Leveraging Moving Parameterization and Adaptive THB-Splines for CAD Surface Reconstruction of Aircraft Engine Components

Carlotta Giannelli, Sofia Imperatore, Angelos Mantzaflaris, Dominik Mokriš

▶ **To cite this version:**

Carlotta Giannelli, Sofia Imperatore, Angelos Mantzaflaris, Dominik Mokriš. Leveraging Moving Parameterization and Adaptive THB-Splines for CAD Surface Reconstruction of Aircraft Engine Components. STAG 2023 - annual international conference Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference 2023, Nov 2023, Matera, Italy. pp.125-134, 10.2312/stag.20231301 . hal-04313150

HAL Id: hal-04313150
https://inria.hal.science/hal-04313150

Submitted on 29 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

STAG: Smart Tools and Applications in Graphics (2023)
K. Lupinetti, F. Banterle, N. Capece, and U. Erra (Editors)

Leveraging moving parameterization and adaptive THB-splines for CAD surface reconstruction of aircraft engine components

Carlotta Giannelli[1], Sofia Imperatore[1], Angelos Mantzaflaris[2] and Dominik Mokriš[3]

[1]Dipartimento di Matematica e Informatica “Ulisse Dini", Università degli Studi di Firenze, Italy
[2]Inria centre at Université Côte d’Azur, Sophia Antipolis, France
[3] MTU Aero Engines AG, Munich, Germany

Abstract

Reconstruction of highly accurate CAD models from point clouds is both paramount and challenging in industries such as aviation. Due to the acquisition process, this kind of data can be scattered and affected by noise, yet the reconstructed geometric models are required to be compact and smooth, while simultaneously capturing key geometric features of the engine parts. In this paper, we present an iterative moving parameterization approach, which consists of alternating steps of surface fitting, parameter correction, and adaptive refinement using truncated hierarchical B-splines (THB-splines). We revisit two existing surface fitting methods, a global least squares approximation and a hierarchical quasi-interpolation scheme, both based on THB-splines. At each step of the adaptive loop, we update the parameter locations by solving a non-linear optimization problem to infer footpoints of the point cloud on the current fitted surface. We compare the behavior of different optimization settings for the critical task of distance minimization, by also relating the effectiveness of the correction step to the quality of the initial parameterization. In addition, we apply the proposed approach in the reconstruction of aircraft engine components from scanned point data. It turns out that the use of moving parameterization instead of fixed parameter values, when suitably combined with the adaptive spline loop, can significantly improve the resulting surfaces, thus outperforming state-of-the-art hierarchical spline model reconstruction schemes.

CCS Concepts

• ***Computing methodologies*** → ***Parametric curve and surface models;***

1. Introduction

Constructing continuous and flexible geometric models that are easy to shape and manipulate is a fundamental requirement in many applications. For example, they can serve for visualization and direct measurements within a design phase or a manufacturing process, as well as for simulations, i.e. performing numerical computations directly on the model or its components. In Computer Aided Design (CAD) and Computer Aided Engineering (CAE) software, spline representations are constructed in terms of B-splines or their non-uniform rational extension (NURBS). This choice relies on the properties of these functions, namely non-negativity, locality, and partition of unity, which results in free-form shaped and flexible geometric models that are easy to manipulate. However, real-world data sets do not usually consist of such models, since the data acquisition procedures, e.g. via laser scanners, diagnostic devices, or photogrammetry schemes, yield raw-data points. Furthermore, depending on the application and the collection method, these data can result in structured point grids, meshes, or scattered point clouds. Therefore, the design of continuous free-from geometric models relies on different data fitting procedures. In this paper, we propose an accurate adaptive approximation framework, applied to industrial scattered data acquired through optical scans of aircraft turbine blade components. We use adaptive spline models combined with a moving parameterization to ensure high flexibility, while simultaneously handling scattered parametric data configurations within the adaptive loop.

The first fundamental step of any parametric fitting method consists of assigning a parameter value to each data point. While this *parameterization* process plays a crucial role it is still an open research topic and several solutions have been proposed, that are not optimal in a universal way. Parameterization methods for triangulated surfaces were proposed in [Flo97, Flo03] and later extended to point clouds [FR01] and periodic surfaces [GJM21]. These methods rely on a barycentric mapping induced from the local neighborhoods of the points. Recently, new data-driven approaches, such as neural networks, have also emerged as viable options for tackling this problem. These include [LFU18, SJ21] for univariate spline and polynomial approximation respectively, while a more general framework was proposed in [DVGIM23, GIMS23] for multivariate spline approximation. We refer the reader to the related survey articles [FH05, ZIYZ22] for more details on the topic of parameterization.

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Often the parameterization and fitting problems are treated separately, that is, first a fixed parameterization is constructed and is kept fixed throughout the fitting algorithm. Hoschek introduced in [Hos88] the idea that the parameterization must also be adjusted to the fitted surface: the so-called intrinsic parametrization (or parameter correction method) is (iteratively) computed starting from a certain initial approximation. The fitted points are projected onto the surface and the projected footpoints take the place of the previous parameter values. The re-fitted surface can then significantly improve the reconstructed geometric model. The core of the method relies on efficient footpoint projection, that is reduced to a non-linear minimization problem. The iterative approach of [Hos88] for finding the footpoint (i.e., the closest point on the surface) has been revisited in [SD03] where a Newton-like method is proposed for the problem. As in all non-linear minimization schemes, the quality of the initial point, the nature of the objective function, and the different stepping strategies play a crucial role in the computation. Among the different contributions that incorporate parameter correction to improve spline approximation and fitting, we just mention the case of *structured* adaptive data fitting with T-splines [WZ13, SFF*19] and of adaptive template mapping [SJŠ19].

As far as surface fitting is concerned, current CAD/CAE technologies rely mostly on tensor-product constructions, which pose several challenges for approximating scattered data point clouds, as well as dealing with noise or outliers or other *local* properties. To this direction, the authors in [RB21] propose a weighted quasi-interpolant, where the definition of suitable weights aims at recovering the local information of the raw data points. A penalized global least squares model is exploited in [MJM23, LYM*23]. In particular, locality is tackled by automatically adapting the penalization power according to the data density. Regarding shape representations, the need to overcome the intrinsic limits of tensor-product spline constructions resulted in the development of new adaptive spline spaces that allow local refinement. Among others, these include T-splines [SZBN03], locally refined (LR) B-splines [DLP13], hierarchical B-splines (HB) [Kra97] and their truncated formulation as truncated hierarchical B-splines [GJS12] (THB). The corresponding family of fitting methods typically accept as input a scattered point cloud, together with a suitable parameterization, and provide as output a locally refined spline geometric model, characterized by a certain prescribed accuracy. In particular, we mention the THB-spline methods developed in [KGZ*14, BGS17, BGGS18] and [BGG*22].

In the present work we follow up on the idea that, as the refinement proceeds in an adaptive setting, not only the geometric model but also the parameter values of each data point should be optimized. In fact, when adaptively designing a parametric geometric model, there is no evidence that the initial (possible) optimality of the parameterization is maintained when changing the approximation space. We revisit two existing surface fitting schemes, a global least squares approximation and a hierarchical quasi-interpolation scheme, both based on THB-splines. At each step of the adaptive loop, we update the parameter locations by solving a non-linear optimization problem to infer footpoints of the point cloud on the currently fitted surface. We compare the behaviour of different optimizers for the critical task of distance minimization, by also relating the effectiveness of the correction step to the quality of the initial parameterization. In addition, we apply the proposed approach in the reconstruction of aircraft engine components from scanned point data. It turns out that the use of a *moving parameterization* instead of fixed parameter values, when suitably combined with the adaptive spline loop, can significantly improve the resulting surfaces, thus outperforming state-of-the-art hierarchical spline model reconstruction schemes.

The paper is organized as follows. Section 2 provides a brief overview of (TH)B-splines constructions. After presenting the scattered data fitting problem, Section 3 introduces adaptive THB-spline approximations with moving parameters. In particular, two THB-spline adaptive fitting algorithms for scattered data [KGZ*14, BGG*22] are revisited with this new perspective. Section 4 shows the benefits of employing the proposed method on a selection of geometric models representing aircraft engine components.

2. B-spline constructions

In this section, we introduce the main concepts of polynomial B-splines and their hierarchical extension.

