An approach to construct a three-dimensional isogeometric model from μ-CT scan data with an application to the bridge of a violin ☆

Sandra Marschke [b,*], Linus Wunderlich [d], Wolfgang Ring [b], Klaus Achterhold [a], Franz Pfeiffer [a,c]

[a] *Chair of Biomedical Physics, Technical University of Munich, James-Franck-Str. 1, 85748 Garching, Germany*
[b] *Institute of Mathematics, University of Graz, Heinrichstraße 36, 8010 Graz, Austria*
[c] *Department of Diagnostic and Interventional Radiology, Klinikum rechts der Isar, Technical University of Munich, Munich, Germany*
[d] *School of Mathematical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom*

ARTICLE INFO

ABSTRACT

We present an algorithm to build a ready to use isogeometric model from scan data gained by a μ-CT scan. Based on a three-dimensional multi-patch reference geometry, which includes the major topological features, we fit the outline, then the cross-section and finally the three-dimensional geometry. The key step is to fit the outline, where a non-linear least squares problem is solved with a Gauss-Newton approach presented by Borges and Pastva (2002). We extend this approach by a regularisation and a precise interpolation of selected data points. The resulting NURBS geometry is ready for applying isogeometric analysis tools for efficient numerical simulations.

As a particular example we examine the scan data of a violin bridge and present the complete workflow from the μ-CT scan up to the numerical simulation based on isogeometric mortar methods. We illustrate the relevance of the constructed geometry with a vibro-acoustical application.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

In mechanical simulation problems concerning complex-shaped real world objects, conventional geometric measurement methods often lead to dissatisfactory results. A remedy in such cases may be a μ-CT scan of the considered object. In this paper we address the question how to construct an accurate spline representation of the object described by such a data set, which is ready for a numerical simulation. The complete workflow based on a non-linear least-squares problem (Borges and Pastva, 2002) and isogeometric mortar methods (Brivadis et al., 2015) is presented in detail throughout this paper.

Fitting spline geometries is a long-known problem as splines were soon recognised to have good approximation properties, which are favourable for geometric design (Schumaker, 2007). A crucial part of the fitting process is the parametrisation of the geometry. When we fix the parameter values for the evaluation points of the spline curve which shall be fitted to a given data point, the optimisation of the spline geometry reduces to a quadratic optimisation problem, see e.g.

☆ Editor: B. Juettler.
* Corresponding author.
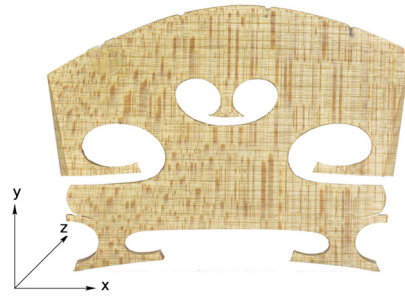*E-mail address:* sandra.marschke@uni-graz.at (S. Marschke).

Fig. 1. Photo of the scanned violin bridge.

(Dierckx, 1995; Piegl and Tiller, 1997). For more precise and flexible results there exist various non-linear optimisation approaches which include the position of the knots (Zhang et al., 2016), the parameter values (Weiss et al., 2002) or different kinds of error projection methods between the data and the fitted surface (Bo et al., 2012). While the approaches mentioned above either do not incorporate the parameterization of the data points or determine the parameterization within a subsequent updating process, we follow the approach of Borges and Pastva (2002). Here both the parameter values and the control points occur as optimisation variables in the cost functional and are optimised simultaneously by employing a Gauss-Newton algorithm. As we observed overfitting for some splines, we regularise the non-linear least squares problem. Moreover, in order to retain characteristic points of the scanned object, we present a way to precisely interpolate selected data points.

In the last years the use of progressive iterative approximation (PIA) became increasingly popular (Carnicer et al., 2012; Deng and Lin, 2014; Liu et al., 2018). These methods concentrate on the recursive construction of optimal control points and are advantageous where only a small set of measured data is available and where computational speed is a crucial issue. In our approach we concentrate on the quality of the model geometry and the accurate representation of certain important geometrical features. We therefore chose a more traditional regularised least-squares data-fit approach which involves several tunable parameters. This helps to enhance the desirable features in the model. As already mentioned above, we also incorporate the parametric values for the given data points into the optimisation process. This allows us to fit a large set of data points while keeping the amount of necessary control points limited and also improves the accuracy of the fit.

