fitting via least-squares tensor-product splines have been recently proposed in [40,43], see also the references therein, e.g., [12,28,50].

For more details on Steps 4 to 6, about marking refinement and parameterization routines, we refer the reader to [26]. Finally, the pseudo-code of the reweighted least squares adaptive spline algorithm is reported in Algorithm 2.

Algorithm 2: Reweighted adaptive least squares spline fitting.

Data: Point cloud $\{\boldsymbol{x}_i, \boldsymbol{f}_i\}_{i=1}^m$, the set of markers $K_I, K_{II} \subseteq \{1, \dots, m\}$, the tolerances tol_I, tol_{II} and $\epsilon > 0$, the penalization weight $\lambda \geq 0$, a tensor product spline space $V = \mathrm{span}\{\beta_1, \dots, \beta_n\}$ and a maximum number hierarchical level L;

Result: $\boldsymbol{v} \in \mathcal{H}$ the reweighted adaptive least squares approximant.

Initialize the weights $\omega_i = 1$ and the point-wise errors $e_i = 1$ for each $i = 1, \dots, m$ and loop = 0
while $\max_i e_i > \epsilon$ *and* loop < L **do**
  Solve the penalized weighted least squares problem (12)
  compute the pointwise errors $e_i = \|\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{f}_i\|_2$;
  for each $i \in K_I$, if $e_i > \mathrm{tol}_I$, update $\omega_i = \omega_i \cdot \alpha(e_i)$, else $K_I = K_I \setminus \{i\}$;
  for each $i \in K_{II}$, if $e_i < \mathrm{tol}_{II}$, update $\omega_i = \omega_i \cdot \alpha(e_i)$, else $K_{II} = K_{II} \setminus \{i\}$;
  mark the domain elements where $e_i > \epsilon$;
  refine by dyadic split of the marked cells;
  update the data parameterization;
  set loop = loop+1
end
return $\boldsymbol{v} \in \mathcal{H}$ *which solves the weighted adaptive spline least squares problem.*

6. Numerical experiments

In this section we present a selection of numerical experiments to show the performance of the proposed fitting schemes. More precisely, in Section 6.1, we show the effectiveness of the proposed reweighted least squares spline fitting method to recover the sharp features of the four different point sequences with type I markers, depicted in Fig. 3, and confront it with ordinary least squares spline fitting. Moreover, in Section 6.2, for the task of curve fitting, we compare the proposed method also with smoothing splines approximations. In Section 6.3, we then extend the proposed method to adaptive spline spaces for surface reconstruction and suggests an automatic recognition of sharp features and type I markers. Finally, in Section 6.4, we show the performance of the adaptive algorithm when only type II markers are considered.

The univariate examples have been implemented in an 8-core laptop (Apple M2) with 8 GB RAM using `MATLAB R2023b`, specifically by employing the `Curve Fitting Toolbox` [34]. The adaptive surface approximations have been implemented within the open source `C++` Geometry + Simulation (`G+Smo`) library [42], by suitably extending the `gsHFitting` class. Both libraries provide an efficient and robust implementation of least squares problems by exploiting suitable solvers for the linear systems at hand, which avoid the direct solution of a linear system of normal equations.