2.1. B-splines

Let $\Omega = [a,b] \subset \mathbb{R}$ be a real interval, $d \in \mathbb{N}$ a polynomial degree, $k := d+1$ the corresponding order, and let $\boldsymbol{t} := [t_0, \dots, t_{n+k}]$ a vector of non-decreasing real values, with $t_{k-1} \equiv a$ and $t_{n+1} \equiv b$, called *knot-vector*. Thereby, $n+1$ univariate B-splines of degree d over $[a,b]$ can be recursively generated [DB02]. To this purpose, let $\beta_{j,k} : \mathbb{R} \to \mathbb{R}$ indicate the j-th B-spline of order $k \geq 1$, for each $j = 0, \dots, n$. For $u \in \mathbb{R}$, if $k = 1$,

$$\beta_{j,1}(u) = \begin{cases} 1 & \text{if} \quad u \in [t_j, t_{j+1}) \\ 0 & \text{otherwise} \end{cases}$$

and $\beta_{j,1}(t_{n+1}) = 1$. If $k \geq 2$,

$$\beta_{j,k}(u) = \omega_{j,k}(u)\beta_{j,k-1}(u) + \big(1 - \omega_{j+1,k}(u)\big)\,\beta_{j+1,k-1}(u),$$

where $\omega_{j,k} : \mathbb{R} \to \mathbb{R}$ is a piecewise *linear* polynomial of the form

$$\omega_{j,k}(u) = \begin{cases} \frac{u - t_j}{t_{j+k-1} - t_j}, & \text{if} \quad u < t_{j+k-1} \\ 0 & \text{otherwise.} \end{cases}$$

Note that each B-spline $\beta_{j,k}$ for $j = 0, \dots, n$ is a piecewise polynomial function of degree d, which has maximum local smoothness on each subinterval of the partition $\boldsymbol{t}$. On the other hand, the regularity at each unique knot $t_j \in \boldsymbol{t}$ is $d - \mu_j$, where $1 \leq \mu_j \leq d$ is the number of times the knot value t_j appears in $\boldsymbol{t}$. Univariate B-splines are characterized by three key properties: (i) non-negativity, $\beta_{j,k}(u) \geq 0$ for all $u \in \mathbb{R}$, (ii) local support, $\beta_{j,k}(u) = 0$ if $u \notin [t_j, t_{j+k})$, and (iii) partition of unity, $\sum_{j=0}^{n} \beta_{j,k}(u) = 1$ if $u \in [t_{k-1}, t_{n+1}] \equiv [a,b]$. The *spline space* of order k and knots $\boldsymbol{t}$ is defined as $\mathcal{V} := \operatorname{span}\{\beta_{0,k}, \dots, \beta_{n,k}\}$.

Univariate B-splines can be easily extended to higher dimensions by considering a tensor-product construction. In particular, let Ω be a hypercube of $\mathbb{R}^D$, $\boldsymbol{d} = (d_1, \dots, d_D)$ the polynomial multi-degree, $\boldsymbol{k} = (k_1, \dots, k_D)$ the corresponding polynomial multi-order

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

and G the tensor-product mesh, defined by suitable knot vectors in each parametric direction $\boldsymbol{t}_1,\dots,\boldsymbol{t}_D$. A tensor-product B-spline $\beta_j : \mathbb{R}^D \to \mathbb{R}$ is then defined as the product of D univariate B-splines, for each $j \in \Gamma_{\boldsymbol{k}} := \{j = (j_1,\dots,j_D) \,|\, j_h = 1,\dots,n_h, h = 1,\dots,D\}$. Because of the tensor construction, many of the simple algebraic properties of univariate B-splines hold. In particular, non-negativity, locality, and partition of unity. The *tensor-product spline space* of multi-order $\boldsymbol{k}$ with respect to the tensor-product grid G is the space $\mathcal{V} := \operatorname{span}\{\beta_j \,|\, j \in \Gamma_{\boldsymbol{k}}\}$.

2.2. Truncated hierarchical B-splines

When dealing with multivariate settings, the tensor-product constructions allow only a global distribution of the degrees of freedom for the problem at hand, whereas local spline refinement enables the possibility of achieving flexible and accurate models by strongly reducing the total number of degrees of freedom when compared with standard tensor-product B-spline representations. This type of mesh design is intrinsically supported by hierarchical splines [FB88], where local refinement is achieved by introducing tensor–product B-splines on multiple hierarchical levels. A selection mechanism to properly identify a hierarchical B-spline basis was originally introduced in [Kra97].

Let Ω be a hypercube of $\mathbb{R}^D$ and let $\boldsymbol{d}$ be a polynomial multi-degree, while $\boldsymbol{k}$ is the corresponding multi-order. We consider a nested sequence of L tensor-product B-spline spaces $\mathcal{V}^0 \subset \dots \subset \mathcal{V}^{L-1}$ defined on $\Omega \subset \mathbb{R}^D$, so that for each level $\ell = 0,\dots,L-1$, $\mathcal{V}^\ell = \operatorname{span}\left\{\beta_j^\ell \,|\, j \in \Gamma_{\boldsymbol{k}}^\ell\right\}$, where $\Gamma_{\boldsymbol{k}}^\ell$ is the set of indices for the tensor-product B-spline basis of level ℓ. In addition, each $\mathcal{V}^\ell$ is associated with a tensor-product mesh G^ℓ and their (non-empty) quadrilateral elements q are commonly addressed as mesh cells of level ℓ. We also consider a nested sequence of closed domains $\Omega \equiv \Omega^0 \supset \dots \supset \Omega^L = \emptyset$, so that each Ω^ℓ is the union of a subset of cells of the tensor product mesh G^ℓ. The hierarchical mesh is defined as

$$\mathcal{G} := \left\{q \in \mathcal{G}^\ell \,|\, \ell = 0,\dots,L-1\right\},$$

where each $\mathcal{G}^\ell := \left\{q \in G^\ell \,|\, q \subset \Omega^\ell \setminus \Omega^{\ell+1}\right\}$ is called the set of *active cells* of level ℓ.

The hierarchical spline construction consists in replacing any B-spline of level ℓ with support completely contained in $\Omega^{\ell+1}$ by B-splines at successively refined levels. More precisely, the hierarchical B-spline basis is defined as

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

where

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

is the set of indices of *active functions* and $\operatorname{supp}\left(\beta_j^\ell\right)$ denotes the intersection of the support of β_j^ℓ with Ω^0. The corresponding hierarchical space is defined as $\operatorname{span}\{\mathcal{H}_{\boldsymbol{k}}\}$. This simple definition, however, leads to different overlaps of coarse and fine B-spline supports, and partition of unity of the basis is lost. Truncated hierarchical B-splines (THB-splines) were introduced in [GJS12] to reduce the interaction between hierarchical functions at different levels and recover the partition of unity property. For any $\ell = 0,\dots,L-2$, let $s \in \mathcal{V}^\ell \subset \mathcal{V}^{\ell+1}$ a tensor-product spline represented in terms of the basis of the refined space $\mathcal{V}^{\ell+1}$ as

$$s(\boldsymbol{u}) = \sum_{j \in \Gamma_{\boldsymbol{k}}^{\ell+1}} c_j^{\ell+1}(s)\beta_j^{\ell+1}(\boldsymbol{u}), \quad \text{for } \boldsymbol{u} \in \Omega,$$

for suitable coefficients $c_j^{\ell+1}(s)$, for each $j \in \Gamma_{\boldsymbol{k}}^{\ell+1}$. The truncation of $s \in \mathcal{V}^\ell$ at level $\ell+1$ is defined as

$$\operatorname{trunc}^{\ell+1}(s) := \sum_{\substack{j \in \Gamma_{\boldsymbol{k}}^{\ell+1} \\ \operatorname{supp}(\beta_j^\ell) \not\subseteq \Omega^{\ell+1}}} c_j^{\ell+1}(s)\,\beta_j^{\ell+1}$$

and the cumulative truncation with respect to all finer levels is

$$\operatorname{Trunc}^{\ell+1}(s) := \operatorname{trunc}^{L-1}\left(\operatorname{trunc}^{L-2}\left(\dots\left(\operatorname{trunc}^{\ell+1}(s)\right)\dots\right)\right),$$

with $\operatorname{Trunc}^L(s) \equiv s$, for $s \in \mathcal{V}^{L-1}$. Finally, the THB-spline basis for the hierarchical space can be defined as

$$\mathcal{T}_{\boldsymbol{k}} := \left\{\tau_j^\ell = \operatorname{Trunc}^{\ell+1}\left(\beta_j^\ell\right) \,|\, j \in A_{\boldsymbol{k}}^\ell, \ell = 0,\dots,L-1\right\},$$

where the B-spline β_j^ℓ is the *mother* B-spline of the truncated τ_j^ℓ. THB-splines are non-negative, have local support, form a partition of unity, and span the same space of the hierarchical B-spline basis, $\operatorname{span}\{\mathcal{H}_{\boldsymbol{k}}\} \equiv \operatorname{span}\{\mathcal{T}_{\boldsymbol{k}}\}$ [GJS12]. The effectiveness of employing THB-splines both for geometric design and isogeometric analysis has been shown in [GJK*16], among others.