We also mention other approaches to surface fitting which make use of genetic algorithms (Gàlvez et al., 2012) or adaptive group testing (Ravari and Taghirad, 2016). These methods consist of multiple stages, whereas our algorithm comprises the complete optimisation procedure into one step.

Processing CT scan data is of high relevance in medical sciences. Apart from their obvious diagnostic values, these data are often the basis for modelling medical and biological processes (Grove et al., 2010). Here our process might point out an alternative to recent techniques which, due to data noise, often depend on pre-smoothing of the data set and thus suffer from a significant loss of information. Another related field of application is as-built modelling in engineering (Dimitrov and Golparvar-Fard, 2014; Patraucean et al., 2015).

We illustrate our approach by the particular example of a violin bridge, which is depicted in Fig. 1. Such a bridge is made out of wood, i.e., an orthotropic material, and about 42 mm wide, 33 mm high and 3 mm thick. As small changes of its shape are already causing clearly noticeable changes in its acoustic properties (Fletcher and Rossing, 1998; Woodhouse, 2005), a highly accurate computational geometry is essential for precise vibro-acoustical simulations.

Our example and also the mentioned applications in medical sciences and engineering often deal with objects of non-trivial shape. We account for this by presenting a method constructing multi-patch surfaces of real world CAD models. It thereby can be an alternative to other techniques for the construction of multi-patch geometries. An example is (Seiler et al., 2018) which has the benefit of leading to $G1$-smoothness of the composite surface. Other examples are (Schmidt et al., 2012; Kim et al., 2010), where trimmed NURBS get employed.

In the following chapters, we present the complete process leading to the numerical simulation of the vibro-acoustics of the violin bridge. The first steps are the acquisition of measurement data from a μ-CT scan (Kak and Slaney, 1988; Buzug, 2008) and the pre-processing of the generated data. Then a spline volume is fitted to the data. Finally, we solve a vibro-acoustical eigenvalue problem with isogeometric methods (Cottrell et al., 2009; Beirão Da Veiga et al., 2014). Because of its complexity, we decompose the geometry of the violin bridge into 16 NURBS patches which are coupled weakly by means of isogeometric mortar methods (Brivadis et al., 2015).

This article is structured as follows. In Chapter 2 we explain the required background about the details and output of the μ-CT scan and the fundamental properties of spline geometries. In the following chapters, we explain our approach via a non-linear least squares problem for the data fit to obtain optimal control points and parameter values of the fitted NURBS functions:. Chapter 3 contains the fit of the one-dimensional outline. This is extended to the final three-dimensional geometry in Chapter 4 within two sub-steps, first the construction of a two-dimensional multi-patch surface (see Chapter 4.1) and then a constrained quadratic optimisation in the third dimension (see Chapter 4.2). Chapter 5 points out the capabilities of the algorithm with some detailed examples. In Chapter 6 we show a concrete application where we use the constructed

violin bridge spline geometry to solve the eigenvalue problem of linear elasticity on it. The main findings of the article are summarized in Chapter 7.

2. Background on μ-CT scan data generation and spline geometries

In this section, the fundamental background of the geometry fit is explained, first the μ-CT scan of the physical object and then the basics about spline geometries that are fitted to these measured data.

2.1. Acquiring μ-CT scan data

The tomographic scans of the violin bridge were conducted by a recently developed μ-CT device, a 'v|tome|x s' (GE Sensing & Inspection Technologies GmbH, Ahrensburg, Germany), at the Technical University of Munich. Its specific technical setup and mode of operation is explained in more detail in the following. For an in-depth background on the physics and mathematics of computed tomography (CT) we refer to Kak and Slaney (1988), Buzug (2008).