6.1. Reweighted spline curve fitting with type I markers and comparison with ordinary least squares

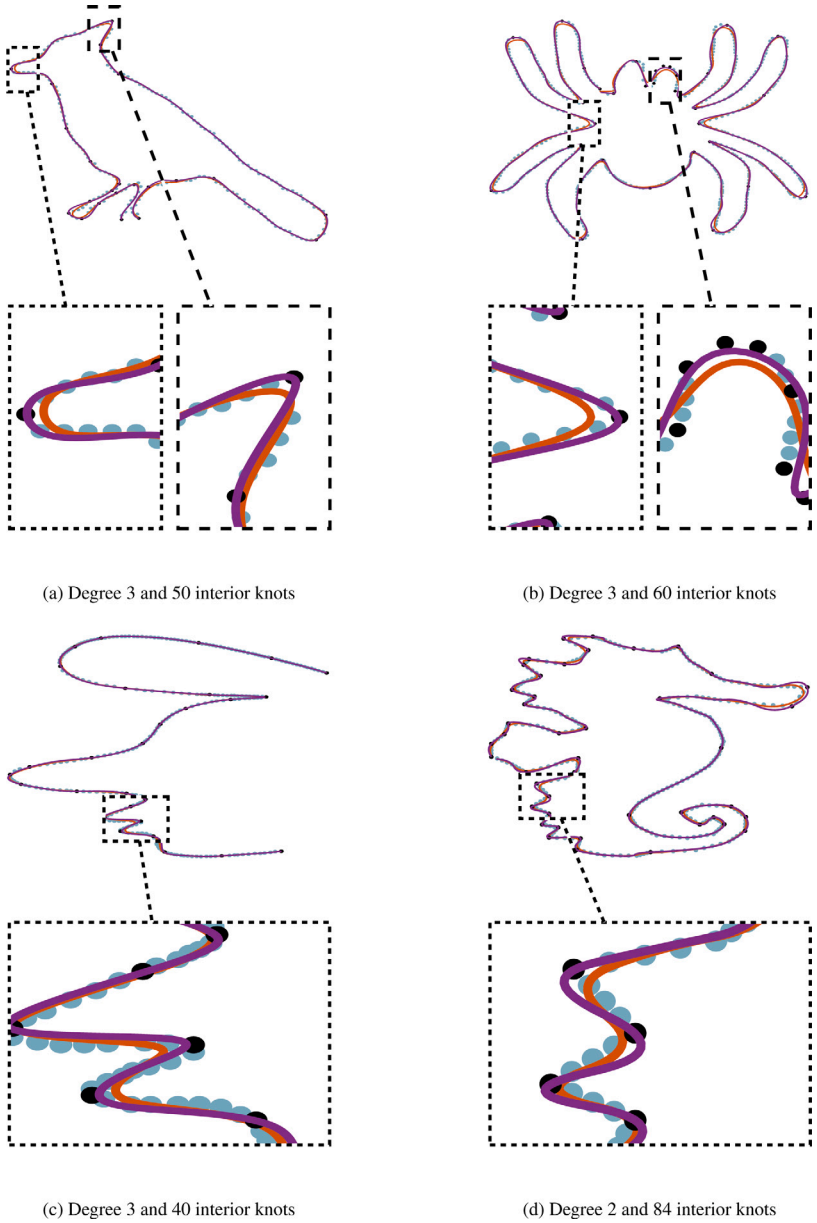
To show the performance of Algorithm 1, we consider the points clouds and markers of type I depicted in Fig. 3. The point clouds are similar to the ones presented in the following works: for Fig. 3(a) [45], (b) [38], (c) and (d) [41]. Our main goal is to improve the reconstruction while keeping the spline space intact. Therefore, each considered point sequence is parameterized accordingly to the *uniform parameterization* and we compute the approximations of each dataset for the same spline space, i.e. with the same polynomial degree the same amount of interior nodes of the *uniform* knot vector. We set the tolerance of the proposed method as $\mathrm{tol}_I = 10^{-3}$ and compare the solution of the reweighted least squares problem (rWLS) with the solution of the ordinary least squares problem (LS) i.e., all the weights are equal to one. The solutions are depicted in Fig. 4(a), (b), (c), and (d), for the input data of Fig. 3(a), (b), (c), and (d) respectively. In particular, we can clearly see that the approximation using the reweighted least squares method (depicted in purple in Fig. 4) shows a better approximation power with respect to the solution of the ordinary least squares problem (shown in red). In particular, one can see when zooming in that the sharp features (illustrated in black) are preserved during the curve fitting. Regarding the approximation error for each experiment, we report that, if considering *all* the observation data, the rooted mean squares error (RMSE) and maximum error (MAX) are comparable with the ordinary least squares. However, if we compare only how well we can preserve the sharp features, the rWLS method considerably outperforms the ordinary LS. More precisely, in Table 1, we report the RMSE and MAX errors measured only for the set of type I marks, respectively for rWLS and LS.

6.2. Reweighted spline curve fitting with type I markers and comparison with smoothing splines

In addition to ordinary least squares, another widely used method for curve fitting is smoothing splines [28,29]. This method is derived as the solution to the penalized least squares problem (12), in terms of cubic splines with knots corresponding to the x-coordinates of the observations.