Because of their properties, THB-splines are a desirable tool for building flexible geometric models. They have been here presented in their more general form, where their construction can be developed in any parametric and physical dimension, namely for any $D, N \in \mathbb{N}_{\geq 1}$. In particular, a (TH)B-spline object is a linear combination of (TH)B-splines, defined as

$$\boldsymbol{s}(\boldsymbol{u}) = \sum_{\ell=0}^{L-1} \sum_{j \in A_{\boldsymbol{k}}^\ell} \boldsymbol{c}_j \tau_j^\ell(\boldsymbol{u}), \quad \boldsymbol{u} \in \Omega, \tag{1}$$

with $\boldsymbol{c}_j \in \mathbb{R}^N$ for $j \in A_{\boldsymbol{k}}^\ell$, $\ell = 0,\dots,L-1$. For $D = 1$ a planar ($N = 2$) or spatial ($N = 3$) (TH)B-spline curve can be specified, whereas for $D = 2$ and $N = 3$ a (TH)B-spline surface can be assembled. Throughout the rest of the paper, by moving to industrial applications, we will always assume $D = 2$ and $N = 3$.

3. Adaptive THB-splines fitting with moving parameters

Let $\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3 \,|\, i = 1,\dots,m\}$ be a (noisy) scattered point cloud of m data belonging to the physical space, the problem of surface fitting consists in finding a surface $\boldsymbol{s} : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}^3$, which approximates each point of $\mathcal{P}$, hence $\boldsymbol{s}_i \approx \boldsymbol{p}_i$ for each $i = 1,\dots,m$, where $\boldsymbol{s}_i$ is a point on the surface associated to the observation $\boldsymbol{p}_i$ for each $i = 1,\dots,m$. This results in an exact match $\boldsymbol{s}_i = \boldsymbol{p}_i$, if we require the approximating model $\boldsymbol{s}$ to interpolate the data, for each $i = 1,\dots,m$.

Due to the THB-spline properties and their effectiveness as geometric design tools in CAD/CAE frameworks, we require $\boldsymbol{s}$ to be a THB-spline surface of a certain fixed bi-degree $\boldsymbol{d}$ and bi-order $\boldsymbol{k}$

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

as defined in (1). A crucial problem in any parametric surface reconstruction scheme consists in the data *parameterization*, namely assigning to each point $\boldsymbol{p}_i \in \mathbb{R}^3$ a corresponding parameter value $\boldsymbol{u}_i \in \Omega \subset \mathbb{R}^2$ for $i = 1,\dots,m$ in order to define the parametric set $\mathcal{U} \coloneqq \{\boldsymbol{u}_i \in \Omega \subset \mathbb{R}^2 | \boldsymbol{u}_i = (u_i, v_i), i = 1,\dots,m\}$. Since the parameters encode intrinsic characteristics of the surface representation, estimating a good point cloud parameterization is a fundamental and delicate issue. State-of-the-art THB-splines fitting methods address the parameterization problem, the construction of the hierarchical space, and the definition of the control net separately. In particular, given $\mathcal{P}$, a suitable parameterization $\mathcal{U}$ is computed and, on such a *fixed* parameterization, both the hierarchical space and the approximating spline model are defined by addressing alternately the knot placement and the computation of the control net within an adaptive loop [KGZ*14, BGS17, BGG*22]. In this section instead, we propose a strategy to properly embed the parameterization within the adaptive fitting schemes.

3.1. Initial spline configurations

To approximate the industrial scattered point clouds examined in this paper, the heuristics developed in [FR01] are employed as initial parametrization. These methods consist of two steps: given a point cloud $\mathcal{P}$, firstly its boundary points $\mathcal{P}_B \subset \mathcal{P}$ are individuated and parameterized with univariate techniques, see e.g. [Lee89, PT95, Lim99, SA04, BÖKK20], so that their parametric values lay anticlockwise on $\partial\Omega$. Subsequently, a graph-connectivity is determined, by defining for each interior point $\boldsymbol{p}_i \in \mathcal{P} \setminus \mathcal{P}_B$ a certain neighbourhood, e.g. by determining the k nearest points to $\boldsymbol{p}_i$ for some appropriate $k \in \mathbb{N}_{\geq 1}$, among others. Thereby, the interior point parameters are computed as *weighted* convex combinations of the parameters of their neighbours. Note that, specific choices of neighbourhoods and weights considered in the convex combinations characterize the final parameterization and lead to different results, see [FR01] for more details. In principle, however, any parameterization method can be employed as a starting parameterization to be further improved within the adaptive scheme.

In addition to the parameterization, the design of spline models is characterized by further unknowns, i.e., the *knot* lines *placement*, which defines the initial knot configuration, and, consequently, the control points of the geometric model $\boldsymbol{s}$. In a more general context, the weights associated with the data observations, a smoothness functional and the real factor controlling its influence are all additional potential unknowns. However, their analysis goes beyond the scope of this paper. For interpolation problems, the amount of degrees of freedom is known a priori since it is constrained to the number of data observations. On the other hand, the input data $\mathcal{P}$ hereby considered are industrial real-world data, namely noisy and unevenly distributed. Hence, any interpolation scheme would potentially result in a costly model with as many degrees of freedom as the number of input points, being nevertheless numerically inaccurate. Therefore, we seek a geometric model $\boldsymbol{s}$, which approximate the data $\mathcal{P}$ within a certain tolerance $\varepsilon \in \mathbb{R}_{>0}$, in the sense that $\operatorname{dist}(\boldsymbol{s}_i, \boldsymbol{p}_i) \leq \varepsilon$ for each $i = 1,\dots,m$, where $\operatorname{dist}(\cdot,\cdot)$ is a suitable distance metric. In particular, given the data $\mathcal{P}$, their parameterization $\mathcal{U}$ and a tensor-product mesh G, we exploit two methods for the computations of the control points: a penalized global least squares and a quasi-interpolation scheme. The final amount of degrees of freedom needed to obtain the desired accuracy is usually derived through an iterative process which performs adaptive spline refinement.

3.2. Hierarchical spline approximation

Once an initial parameter and mesh configuration is chosen, any adaptive approximation procedure is characterised by four main steps which are successively repeated: (1) computation of the approximation on the current mesh, (2) error estimation, (3) marking and (4) refinement strategies to suitably identify the adaptive mesh to be used in the next iteration of the adaptive loop. In particular, we revisit the adaptive global least squares method (LS) proposed in [KGZ*14] and the adaptive hierarchical quasi-interpolation (QI) approach proposed in [BGG*22] to define adaptive THB-spline surface fitting schemes with parameter correction.