Whereas in clinical CTs, X-ray tube and detector rotate around the patient, in μ-CTs the sample rotates while tube and detector stay fixed. We used a 'xs 240 D' direct X-ray tube (GE, Germany) operated at 60 kVp voltage and 130 µA current. The resulting polychromatic spectrum was filtered by a 200 µm aluminum filter to reduce beam hardening artefacts (Kak and Slaney, 1988; Buzug, 2008). The minimum spot size of this tube is 7 µm (Singhal et al., 2013), but it can range up to about 10 µm for the tube settings like the ones used for the experiment (Rueckel et al., 2014). A 1000x1000 pixel detector 'DXR-250RT' (GE, Germany) was used without further binning of the pixels of 200 µm x 200 µm size. It is an amorphous Si flat panel detector with a CsI scintillator converting the X-rays into visible light. 1601 projections over 360° were taken by rotating the sample. For every rotation angle 5 projections of 2 sec accumulation time were collected. The first image was always discarded to reduce artefacts due to the afterglow of the CsI scintillator. The 4 remaining projections were averaged to increase the signal to noise ratio. The violin bridge was held by Polystyrene foam to minimize additional absorption and to ensure a stable mounting of the sample at the same time. The reconstruction of the projection data was done with a standard filtered back projection software with cone-beam geometry (GE, Phoenix datos|x). With a fixed focus detector distance of 811.72 mm and a focus object distance of 231.34 mm, the geometric magnification of the sample onto the detector was 3.5. Hence the effective voxel size of the reconstructed volume is 57 µm. The focus-to-object distance was optimised for the highest resolution by given sample and detector sizes. For a given X-ray spot size of less than 10 µm and the geometric magnification of 3.5, the real voxel size is still about 57.5 µm (Rueckel et al., 2014).

The volumetric data were converted into a surface mesh in STL file format by the Calibrate Object tool of the software VGStudio MAX Release 2.0 (Volume Graphics GmbH, Heidelberg, Germany). For the automatic calibration of the surface grey value responsible for thresholding, example areas of background and sample were defined.

2.2. Spline geometries

The geometry of the violin bridge, measured by the μ-CT scan, shall be represented by a three-dimensional multi-patch geometry of 16 spline patches Ω_n, $n = 1, ..., 16$, i.e., $\overline{\Omega} = \bigcup_{n=1}^{16} \overline{\Omega}_n$. The number of 16 patches was chosen based on a reference geometry which incorporates the main topological features. Each spline patch parametrisation is based on the control points $\mathbf{C}_{jkl}^n = (C_{jkl}^{x,n}, C_{jkl}^{y,n}, C_{jkl}^{z,n})^\top \in \mathbb{R}^3$ and is defined as the diffeomorphic image of a spline function $\mathbf{F}_n : \widehat{\Omega} \to \mathbb{R}^3$, $\Omega_n = \mathbf{F}_n(\widehat{\Omega})$ with $\widehat{\Omega} = (0,1)^3$. For notational simplicity, we neglect the patch-index n whenever there is no confusion to be expected.

With the basis $(N_{jkl}^p)_{j,k,l}$ of non-uniform rational B-splines (NURBS) of degree p, the parametrised mapping for each patch is given as

$$\mathbf{F}(\xi, \eta, \theta) = \sum_{j,k,l=1}^{J,K,L} \mathbf{C}_{jkl} N_{jkl}^p(\xi, \eta, \theta).$$

In general, NURBS are rational splines, whose definition involve fixed weights $w_{jkl} > 0$ and a corresponding weight function $W = \sum_{j,k,l} w_{jkl} B_{jkl}^p(\xi, \eta, \theta)$ for a B-spline basis B_{jkl}^p. We then set $N_{jkl}^p = B_{jkl}^p / W$. The multivariate B-spline basis is defined in a tensor-product structure based on univariate splines:

$$B_{jkl}^p(\xi, \eta, \theta) = B_j^p(\xi) B_k^p(\eta) B_l^p(\theta),$$

with one-dimensional spline bases of order p, $B_j^p, B_k^p, B_l^p : (0,1) \to \mathbb{R}$. We briefly explain some properties of a B-spline basis with open non-uniform knot vector $\boldsymbol{\zeta} = (\zeta_1 = 0, ..., \zeta_{p+J+1} = 1)$ of length $p + 1 + J$. Here J is the number of control points which coincides with the dimension of the vector space generated by the basis functions. Open here means that the first and the last values in the knot vector have multiplicity $p + 1$, i.e., $\boldsymbol{\zeta} = (0, ..., 0, \zeta_{p+2}, ..., \zeta_J, 1, ..., 1)$. While the spline functions are in general smooth up to order $\mathcal{C}^{p-1}$, repeating a knot ζ_i, $i = 1, ..., p + J + 1$, in $\boldsymbol{\zeta}$ several times reduces the smoothness of the basis in the corresponding point. By keeping the overall length of $\boldsymbol{\zeta}$ at $p + 1 + J$ the smoothness of the

basis functions stays the same at the other knot values (Cottrell et al., 2009). In particular, if a knot ζ appears p times in the knot vector, the spline basis is interpolatory at the parameter value ζ, the associated basis function has a kink, and the generated geometry may have a corner.