Therefore, we considered three additional point clouds sampled from the following functions:

$$f_1(x) = \left| \frac{9\sin(3\pi x)}{\tanh(-1.5x+1)+1} \right|, \quad f_2(x) = \frac{1}{0.02\sqrt{\pi}} \exp\left\{ -\left(\frac{x-0.5}{0.02} \right)^2 \right\}, \quad f_3(x) = \tanh\left(\frac{\cos(2\pi x)}{0.05} \right)$$



(a) Degree 3 and 50 interior knots

(b) Degree 3 and 60 interior knots

(c) Degree 3 and 40 interior knots

(d) Degree 2 and 84 interior knots

Fig. 4. Curve fitting experiment described in Section 6.1 using ordinary LS (red) and rWLS (purple) for data with markers of type I. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and create 3 point clouds with $62, 88, 71$ points, respectively. Specifically, we consider an irregular distribution of the abscissas to obtain more observations around the functions' sharp features.

We compute the smoothing spline fitting using the `fit` function in MATLAB and use the default "interesting range" value of the smoothing weight λ, which depends on the abscissa distribution of each experiment. Moreover, we set the weights of (12) to 1. To ensure a fair comparison, for the rWLS algorithm, we fixed the spline degree to 3 and selected the knot vectors following the "averaging" technique described in [47, Eq. (9.8)], known in the literature also as NKTP, with 37, 47, and 47 interior knots, respectively. This technique has been chosen since it also considers the values of the abscissas as in the case of smoothing splines.

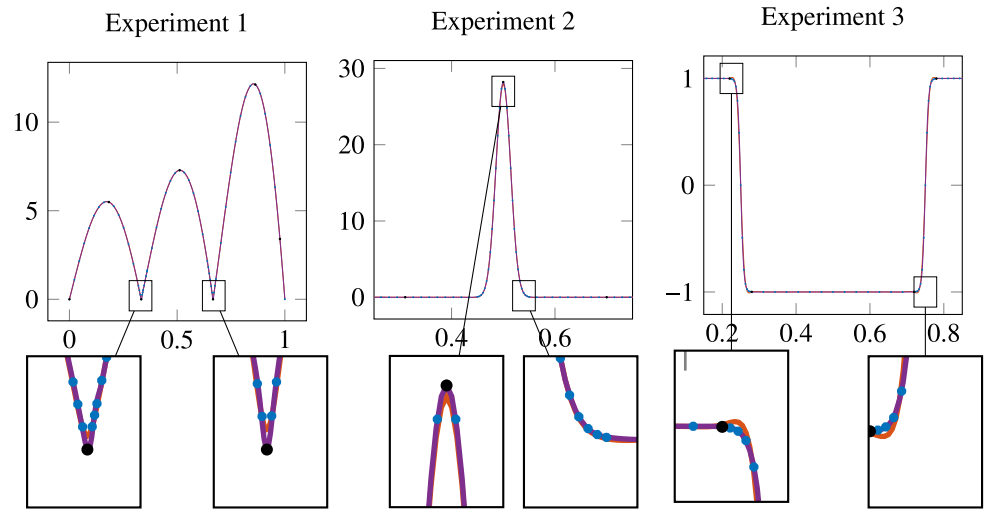
The solutions obtained using rWLS from Algorithm 1 are depicted in Fig. 5. Additionally, we report the rooted mean squared error (RMSE) and maximum error (MAX) measured only for the set of type I marks for both rWLS and smoothing splines in Table 2. To summarize, in the presence of sharp features, better results can be obtained using the proposed rWLS from Algorithm 1 with fewer degrees of freedom compared to the smoothing spline technique.

Table 1
RMSE and MAX errors for type I markers of LS and rWLS solutions of the experiment described in Section 6.1. For rWLS also the number of iterations (Iterations) is reported.