Within the surface fitting framework, for a fixed THB-spline space, the first step of the adaptive loop consists in computing the control points $\boldsymbol{c}_j^\ell$ for each $j \in A_{\boldsymbol{k}}^\ell$ and $\ell = 0,\dots,L-1$ to define the geometric model in (1). As concerns LS, this is achieved by solving the penalized least squares problem

$$\min_{\substack{\boldsymbol{c}_j^\ell, j \in A_{\boldsymbol{k}}^\ell, \\ \ell=0,\dots,L-1}} \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{s}(\boldsymbol{u}_i) - \boldsymbol{p}_i\|_2^2 + \lambda J(\boldsymbol{s}),$$

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

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

As concerns the QI, each control point $\boldsymbol{c}_j^\ell$ depends on a (local) subset of the input point cloud $\mathcal{P}_j \subset \mathcal{P}$ and its corresponding parameters $\mathcal{U}_j \subset \mathcal{U}$, i.e. $\boldsymbol{c}_j^\ell = \boldsymbol{c}_j^\ell(\mathcal{P}_j, \mathcal{U}_j)$ for each $j \in A_{\boldsymbol{k}}^\ell$ and $\ell = 0,\dots,L-1$ and therefore, the resolution of a global linear system is avoided. Note that by exploiting THB-spline constructions, it is possible to define hierarchical quasi-interpolation schemes without any additional efforts with respect to their tensor-product formulations [SM16]. More specifically, the computation of the QI control net consists of implementing a two-stage algorithm by combining local penalized least squares spline approximations (first stage) with the assembly of the hierarchical quasi-interpolant (second stage). In particular, for each $j \in A_{\boldsymbol{k}}^\ell$ and $\ell = 0,\dots,L-1$, let β_j^ℓ be the *mother* tensor-product B-spline of the truncated τ_j^ℓ, i.e., $\tau_j^\ell = \mathrm{Trunc}^{\ell+1}(\beta_j^\ell)$, and $\Omega_j \subset \Omega$ a suitably chosen local subdomain which has a non-empty intersection with β_j^ℓ, namely $\Omega_j \cap \operatorname{supp}(\beta_j^\ell) \neq \emptyset$. Subsequently, individuate the indices $I_j = \{i \,|\, \boldsymbol{u}_i \in \mathcal{U} \cap \Omega_j\} \subset \{1,\dots,m\}$, so that $n_{min} \leq |I_j| \ll m$ and the related data

$$\mathcal{P}_j = \{\boldsymbol{p}_i \,|\, i \in I_j\}, \quad \mathcal{U}_j = \{\boldsymbol{u}_i \,|\, i \in I_j\}. \tag{3}$$

For more details on the choice of n_{min}, I_j, $\mathcal{P}_j$ and $\mathcal{U}_j$ see [BGG*22]. Finally, denote with $\left\{\beta_r^\ell \,|\, r \in \Lambda_{\boldsymbol{k}}^{\ell,j} \subset \Gamma_{\boldsymbol{k}}^\ell\right\}$ the subset of tensor-

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

product B-splines which do not vanish on Ω_j (by definition β_j^ℓ belongs to it). We then approximate $\mathcal{P}_j$ with

$$\boldsymbol{s}_j(\boldsymbol{u}) = \sum_{r \in \Lambda_{\boldsymbol{k}}^{\ell,j}} \boldsymbol{\alpha}_r^\ell \beta_r^\ell(\boldsymbol{u}) \tag{4}$$

by computing the following penalized tensor-product B-spline *local* approximation,

$$\min_{\boldsymbol{\alpha}_r^\ell, r \in \Lambda_{\boldsymbol{k}}^{\ell,j}} \sum_{i \in I_j} \left\| \boldsymbol{s}_j(\boldsymbol{u}_i) - \boldsymbol{p}_i \right\|_2^2 + \lambda J(\boldsymbol{s}_j),$$

where the penalization term has been chosen again as the thin-plate energy in (2). The computation of all the local approximations (4) for each $j \in A_{\boldsymbol{k}}^\ell$ and $\ell = 0, \dots, L-1$ terminates the first stage. The second stage consists of assembling the global approximation defined in (1). In particular, due to the THB-spline properties, each coefficient $\boldsymbol{c}_j^\ell = \boldsymbol{c}_j^\ell(\mathcal{P}_j, \mathcal{U}_j)$ in (1) directly corresponds to the coefficient $\boldsymbol{\alpha}_j^\ell$ associated with β_j^ℓ in (4).

Subsequently, the second step of the adaptive fitting loop consists of evaluating the THB-spline approximant on the parameter sites $\boldsymbol{u}_i \in \Omega$ related to the data points $\boldsymbol{p}_i$ to compute a suitable error indicator. In particular, we choose the point-wise error distance $\|\boldsymbol{s}(\boldsymbol{u}_i) - \boldsymbol{p}_i\|_2$ for each $i = 1, \dots, m$, among others. The error indicator indicates the region of the domain Ω where additional degrees of freedom are needed to meet the prescribed surface accuracy, by individuating the parametric sites $\boldsymbol{u}_i \in \mathcal{U}$ where it exceeds a certain input threshold, i.e. $\|\boldsymbol{s}(\boldsymbol{u}_i) - \boldsymbol{p}_i\|_2 \geq \varepsilon$.

As concerns the LS scheme, for the making strategy we identify the cells of the current hierarchical level ℓ which contain the parameters $\boldsymbol{u}_i$ identified by the error indicator and mark them for refinement, together with two surrounding rings of cells in the hierarchical mesh. For more details about the feasible configurations of hierarchical meshes with THB-splines for geometric design, see [GJK*16]. In the QI scheme instead, the marking process is naturally implemented on the basis functions. In particular, the basis function of level ℓ which are active on the parameter sites characterized by $\|\boldsymbol{s}(\boldsymbol{u}_i) - \boldsymbol{p}_i\|_2 \geq \varepsilon$ are marked for refinemnt and replaced by basis functions of the successive hierarchical level $\ell + 1$.

Finally, the refinement strategy of the two THB-spline fitting schemes has to be considered. In LS, the marked cells are dyadically split, whereas in QI the refinement strategy is more sophisticated, to better handle scattered data configurations and exploit local adaptivity. In particular, in some situations, the parameter values corresponding to the local data set $\mathcal{U}_j$ can be concentrated in a small part of the support of a marked function, thus splitting its support may affect the quality of the final approximation. The support of each marked basis function is then analyzed as follows: if it contains a minimum number of data points, its dyadic refinement is considered, otherwise the local refinement of the corresponding area is prevented, see [BGG*22] for the details.

At this stage, the hierarchical space has been updated and both in [KGZ*14] and [BGG*22] a new iteration of the adaptive loops begins. We now enrich the hierarchical approximation scheme by updating the current parameterization, after the refinement strategy. This additional step in the adaptive loop enables the definition of adaptive THB-spline fitting with moving parameters.

3.3. Moving parameters

In the context of surface fitting, the parameters update within the described adaptive methodology is performed by adding a parameter correction (PC) routine [Hos88], at the end of the standard adaptive loop. This method consists of locating the points on the geometric model which are the closest to the data points, in terms of Euclidean distance. Given a point cloud $\mathcal{P}$, its parameterization $\mathcal{U}$ and a surface $\boldsymbol{s} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$, the surface closest point problem consists in solving the following minimization problem,

$$\min_{(u_i, v_i)} \frac{1}{2} \left\| \boldsymbol{s}(u_i, v_i) - \boldsymbol{p}_i \right\|_2^2, \qquad \text{for each } i = 1, \dots, m.$$

This two-dimensional nonlinear problem can be explicitly formulated as

$$\left(\boldsymbol{s}(u_i, v_i) - \boldsymbol{p}_i \right)^T \nabla \boldsymbol{s}(u_i, v_i) = 0, \text{ for each } i = 1, \dots, m, \tag{5}$$

where $\nabla \boldsymbol{s}$ indicates the gradient of the surface $\boldsymbol{s}$, and solved by employing a suitable optimizer. In view of (5), the vector connecting the data point $\boldsymbol{p}_i$ to the surface point $\boldsymbol{s}(u_i, v_i)$ has to be orthogonal to the tangent plane of the surface and $\boldsymbol{s}(u_i, v_i)$ is then usually called the *foot-point* of $\boldsymbol{p}_i$ over $\boldsymbol{s}$ for each $i = 1, \dots, m$. The updated parameterization $\bar{\boldsymbol{u}}_i = (\bar{u}_i, \bar{v}_i)$ is defined as the solution of (5), for $i = 1, \dots, m$. Note that after performing one step of PC, the geometric model can be updated by fitting again the surface $\boldsymbol{s}$ to the points $\mathcal{P}$ with the corrected parameters. Consequently, a new projection and the corresponding correction can take place. Usually, a few steps of PC are sufficient to improve the accuracy of the approximation.