One- and two-dimensional geometries are defined analogously as the images of

$$\mathbf{F}(\xi) = \sum_{j=1}^{J} \mathbf{C}_j B_j^p(\xi)/W(\xi), \qquad\qquad \text{for curves and}$$

$$\mathbf{F}(\xi, \eta) = \sum_{j,k=1}^{J,K} \mathbf{C}_{jk} B_j^p(\xi) B_k^p(\eta)/W(\xi, \eta), \qquad \text{for surfaces.}$$

Due to the tensorial construction, the boundary sides of a spline geometry are again spline geometries of a lower dimension. In this work, the choice of the basis functions is given by an initial sketch of the geometry, but it can in general be chosen flexibly and, if necessary, adaptively. For a detailed presentation of the introduced concepts see Schumaker (2007), Cottrell et al. (2009).

3. Total least squares problem for the data fit

In this chapter we discuss the process to derive a spline geometry based on the unstructured set of data points which were generated by a μ-CT scan.

In our case, we start from a rough initial bridge geometry, which we use to select the patch-decomposition, see Fig. 2. This initial geometry only roughly represents the correct outline and has a trivial shape in the third direction, which makes it too imprecise for the desired vibro-acoustical applications. A NURBS based representation of this draft geometry was first presented in Horger et al. (2017) and is used as a guideline for our more realistic geometric model.

The precise geometry is only given by the output of the μ-CT scan. This is a set of data points

$$\Sigma = \{(u_i, v_i, w_i)^\top\}_{i=1}^{N_{\text{dat}}} \subset \mathbb{R}^3$$

which describes the surface of the geometry of the scanned violin bridge with a high accuracy. To construct a spline geometry for one of the patches from these points, we first have to decide how many control points are required in each of the three parametric directions ξ, η and θ. We denote the number of control points in the three directions by $J, K, L \in \mathbb{N}$ respectively. This yields a total of $M = J \cdot K \cdot L$ control points. For the bridge of the violin, we choose the number of control points of the initial draft geometry.

The control points $\mathbf{C}_{jkl} = (C_{jkl}^x, C_{jkl}^y, C_{jkl}^z)^\top \in \mathbb{R}^3$ of each patch are fitted iteratively in a three step procedure. As a first step we construct the boundary splines of each patch which is explained in detail in the remainder of this chapter. In a second step, we combine the boundary splines to a two-dimensional spline surface with the help of a Coon's patch (Coons, 1967) (see Chapter 4.1). Finally, in the third step, we extend the spline surface to the actual three-dimensional geometry via a constrained quadratic optimisation to fit the z-components of the control points; see Chapter 4.2.