	Method	Iterations	Time	RMSE	MAX
(a)	LS	–	–	6.27×10^{-3}	1.30×10^{-2}
	rWLS	1442	1.347 s	8.31×10^{-4}	8.95×10^{-4}
(b)	LS	–	–	2.25×10^{-3}	3.94×10^{-3}
	rWLS	1250	1.786 s	6.63×10^{-4}	1.78×10^{-3}
(c)	LS	–	–	4.63×10^{-3}	1.22×10^{-2}
	rWLS	30	0.034 s	1.47×10^{-3}	3.54×10^{-3}
(d)	LS	–	–	5.94×10^{-3}	1.25×10^{-2}
	rWLS	81	0.137 s	3.77×10^{-4}	8.81×10^{-4}

Table 2
RMSE and MAX errors for type I markers of smoothing spline and rWLS solutions of the experiment described in Section 6.2. For rWLS also the number of iterations (Iterations) is reported.

	Method	Iterations	Time	RMSE	MAX
(1)	Smoothing spline	–	0.202 s	1.02×10^{-1}	2.21×10^{-1}
	rWLS	10	0.086 s	9.13×10^{-6}	1.86×10^{-5}
(2)	Smoothing spline	–	0.188 s	1.36×10^{-1}	3.04×10^{-1}
	rWLS	7	0.027 s	1.43×10^{-5}	3.18×10^{-5}
(3)	Smoothing spline	–	0.143 s	2.96×10^{-3}	3.86×10^{-3}
	rWLS	4	0.011 s	1.30×10^{-5}	2.58×10^{-5}



(1) Approximation of $f_1(x)$ with 37 interior knots

(2) Approximation of $f_2(x)$ with 47 interior knots

(3) Approximation of $f_3(x)$ with 47 interior knots

Fig. 5. Curve fitting experiment described in Section 6.2 using smoothing spline (red) and rWLS (purple) for data with markers of type I (black dots). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.3. Reweighted adaptive spline fitting for type I markers

In this numerical experiment we apply Algorithm 2, combined with truncated hierarchical B-splines (THB-splines) as in [26], to a point cloud obtained by sampling 100×100 gridded data from function (11) used in Section 4.2. This function is characterized by sharp features which we want the final model to capture, i.e. we would deal with type I markers only, hence $K_{II} = \emptyset$. In particular to initialize the set K_I we perform an ordinary least squares fitting with tensor-product B-splines, of bi-degree 3 and a 15×15 mesh, and individuate the data sites whose approximation error is above a certain threshold ϵ, namely $K_I := \{i \in \{1, \dots, m\} : \|v(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\|_2 > \epsilon\}$. Given the data points, the markers K_I, the tolerances and the initial tensor-product B-spline space, we perform Algorithm 2 with $\mathrm{tol}_I = 10\epsilon$ augmenting the weights value by 25% at each iteration of the adaptive loop. We then compare the output of the reweighted adaptive least squares fitting method (rWLS) with the result of the ordinary adaptive least squares fitting (LS), namely by keeping all the weights equal to one all along the iterative procedure. More precisely, we compare the maximum error (MAX) of the final approximations with respect to the number of degrees of freedom (DOFs), choosing

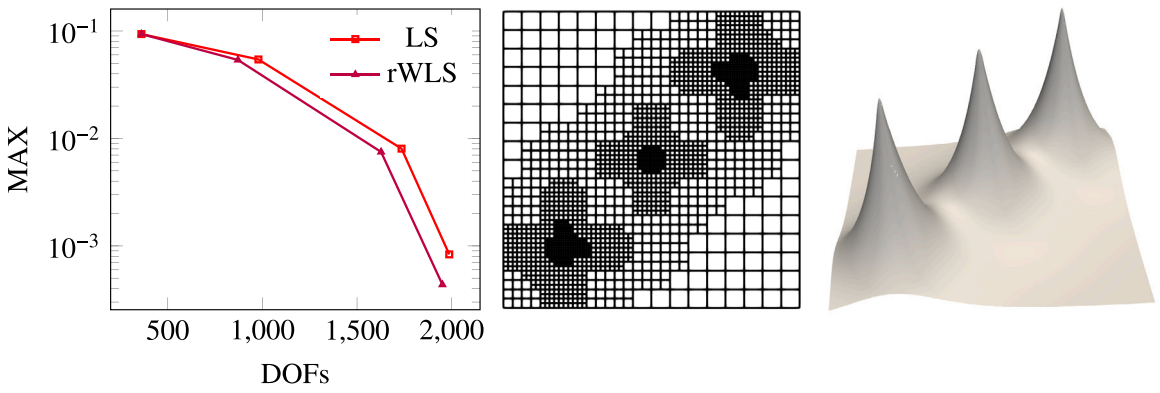


Fig. 6. MAX error w.r.t. DOFs for the ordinary LS and rWLS fitting scheme THB-splines (left). Hierarchical mesh (centre) and geometry (right) are obtained in output from the rWLS method.