The solution of the optimization problem leads to computing the optimal intrinsic parameterization of a given spline model. Within this framework, one step of PC consists of solving (5) for $\boldsymbol{s}$ belonging to the refined hierarchical space and subsequently projecting the points of $\mathcal{P}$ on the refined THB-spline surface $\boldsymbol{s}$ and finally considering their foot-point as the new parameters $\mathcal{U}$. After a certain number of PC steps, the new parameters and the refined adaptive space are employed to implement another iteration of the adaptive loop. Note that the parameter correction is applied globally to the entire approximant. In particular, for the QI scheme, for each local problem the parameters $\mathcal{U}_j$ individuated in (3) are kept fixed.

At the beginning of the next iteration, the corrected geometry is updated by solving the approximation scheme again and the adaptive procedure is iterated. As for standard adaptive fitting strategies, this loop is performed until the maximum point-wise error is within an input tolerance ε or a maximum number of hierarchical level L is reached. Note that PC steps should naturally be embedded in any adaptive scheme, i.e. not limited to fitting problems, since even if the starting parameterization benefits from optimality within the initial space, there is no evidence that optimality is maintained when moving to wider approximation spaces, related to their adaptive extensions.

3.4. Interaction of foot-point projection with adaptive spline configurations

Finding successfully the foot point projection of a point on a surface is a challenging open problem, see e.g. [KS14] and references therein. As far as parameter correction is concerned, the foot-point

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

projection in [Hos88] is performed iteratively, by linearizing the problem in (5). Subsequently, in [SD03], the iterative procedure has been replaced by the application of a Newton-like approach. Due to the locality of Newton-like gradient methods, to properly correct the parameters, it is paramount to start from a good initial parameterization guess as well as with a reasonably good geometric model.

Figure 1 shows the consequences of incorporating the PC routine at a too-early stage of adaptive fitting schemes. More specifically, we fit a point cloud $\mathcal{P}$ of an industrial *tensile* part, about $2.5 \cdot 10^{-2}$m long, and consisting of 9281 scattered data, with the LS scheme presented in [KGZ*14]. An initial tensor-product B-spline approximation of bi-degree $\boldsymbol{d} = (2,2)$ is built on a *coarse* 4×4 tensor-product mesh and illustrated in Figure 1 (a). This initial coarse mesh is then adaptively refined, using a criterion of error threshold with $\varepsilon = 5\text{e}{-5}$m, leading to a final THB-spline model with 1136 degrees of freedom (DOFs) that registers a maximum error (MAX) of 8.333e−5m. The THB-spline approximation obtained by introducing 1 PC step at each iteration of the adaptive loop is shown in Figure 1 (b) and is characterized by 969 DOFs, MAX error of 3.059e−4m. Consequently, we may note that using the inaccurate approximation shown in Figure 1 (a) to compute the foot-point projections and correct the parameters, leads to a final fitting result that is not suitable for further processing due to self intersections seen in Figure 1 (b). We then consider as initial spline configuration the tensor-product B-spline approximation built on a tensor-product mesh with two additional dyadic refinements with respect to Figure 1 (a), i.e., with a 16×16 mesh, as shown in Figure 1 (c). These settings lead to stable foot point projection, due to the quality of the initial approximation and the LS scheme with moving parameters results in a model with 607 DOFs and MAX error 8.467e−5m, shown in Figure 1 (d).

The choice of the optimizer and its settings are fundamental for the successful computation of the foot-point projections. Here, we consider the LS adaptive fitting method with moving parameters and compare the results with respect to the use of different optimizers and optimization settings. In particular, we start with the configuration of Figure 1 (c), i.e. bi-degree $\boldsymbol{d} = (2,2)$ and a 16×16 tensor-product mesh, and apply 1 PC step at each adaptive iteration. For a better comparison, we also fix the total number of adaptive refinement steps. The results are shown in Tables 1. As a benchmark, we consider the commercial library Parasolid [Sie]. We can see that taking a too-big minimum step s in the gradient descent method (GD) leads to a final result that can be *worse* than not moving the parameters at all, with respect to to the number of DOFs of the final models and its accuracy in terms of MAX and MSE errors. However, if carefully fine-tuned, the results obtained by Parasolid in terms of DOFs, MAX and MSE, can be reached as in this case for $s = 1\text{e}{-12}$. In the following numerical examples, we employed the hybrid limited memory Broyden-Fletcher-Goldfarb-Shanno method (HLBFGS) [Liu] with minimum step length $s = 1\text{e}{-9}$, since it has always shown to provide extremely similar results to Parasolid in all the considered examples. The results obtained employing HLBFGS for the above-mentioned configuration are also shown in Table 1.

Finally, we remark that including the parameter correction routine on a proper initial guess and choosing suitable settings for the optimizer leads to outperforming the state-of-the-art adaptive fitting schemes as we prove in the numerical examples of Section 4.

method	pts $< \varepsilon$	MAX (m)	MSE (m^2)	DOFs
no PC	92.770%	1.96820e−4	7.29588e−10	751
GD, $s = 1\text{e}-9$	91.908%	2.92276e−4	1.06473e−9	748
GD, $s = 1\text{e}-10$	96.875%	1.62567e−4	4.42720e−10	690
GD, $s = 1\text{e}-11$	99.289%	8.47310e−5	2.40847e−10	607
GD, $s = 1\text{e}-12$	99.310%	8.46760e−5	2.37395e−10	607
HLBFGS	99.310%	8.46695e−5	2.37373e−10	607
Parasolid	99.310%	8.46716e−5	2.37358e−10	607

Table 1: *Influence of optimizers used for the foot-point computation loop starting from configuration of Figure 1 (c).*

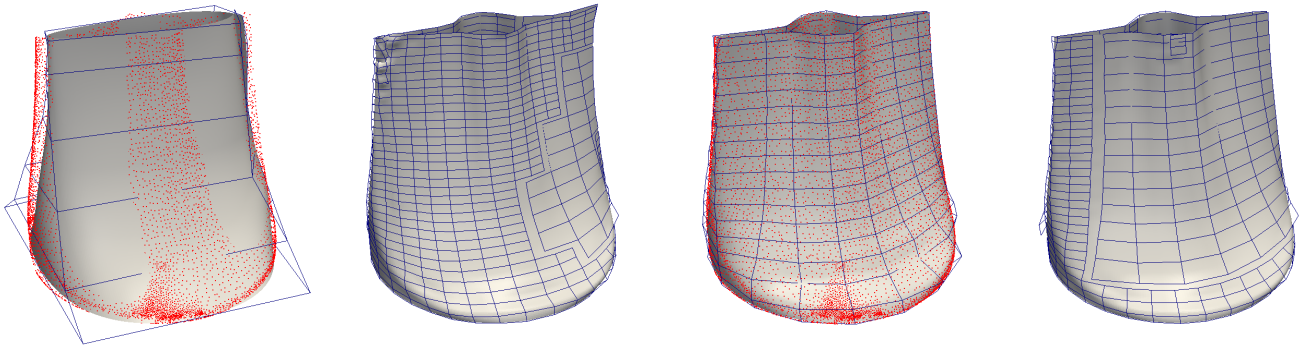
4. Numerical examples

Throughout this section, we use two fitting methods, namely the global adaptive least-squares approximation with THB-splines (LS) and the quasi-interpolation based scheme using local spline approximation (QI), both described in the previous section, to assess the benefits of the moving parameterization approach. We compare the results with and without moving parameters on scanned data provided by MTU Aero Engines. All experimentations were performed using the open-source library G+Smo [JLM*14, Man20], available at https://github.com/gismo.