3.1. Preprocessing of the scan data

In a preprocessing step we perform mostly automated cleaning of the measurement data. In particular, we have to remove some outliers which arose from measurement errors. To fully automise this step, one can apply a filtering method based on the local point density to remove outliers as in Schall et al. (2005). Also a neighbourhood based method can be applied like in Wang et al. (2013), for example. Recently even techniques making use of deep learning have been introduced (Rakotosaona et al., 2019). A survey of existing methods to remove outliers from 3D-scan data can be found in Han et al. (2017). Given a STL file which contains a surface triangulation of the measured object, we then use the pre-calculated normal vectors to align the coordinate system such that the flat back of the violin bridge is oriented in the x-y-plane. Afterwards we orthogonally project all points into a plane and triangulate them using a Delaunay triangulation. This step is possible due to the geometric structure of the bridge, which can be characterised by the two-dimensional flat back and non-linear values of the thickness measured in relation to the flat back. Mapping all points into the plane results in more data points than can be used to fit the outline of the two-dimensional back. Using the planar triangulation, we can determine the edges which are adjacent to only one triangle. These edges form the boundary of the flat back and the vertices belonging to these edges are collected as the desired set of boundary vertices. An easy and elegant way to find boundary edges and vertices is the MATLAB function `freeBoundary`, which makes use of the connectivity list of the triangulation. This results in a set of two-dimensional data points $\{(u_i, v_i)\}_{i=1}^{I}$, which describe the outline of the back of the violin. These data points are then manually distributed among the boundary curves of the 16 patches. For each boundary curve, the data points are sorted automatically, either according to their x-coordinate, their y-coordinate or the angular variable around the centre of mass of the data points. The assignment of these data points to the 16 patches of the future spline geometry is displayed in Fig. 3. If no patch decomposition is given a-priori, various approaches are available to obtain meaningful domain segmentations, e.g., Buchegger and Jüttler (2017), Xiao et al. (2018) and Falini and Jüttler (2019).
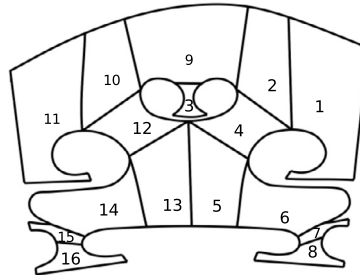
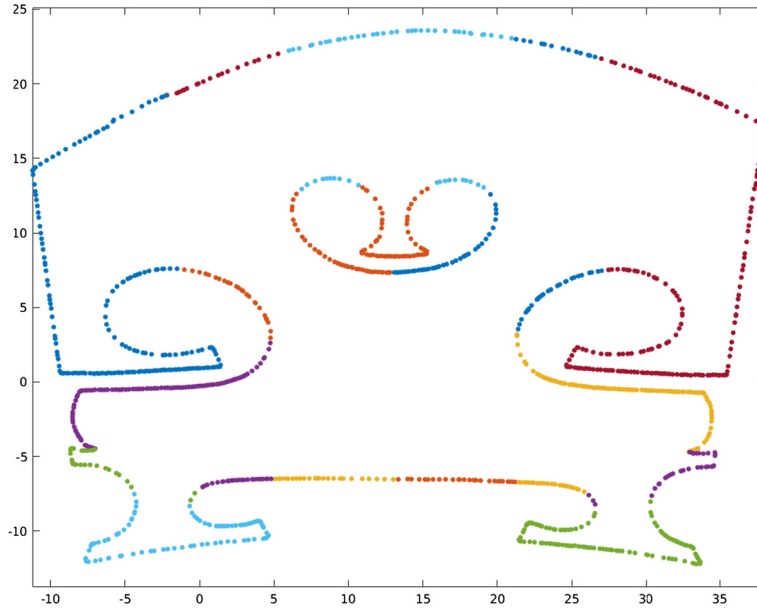**Fig. 2.** Patch decomposition of the reference geometry.

Fig. 3. Assignment of the data points to the 16 patches of the computational geometry indicated by different colours. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

3.2. Fitting the boundary splines

In this chapter, we describe one of the key procedures, the fitting of the (one-dimensional) boundary splines. Adopting the idea proposed in Borges and Pastva (2002) and incorporating additional constraints, we obtain a Gauss-Newton approach to minimise the total least squares error between ordered data and a B-spline curve. This results in a precise fit of complicated curved shapes including sharp edges. Within the presented procedure, one only has to solve a single minimisation problem with respect to a pair of optimisation variables, the control points and the parameter values of the data points.

Given a set of *ordered* data points $\{(u_i, v_i)^\top\}_{i=1}^I$ and an univariate NURBS basis $\{N_j^p\}_{j=1}^J$ of degree p (in our case $p = 3$), we want to find I parametric points $\{\xi_i\}_{i=1}^I$ with $0 \leq \xi_1 < \xi_2 < \cdots < \xi_I \leq 1$ and J control points $\{(C_j^x, C_j^y)^\top\}_{j=1}^J$ that minimise the quadratic distance between the curve evaluated at the parametric points and the data:

$$\sum_{i=1}^I \left\| \begin{pmatrix} u_i \\ v_i \end{pmatrix} - \sum_{j=1}^J N_j^p(\xi_i) \cdot \begin{pmatrix} C_j^x \\ C_j^y \end{pmatrix} \right\|_2^2 \longrightarrow \min_{(\xi_i)_i, (C_j^x)_j, (C_j^y)_j} . \tag{1}$$

The data points are ordered in the parametric direction of the spline to be constructed. In the following, we neglect the fixed degree p and write N_j for elements of the spline basis.

We now briefly recapitulate the method and then extend it by incorporating several constraints. For further details, we refer to the original article (Borges and Pastva, 2002).