ϵ to achieve a comparable number of DOFs. The comparison is reported in Fig. 6 (left), together with the final hierarchical mesh (centre) and THB-spline approximation (right) obtained with the rWLS method. More precisely, LS and rWLS methods have the same starting point and as long as the adaptive refinement proceeds, we can notice that rWLS can register a smaller MAX error with fewer DOFs in comparison to LS.

6.4. Reweighted adaptive spline fitting for type II markers

In this final numerical experiment, we perform the reweighted adaptive spline least squares fitting scheme to approximate a *scattered* point cloud of 9000 points representing a Nefertiti bust, illustrated in Fig. 7 (top-left). In particular, these data require a parameterization and some of them result in being corrupted. To address the first problem, we employ the data-driven parameterization method developed in [27]. To tackle the presence of corrupted data, we exploit the use of the weights associated with the data. More precisely, together with the points, we are given a set of markers of the second type K_{II}, highlighted in black in Fig. 7 (bottom-left) which we use to perform the reweighted adaptive scheme. Moreover, due to the complexity of the problem, we decided not to update the markers K_{II} but to keep them fixed along the iterative scheme, by setting tol_{II} sufficiently small. We then compare the results obtained by the proposed reweighted adaptive least squares scheme (rWLS) with the ordinary adaptive least squares scheme (LS), namely keeping all the weights set to one. Such a comparison is illustrated in Fig. 7, where we show on the top the hierarchical approximation obtained by LS and on the bottom the one obtained by rWLS. In particular, the region corresponding to the corrupted data is better approximated by the rWLS method, which results free of self-intersections and it is smoother than the one obtained with LS. Finally, the MAX error registered by LS is $3.28{\cdot}10^{-2}$, whereas the MAX of rWLS is $1.57{\cdot}10^{-2}$.

7. Conclusions

We provided new insights into the reweighted least squares method for the approximation of a given set of observations. In particular, our theoretical study revealed that any reweighted least squares model can be interpreted as a convex combination of suitable interpolation models. Furthermore, we exploited the weights for spline fitting problems, such as adaptive spline constructions. Thereby, we provided a strategy to automatically update the weights within the fitting scheme, either to emphasize data marked as sharp features or to smoothen data marked as corrupted. In the present work, we assumed that the given data have been previously processed, hence their features or the presence of noise have already been identified. This information was exploited in the fitting schemes via an iterative update of the weights. In future research, it is of interest to automatically classify the data marker used in the spline approximation scheme in order to drive the model design process. Another interesting future research direction is the application of the proposed method to a non-geometric modelling framework, e.g., the approximation of time-series data. In addition, in this context, it would also be interesting to investigate how the choice of the smoothing parameter affects the goodness of the fit from a statistical point of view.

CRediT authorship contribution statement

Carlotta Giannelli: Conceptualization, Formal analysis, Investigation, Methodology, Resources, Writing – original draft, Writing – review & editing. **Sofia Imperatore:** Conceptualization, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Lisa Maria Kreusser:** Conceptualization, Formal analysis, Methodology, Resources, Writing – original draft, Writing – review & editing. **Estefanía Loayza-Romero:** Conceptualization, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Fatemeh Mohammadi:** Conceptualization, Resources. **Nelly Villamizar:** Writing – original draft, Writing – review & editing.