4.1. Example: Tensile

In this section, we revisit the second configuration of Example 1 in [BGG*22], where the point cloud of the tensile part considered in the previous section, is initially approximated by a bi-quadratic spline on a tensor-product basis defined on a 4×16 mesh with tolerance $\varepsilon = 5\text{e}-5$m. We compare the performance of the QI scheme without parameter correction (0 PC) and with one or more rounds of PC after the adaptive refinement step. The HLBFGS optimizer with minimum step-size $s = 1\text{e}{-9}$ is used for the foot point projection. In particular, the resulting geometries when applying 0, 1 or 2 rounds of PC after each refinement step are shown in Figure 2. We can observe that all the fitting surfaces are of good quality. Nevertheless, in this case suitably embedding the parameter correction routine in the adaptive loop led to gaining the same precision in terms of MAX and MSE errors, while using fewer degrees of freedom to reconstruct the final geometric model, as summarized in the first three lines of Table 2. In particular, applying 3 steps of PC after each hierarchical mesh refinement produces an approximation with 15.20% fewer DOFs with respect to the one achieved with the standard adaptive QI scheme, while simultaneously reducing the MAX error by 9.89% and the MSE by 17.20%. Note that additional rounds (from 4 to 6 in Table 2) of parameter correction, yield diminishing improvements, since the DOFs are still reduced, but there is no gain in terms of MAX and MSE. This suggests that from 1 to 3 PC steps are in general a proper choice when including the parameter correction within adaptive spline fitting schemes.

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.



(a) *coarse initial B-spline model* **(b)** *THB-spline model (969 DOFs)* **(c)** *refined initial B-spline model* **(d)** *THB-spline model (607 DOFs)*

Figure 1: *Adaptive THB-spline LS with moving parameters: effect of the initial tensor-product B-spline approximation. The THB-spline model (b) constructed from the initial configuration (a) defined on a* 4×4 *tensor-product mesh is affected by self-intersection due to the too coarse initial mesh. The THB-spline model (d) constructed from the initial configuration (c) defined on a* 16×16 *tensor-product mesh is successfully computed.*

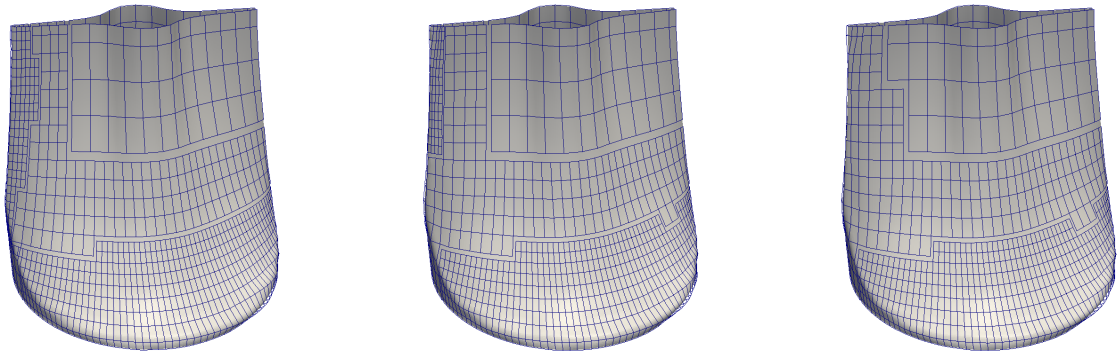


Figure 2: *Adaptive THB-spline QI with moving parameters in Example 4.1. From left to right: models computed with 0, 1, and 2 PC steps.*

method	%pts $< \varepsilon$	MAX (m)	MSE (m^2)	DOFs	
QI + 0 PC	99.364	8.10814e−5	1.41611e−10	1960	
QI + 1 PC	99.332	8.16606e−5	1.40676e−10	1834	6.43%
QI + 2 PC	99.397	7.96751e−5	1.25243e−10	1718	12.35%
QI + 3 PC	99.461	7.30878e−5	1.17258e−10	1662	15.20%
QI + 4 PC	99.149	8.65881e−5	1.21293e−10	1659	15.36%
QI + 5 PC	99.020	8.51664e−5	1.23207e−10	1621	17.30%
QI + 6 PC	99.106	9.18625e−5	1.15098e−10	1621	17.30%

Table 2: *Analysis in terms of points within the prescribed tolerance (%pts* $< \varepsilon$*), MAX or MSE, and final number of DOFs for the adaptive THB-spline QI scheme in Example 4.1. The percentages of the DOFs reduction with respect to QI without PC are also reported.*

4.2. Example: Blade

In this example, we revisit the second example presented in [BGG*22]. However, in view of the considerations of Section 3.4, the initial tensor-product basis of bidegree $\boldsymbol{d} = (3,3)$ is now more refined, i.e. we start with a 8×8 tensor product mesh. The results are still comparable, as the lowest level basis is entirely refined in the configuration considered in [BGG*22]. We then fit the set of 27191 measured data points from a blade geometry that has a length of about $5 \cdot 10^{-2}$m. We compare LS and QI, by always considering as smoothing coefficient $\lambda = 1e-8$. In both cases, we stop when at least 95% of data points are within the tolerance $\varepsilon = 2e-5$m.

Figure 3 shows the final THB-spline models obtained with LS (top) and QI (bottom) when 0, 1, or 2 PC steps are considered in the adaptive THB-spline fitting schemes. In particular, we can observe that the QI geometries are characterized by a smaller amount of oscillations along the sharp features. To have more insights into this phenomenon, we present the reflection lines on the different THB-spline models in Figure 4. By analyzing the top and bottom row of this figure from left to right, we see that the PC improves the

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

surface quality by reducing the oscillations near the top left corner. Moreover, by comparing the LS (top) and QI (bottom) results in each column, we can see that the oscillations just under the feature in the top right area of the blade are reduced in the THB-spline models obtained with QI scheme.

The quantitative analysis for this example is conducted in Table 3. In particular, both for LS and QI schemes, we report the final percentage of points within the tolerance, the MAX and MSE errors, as well as the number of DOFs, when 0, 1, or 2 parameter corrections are considered. Note that the stopping criterion (95% pts $< 2e-5m$) is met in all cases, with a significant reduction ($\sim$70%) of DOFs both for LS and QI schemes when 1 or 2 PC steps are applied. This is due to the fact that the prescribed precision is achieved two iterations earlier, preventing the introduction of two additional hierarchical levels. This can be explicitly seen in Figure 5 (LS) and 6 (QI) where the number of points below the prescribed tolerance versus the amount of DOFs at each adaptive iteration is reported. In this example, either one or two PC steps, after each refinement procedure, are a good choice, since the results are very similar, iteration by iteration, both for LS and QI.

5. Conclusions

In this work, we enhance the power of THB-spline approximations with an update of the point parameters in each adaptive fitting step, by suitably moving the parameterization within the iterative loop, both for least squares and quasi-interpolation schemes. Our study reveals that using the parameter correction step can improve the fitting results while also reducing the number of degrees of freedom required to achieve a certain accuracy. We observe that this correction can lead to earlier termination of the adaptive process, thus providing more compact models with less refinement depth. However, if used on a coarse fitting, the effect can be negative, since it can lead to an invalid parameterization and/or self-intersections in the model. It is then important to have an initial fit that is close enough to the point cloud. The robustness of the footpoint projection is also paramount for the overall process and needs to be accurate enough to gain quality and degrees of freedom.

Acknowledgements

CG acknowledges the contribution of the National Recovery and Resilience Plan, Mission 4 Component 2 – Investment 1.4 – CN_00000013 "CENTRO NAZIONALE HPC, BIG DATA E QUANTUM COMPUTING", spoke 6. CG and SI are members of the INdAM group GNCS. The INdAM-GNCS support is gratefully acknowledged. CG, SI, AM also acknowledge the PHC GALILEE project 47786N and AM acknowledges H2020 Marie Skłodowska-Curie grant GRAPES No. 860843.

References

[BGG*22] BRACCO C., GIANNELLI C., GROSSMANN D., IMPERATORE S., MOKRIŠ D., SESTINI A.: THB-spline approximations for turbine blade design with local B-spline approximations. In *Mathematical and Computational Methods for Modelling, Approximation and Simulation*, Barrera D., Remogna S., Sbibih D., (Eds.). Springer International Publishing, 2022, pp. 63–82.

[BGGS18] BRACCO C., GIANNELLI C., GROSSMANN D., SESTINI A.: Adaptive fitting with THB-splines: Error analysis and industrial applications. *Computer Aided Geometric Design 62* (2018), 239–252.

[BGS17] BRACCO C., GIANNELLI C., SESTINI A.: Adaptive scattered data fitting by extension of local approximations to hierarchical splines. *Computer Aided Geometric Design 52* (2017), 90–105.

[BÖKK20] BALTA C., ÖZTÜRK S., KUNCAN M., KANDILLI I.: Dynamic centripetal parameterization method for B-spline curve interpolation. *IEEE Access 8* (2020), 589–598.

[DB02] DE BOOR C.: *A Practical Guide to Splines*, vol. 27 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.

[DLP13] DOKKEN T., LYCHE T., PETTERSEN K. F.: Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design 30*, 3 (2013), 331–356.

[DVGIM23] DE VITA M., GIANNELLI C., IMPERATORE S., MANTZAFLARIS A.: Parameterization learning with convolutional neural networks for gridded data fitting. *Lectures Notes in Networks and Systems* (2023), in press.

[FB88] FORSEY D. R., BARTELS R. H.: Hierarchical b-spline refinement. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1988), SIGGRAPH '88, Association for Computing Machinery, p. 205–212.

[FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. *Adv. in multiresolution for geom. modelling* (2005), 157–186.

[Flo97] FLOATER M. S.: Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design 14*, 3 (1997), 231–250.

[Flo03] FLOATER M. S.: Mean value coordinates. *Computer aided geometric design 20*, 1 (2003), 19–27.

[FR01] FLOATER M. S., REIMERS M.: Meshless parameterization and surface reconstruction. *Computer Aided Geometric Design 18*, 2 (2001), 77–92.

[GIMS23] GIANNELLI C., IMPERATORE S., MANTZAFLARIS A., SCHOLZ F.: Learning meshless parameterization with graph convolutional neural networks. *Lectures Notes in Networks and Systems* (2023), in press.

[GJK*16] GIANNELLI C., JÜTTLER B., KLEISS S. K., MANTZAFLARIS A., SIMEON B., ŠPEH J.: THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering 299* (2016), 337–365.

[GJM21] GROISS L., JÜTTLER B., MOKRIŠ D.: 27 variants of Tutte's theorem for plane near-triangulations and an application to periodic spline surface fitting. *Comp. Aided Geom. Design 85* (2021), 101–975.

[GJS12] GIANNELLI C., JÜTTLER B., SPELEERS H.: THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design 29*, 7 (2012), 485–498. Geometric Modeling and Processing 2012.

[Hos88] HOSCHEK J.: Intrinsic parametrization for approximation. *Computer Aided Geometric Design 5*, 1 (1988), 27–31.

[JLM*14] JUETTLER B., LANGER U., MANTZAFLARIS A., MOORE S., ZULEHNER W.: Geometry + simulation modules: Implementing isogeometric analysis. *Proc. Appl. Math. Mech. 14*, 1 (2014), 961–962.

[KGZ*14] KISS G., GIANNELLI C., ZORE U., JÜTTLER B., GROSSMANN D., BARNER J.: Adaptive CAD model (re-) construction with THB-splines. *Graphical models 76*, 5 (2014), 273–288.

[Kra97] KRAFT R.: Adaptive and Linearly Independent Multilevel B-Splines. In *Surface Fitting and Multiresolution Methods*, Le Méhauté A., Rabut C., Schumaker L. L., (Eds.). Vanderbilt University Press, 1997, pp. 209–218.

[KS14] KO K., SAKKALIS T.: Orthogonal projection of points in CAD/CAM applications: an overview. *Journal of Computational Design and Engineering 1*, 2 (2014), 116–127.

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

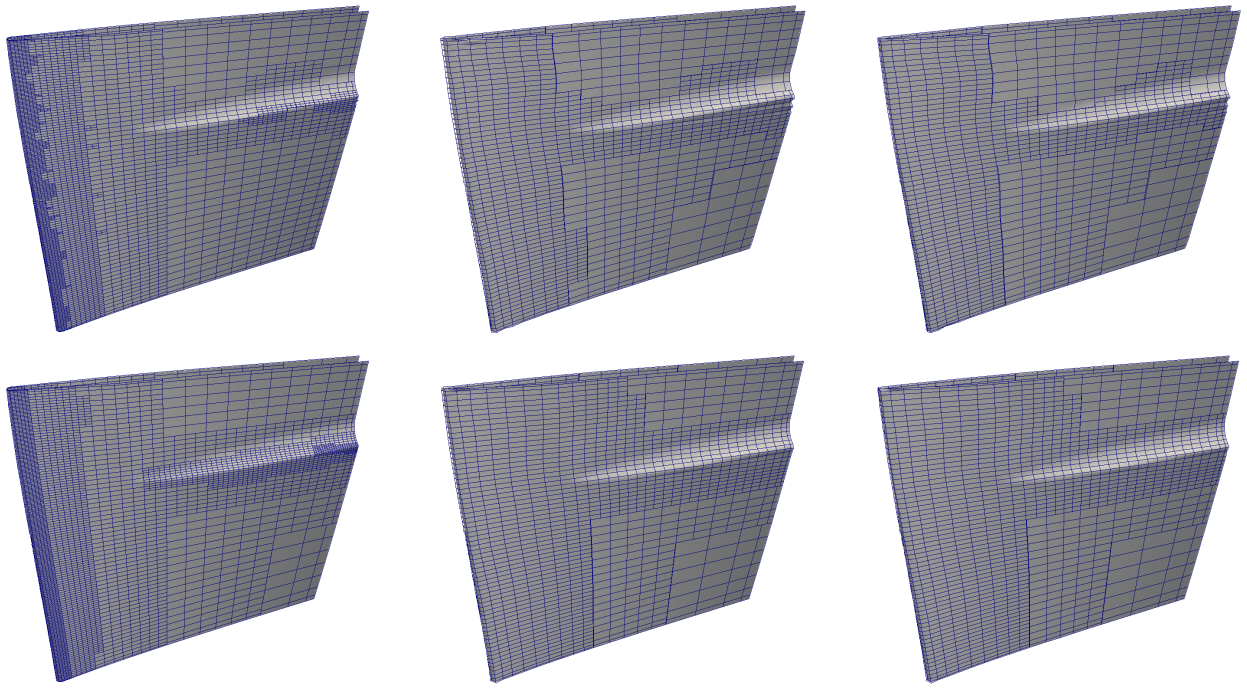


Figure 3: *THB-spline models with control nets in Example 4.2. Top row: LS, bottom row: QI. From left to right: 0, 1, and 2 PC steps.*

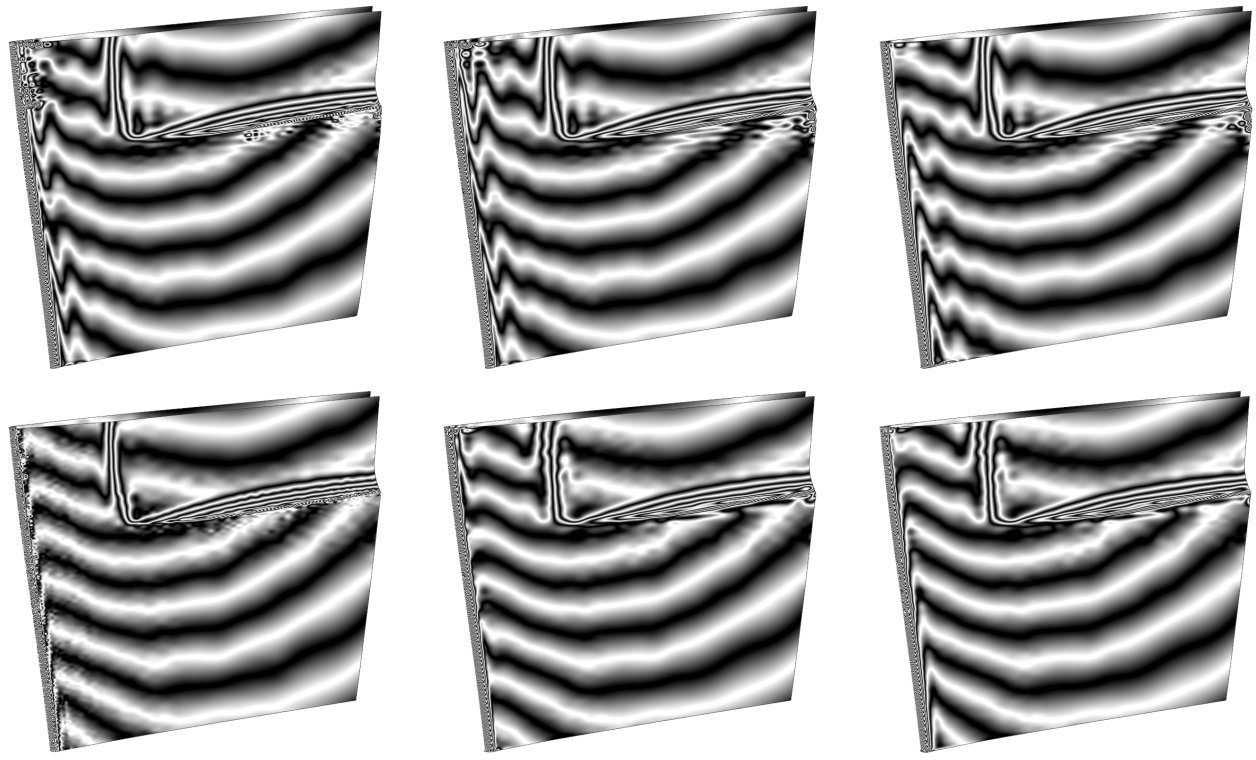


Figure 4: *THB-spline models with reflection lines in Example 4.2. Top row: LS, bottom row: QI. From left to right: 0, 1, and 2 PC steps.*