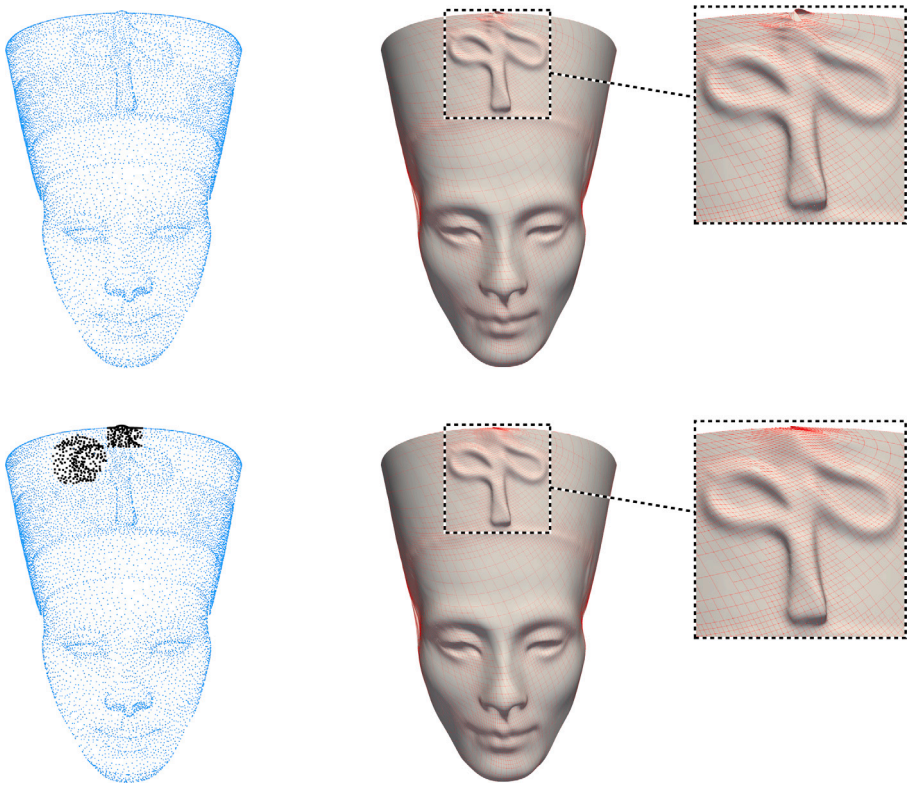


Fig. 7. Adaptive surface fitting experiment described in Section 6.4 using adaptive LS (top) and rWLS (bottom) for data with markers of type II. The data point clouds and the marked points are also shown.

Acknowledgements

The authors would like to acknowledge the support provided by the 4th WiSh: Women in Shape Analysis Research Workshop. This collaboration began during the workshop, and we are deeply grateful for the opportunity to work with fellow researchers in the field. CG and SI are members of the INdAM group GNCS, whose support is gratefully acknowledged. LMK acknowledges support from Magdalene College, Cambridge (Nevile Research Fellowship). ELR work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044 –390685587, Mathematics Münster: Dynamics–Geometry–Structure. FM was partially supported by the FWO grants (G0F5921N, G023721N), the KU Leuven iBOF/23/064 grant, and the UiT Aurora MASCOT project. NV was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) New Investigator Award EP/V012835/1.

References

[1] A. Beck, On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes, SIAM J. Optim. 25 (1) (2015) 185–209.
[2] Å. Björck, Numerical Methods for Least Squares Problems, Society for Industrial and Applied Mathematics, 1996.
[3] F.L. Bookstein, The Measurement of Biological Shape and Shape Change, Springer, Berlin, Heidelberg, 1978.
[4] L. Bos, A. Sommariva, M. Vianello, Least-squares polynomial approximation on weakly admissible meshes: Disk and triangle, J. Comput. Appl. Math. 235 (3) (2010) 660–668.
[5] J.G. Broida, S.G. Williamson, A comprehensive introduction to linear algebra, in: Advanced Book Program, Addison-Wesley, 1989.
[6] L. Brugnano, D. Giordano, F. Iavernaro, G. Rubino, An entropy-based approach for a robust least squares spline approximation, J. Comput. Appl. Math. 443 (2024) 115773.
[7] R. Cavoretto, A. De Rossi, F. Dell'Accio, F. Di Tommaso, Fast computation of triangular Shepard interpolants, J. Comput. Appl. Math. 354 (2019) 457–470.
[8] R. Cavoretto, A. De Rossi, F. Dell'Accio, F. Di Tommaso, An efficient trivariate algorithm for tetrahedral shepard interpolation, J. Sci. Comput. 82 (3) (2020).
[9] Q. Chang, C. Deng, M.S. Floater, An interpolatory view of polynomial least squares approximation, J. Approx. Theory 252 (2020) 105360.
[10] W.S. Cleveland, Robust locally weighted regression and smoothing scatterplots, J. Amer. Statist. Assoc. 74 (368) (1979) 829–836.
[11] W.S. Cleveland, LOWESS: A program for smoothing scatterplots by robust locally weighted regression, Amer. Statist. 35 (1) (1981) 54.
[12] P. Craven, G. Wahba, Smoothing noisy data with spline functions, Numer. Math. 31 (4) (1978) 377–403.
[13] P.J. Davis, Interpolation and approximation, in: Dover Books on Mathematics, Dover Publications, 1975.
[14] C. de Boor, A Practical Guide to Splines, Springer, New York, NY, 1978.

[15] C. de Boor, K. Höllig, S. Riemenschneider, Box splines, first ed., in: Applied Mathematical Science, Springer, New York, NY, 1993.
[16] S. De Marchi, F. Dell'Accio, M. Mazza, On the constrained mock-Chebyshev least-squares, J. Comput. Appl. Math. 280 (2015) 94–109.
[17] F. Dell'Accio, F. Di Tommaso, O. Nouisser, N. Siar, Solving Poisson equation with Dirichlet conditions through multinode Shepard operators, Comput. Math. Appl. 98 (2021) 254–260.
[18] F. Dell'Accio, F. Di Tommaso, F. Nudo, Generalizations of the constrained mock-Chebyshev least squares in two variables: Tensor product vs total degree polynomial interpolation, Appl. Math. Lett. 125 (2022) 107732.
[19] J.-J. Fang, C.-L. Hung, An improved parameterization method for B-spline curve and surface interpolation, Comput. Aided Des. 45 (6) (2013) 1005–1028.
[20] S. Farahmand, G.B. Giannakis, Robust RLS in the presence of correlated noise using outlier sparsity, IEEE Trans. Signal Process. 60 (6) (2012) 3308–3313.
[21] S. Farahmand, G.B. Giannakis, D. Angelosante, Doubly robust smoothing of dynamical processes via outlier sparsity constraints, IEEE Trans. Signal Process. 59 (10) (2011) 4529–4543.
[22] G. Farin, Curves and Surfaces for Computer-Aided Geometric Design, third ed., Academic Press, Boston, 1993.
[23] M.S. Floater, M. Reimers, Meshless parameterization and surface reconstruction, Comput. Aided Geom. Design 18 (2) (2001) 77–92.
[24] D.R. Forsey, R.H. Bartels, Hierarchical B-spline refinement, SIGGRAPH Comput. Graph. 22 (4) (1988) 205–212.
[25] J.H. Friedman, Multivariate adaptive regression splines, Ann. Statist. 19 (1) (1991) 1–67.
[26] C. Giannelli, S. Imperatore, A. Mantzaflaris, D. Mokriš, Leveraging moving parameterization and adaptive THB-splines for CAD surface reconstruction of aircraft engine components, in: F. Banterle, G. Caggianese, N. Capece, U. Erra, K. Lupinetti, G. Manfredi (Eds.), Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference, The Eurographics Association, 2023, pp. 125–134.
[27] C. Giannelli, S. Imperatore, A. Mantzaflaris, F. Scholz, BIDGCN: boundary informed dynamic graph convolutional network for adaptive spline fitting of scattered data, Neural Comput. Appl. (2024) accepted, URL: https://hal.science/hal-04313629/.
[28] C. Gu, Cross-validating non-Gaussian data, J. Comput. Graph. Statist. 1 (2) (1992) 169–179.
[29] C. Gu, Smoothing Spline ANOVA Models, Springer, 2002.
[30] A. Gupta, O.P. Kharbanda, V. Sardana, R. Balachandran, H.K. Sardana, A knowledge-based algorithm for automatic detection of cephalometric landmarks on CBCT images, Int. J. Comput. Assist. Radiol. Surg. 10 (2015) 1737–1752.
[31] P.W. Holland, R.E. Welsch, Robust regression using iteratively reweighted least-squares, Comm. Statist. Theory Methods 6 (9) (1977) 813–827.
[32] L. Hu, W. Zhang, NSGA-II approach for proper choice of nodes and knots in B-spline curve interpolation, Comput. Aided Des. 127 (2020) 102885.
[33] T.J. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39) (2005) 4135–4195.
[34] T.M. Inc, Curve fitting toolbox version 3.8 (R2022b), 2024.
[35] H. Kang, F. Chen, J. Deng, Hierarchical box splines, in: 2015 14th International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics, 2015, pp. 73–80.