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.

method	%pts < ε	MAX (m)	MSE (m^2)	DOFs	
LS + 0 PC	99.996	2.11546e−5	6.31779e−12	8164	
LS + 1 PC	99.188	4.04400e−5	2.48002e−11	2314	71.66%
LS + 2 PC	99.651	2.90474e−5	2.37248e−11	2212	72.91%
QI + 0 PC	99.985	2.74178e−5	1.00013e−11	8278	
QI + 1 PC	97.422	8.85969e−5	4.71282e−11	2554	69.15%
QI + 2 PC	95.969	5.71231e−5	5.92761e−11	2539	69.33%

Table 3: *Analysis in terms of points within the tolerance (%pts < ε), MAX or MSE, and number of DOFs for the adaptive THB-spline LS and QI schemes in Example 4.2. The percentages of the DOFs reduction with respect to LS and QI without PC are also reported.*

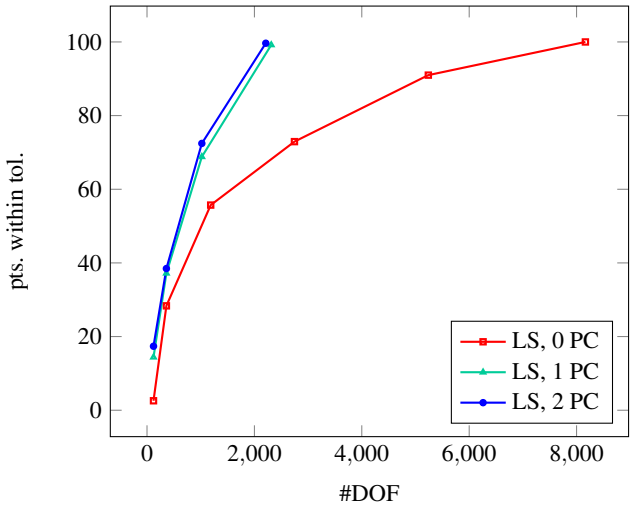


Figure 5: *Percentage of points within the prescribed tolerance during the THB-spline LS scheme in Example 4.2.*

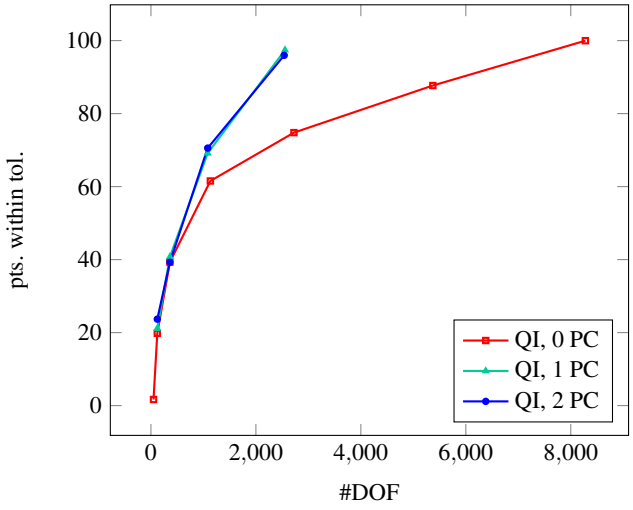


Figure 6: *Percentage of points within the prescribed tolerance during the THB-spline QI scheme in Example 4.2.*

[Lee89] LEE E. T.: Choosing nodes in parametric curve interpolation. *Computer-Aided Design 21*, 6 (1989), 363–370.

[LFU18] LAUBE P., FRANZ M. O., UMLAUF G.: Deep Learning Parametrization for B-Spline Curve Approximation. In *2018 International Conference on 3D Vision (3DV)* (2018), pp. 691–699.

[Lim99] LIM C.-G.: A universal parametrization in b-spline curve and surface interpolation. *Comp. Aided Geom. Design 16*, 5 (1999), 407–422.

[Liu] LIU Y.: HLBFGS: a hybrid l-bfgs optimization framework. URL: https://xueyuhanlang.github.io/software/HLBFGS/.

[LYM*23] LENZ D., YEH R., MAHADEVAN V., GRINDEANU I., PETERKA T.: Customizable adaptive regularization techniques for B-spline modeling. *Journal of Computational Science 71* (2023).

[Man20] MANTZAFLARIS A.: An overview of geometry plus simulation modules. In *Mathematical Aspects of Computer and Information Sciences* (2020), Slamanig D., Tsigaridas E., Zafeirakopoulos Z., (Eds.), Springer, pp. 453–456.

[MJM23] MERCHEL S., JÜTTLER B., MOKRIŠ D.: Adaptive and local regularization for data fitting by tensor-product spline surfaces. *Advances in Computational Mathematics 49*, 4 (2023), 58.

[PT95] PIEGL L., TILLER W.: *The NURBS Book*. Springer-Verlag, 1995.

[RB21] RAFFO A., BIASOTTI S.: Weighted quasi-interpolant spline approximations: Properties and applications. *Numerical Algorithms 87* (2021), 819–847.

[SA04] SHAMSUDDIN S. M. H., AHMED M. A.: A hybrid parameterization method for nurbs. In *Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004.* (2004), IEEE, pp. 15–20.

[SD03] SAUX E., DANIEL M.: An improved hoschek intrinsic parametrization. *Computer Aided Geometric Design 20*, 8-9 (2003), 513–521.

[SFF*19] SHANG C., FU J., FENG J., LIN Z., LI B.: Effective reparameterization and GA based knot structure optimization for high quality t-spline surface fitting. *Computer Methods in Applied Mechanics and Engineering 351* (2019), 836–859.

[Sie] SIEMENS SOFTWARE: Parasolid. URL: https://www.plm.automation.siemens.com/global/en/products/plm-components/parasolid.html.

[SJ21] SCHOLZ F., JÜTTLER B.: Parameterization for polynomial curve approximation via residual deep neural networks. *Computer Aided Geometric Design 85* (2021), 101977.

[SJŠ19] SAJAVIČIUS S., JÜTTLER B., ŠPEH J.: Template Mapping Using Adaptive Splines and Optimization of the Parameterization. In *Advanced Methods for Geometric Modeling and Numerical Simulation* (2019), Giannelli C., Speleers H., (Eds.), vol. 35 of *Springer INdAM Series*, Springer International Publishing, pp. 217–238.

[SM16] SPELEERS H., MANNI C.: Effortless quasi-interpolation in hierarchical spaces. *Numerische Mathematik 132*, 1 (2016), 155–184.

[SZBN03] SEDERBERG T. W., ZHENG J., BAKENOV A., NASRI A.: T-splines and T-NURCCs. *ACM transactions on graphics (TOG) 22*, 3 (2003), 477–484.

[WZ13] WANG Y., ZHENG J.: Curvature-guided adaptive T-spline surface fitting. *Computer-Aided Design 45*, 8 (2013), 1095–1107.

[ZIYZ22] ZHU Z., IGLESIAS A., YOU L., ZHANG J. J.: A review of 3D point clouds parameterization methods. In *International Conference on Computational Science* (2022), Springer, pp. 690–703.

© 2023 The Author(s)
Eurographics Proceedings © 2023 The Eurographics Association.