[36] G. Kiss, C. Giannelli, U. Zore, B. Jüttler, D. Großmann, J. Barner, Adaptive CAD model (re-) construction with THB-splines, Graph. Models 76 (5) (2014) 273–288.
[37] R. Kraft, Adaptive and linearly independent multilevel B–splines, in: A.L. Méhauté, C. Rabut, L.L. Schumaker (Eds.), Surface Fitting and Multiresolution Methods, Vanderbilt University Press, Nashville, 1997, pp. 209–218.
[38] P. Laube, P. Laube, Parametrization in curve and surface approximation, in: Machine Learning Methods for Reverse Engineering of Defective Structured Surfaces, Springer, 2020, pp. 51–95.
[39] E.T. Lee, Choosing nodes in parametric curve interpolation, Comput. Aided Des. 21 (6) (1989) 363–370.
[40] D. Lenz, R. Yeh, V. Mahadevan, I. Grindeanu, T. Peterka, Customizable adaptive regularization techniques for B-spline modeling, J. Comput. Sci. 71 (2023).
[41] Z. Liu, H. Leung, L. Zhou, H.P. Shum, High quality compatible triangulations for 2D shape morphing, in: Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology, 2015, pp. 85–94.
[42] A. Mantzaflaris, An overview of geometry plus simulation modules, in: D. Slamanig, E. Tsigaridas, Z. Zafeirakopoulos (Eds.), Mathematical Aspects of Computer and Information Sciences, in: Lecture Notes in Computer Science, vol. 11989, Springer International Publishing, Cham, 2020, pp. 453–456.
[43] S. Merchel, B. Jüttler, D. Mokriš, Adaptive and local regularization for data fitting by tensor-product spline surfaces, Adv. Comput. Math. 49 (4) (2023) 58.
[44] X. Ning, F. Li, G. Tian, Y. Wang, An efficient outlier removal method for scattered point cloud data, PLoS One 13 (8) (2018) e0201280.
[45] S. Ohrhallinger, J. Peethambaran, A.D. Parakkat, T.K. Dey, R. Muthuganapathy, 2D points curve reconstruction survey and benchmark, Comput. Graph. Forum 40 (2021).
[46] M.R. Osborne, Finite Algorithms in Optimization and Data Analysis, John Wiley & Sons, Inc., USA, 1985.
[47] L. Piegl, W. Tiller, The NURBS Book, 2, Springer, Berlin, Heidelberg, 1997.
[48] C. Runge, Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, Z. Math. Phys. 46 (224–243) (1901) 20.
[49] T. Strutz, Data Fitting and Uncertainty: A Practical Introduction to Weighted Least Squares and Beyond, second ed., Springer Vieweg, 2015.
[50] G. Wahba, Spline models for observational data, in: CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, 1990.
[51] Y. Wang, J. Zheng, Curvature-guided adaptive T-spline surface fitting, Comput. Aided Des. 45 (8) (2013) 1095–1107.
[52] R. Wolke, H. Schwetlick, Iteratively reweighted least squares: Algorithms, convergence analysis, and numerical comparisons, SIAM J. Sci. Stat. Comput. 9 (5) (1988) 907–921.
[53] Y. Zhang, J. Cao, Z. Chen, X. Li, X.-M. Zeng, B-spline surface fitting with knot position optimization, Comput. Graph. (2016).
[54] Z. Zhu, A. Iglesias, L. You, J.J. Zhang, A review of 3D point clouds parameterization methods, in: Computational Science–ICCS 2022: 22nd International Conference, London, UK, June 21–23, 2022, Proceedings, Part III, Springer, 2022, pp. 690–703.
[55] S. Zulqarnain Gilani, F. Shafait, A. Mian, Shape-based automatic detection of a large number of 3D facial landmarks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4639–4648.