Introducing the *NURBS basis matrix* which contains all J NURBS basis functions N_j evaluated at a vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_I)^\top$ of parametric points

$$\mathcal{N}(\boldsymbol{\xi}) = \begin{pmatrix} N_1(\xi_1) & \cdots & N_J(\xi_1) \\ \vdots & \ddots & \vdots \\ N_1(\xi_I) & \cdots & N_J(\xi_I) \end{pmatrix},$$

we can reformulate (1) in a matrix version:

$$\left\| \mathbf{u} - \mathcal{N}(\boldsymbol{\xi}) \mathbf{C}^x \right\|_2^2 + \left\| \mathbf{v} - \mathcal{N}(\boldsymbol{\xi}) \mathbf{C}^y \right\|_2^2 \longrightarrow \min_{\boldsymbol{\xi}, \mathbf{C}^x, \mathbf{C}^y} .$$

Here $\mathbf{u}$, $\mathbf{v}$, $\mathbf{C}^x$, $\mathbf{C}^y$ are vectors which contain the elements u_i, v_i and C_j^x, C_j^y respectively. In the following we omit the $\boldsymbol{\xi}$ dependency in the notation when it is not in the foreground and simply write $\mathcal{N}$ instead of $\mathcal{N}(\boldsymbol{\xi})$. We consider the over-determined case where more data points than degrees of freedom are present, i.e., $I > J$, and assume that $\mathcal{N}$ has full rank.

For a fixed parameter vector $\boldsymbol{\xi}$, the optimisation for the control points reduces to a quadratic least-squares problem. This allows us to locally eliminate the control points. Then (1) reads

$$\|\mathbf{r}(\boldsymbol{\xi})\|^2 \longrightarrow \min_{\boldsymbol{\xi}}, \quad \text{where } \mathbf{r}(\boldsymbol{\xi}) = \begin{pmatrix} \mathbf{u} - \mathcal{N}(\boldsymbol{\xi}) \mathcal{N}^+(\boldsymbol{\xi}) \mathbf{u} \\ \mathbf{v} - \mathcal{N}(\boldsymbol{\xi}) \mathcal{N}^+(\boldsymbol{\xi}) \mathbf{v} \end{pmatrix}. \tag{2}$$

Here $\mathcal{N}^+$ denotes the Moore Penrose inverse $\mathcal{N}^+ = (\mathcal{N}^\top \mathcal{N})^{-1} \mathcal{N}^\top$ of the matrix $\mathcal{N}$; see Ben-Israel and Greville (2003). This formulation is independent of the control points $(C_j^x, C_j^y)^\top$ and only needs to be solved for the optimal parameter values $\{\xi_i\}_{i=1}^I$.

To iteratively solve the minimisation problem (2), we apply a Gauss-Newton scheme with the iteration steps

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k - J_{\mathbf{r}}^+(\boldsymbol{\xi}_k) \mathbf{r}(\boldsymbol{\xi}_k) .$$

Again, $J_{\mathbf{r}}^+(\boldsymbol{\xi}_k)$ denotes the Moore Penrose inverse of the Jacobi matrix $J_{\mathbf{r}}(\boldsymbol{\xi}_k)$ for the parameter vector $\boldsymbol{\xi}_k$ at iteration step k.

$J_{\mathbf{r}}(\boldsymbol{\xi}_k)$ is the Jacobi matrix of $\mathbf{r}(\boldsymbol{\xi})$, so its evaluation is not trivial. In Borges and Pastva (2002, Theorem 1) an elegant technique is explained to differentiate a generalised inverse, which we state very briefly. For further details, we refer to the original article, where they show

$$J_{\mathbf{r}} = \begin{pmatrix} \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_1}(\mathbf{u}), \ldots, \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_I}(\mathbf{u}) \\ \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_1}(\mathbf{v}), \ldots, \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_I}(\mathbf{v}) \end{pmatrix}$$

with

$$\frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_j} = - Q_2 Q_2^\top \frac{\partial \mathcal{N}}{\partial \xi_j} \Pi R^{-1} Q_1^\top - \left(Q_2 Q_2^\top \frac{\partial \mathcal{N}}{\partial \xi_j} \Pi R^{-1} Q_1^\top \right)^\top$$

and

$$P_{\mathrm{rg}(\mathcal{N})} = Q_1 Q_1^\top , \qquad P_{\mathrm{rg}(\mathcal{N})^\perp} = Q_2 Q_2^\top .$$

The permutation matrix Π, the orthogonal matrix $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \in \mathbb{R}^{I \times I}$ with $Q_1 \in \mathbb{R}^{I \times J}$, $Q_2 \in \mathbb{R}^{I \times (I-J)}$ and the upper triangular matrix $R \in \mathbb{R}^{J \times J}$ result from the QR-decomposition of the matrix $\mathcal{N}$.

With this on hand, the Gauss-Newton scheme can be performed. Note that it is reasonable to include a function checking and safe-guarding that the parameter vector $\boldsymbol{\xi}$ is a monotone-increasing sequence, preventing that $\mathcal{N}(\boldsymbol{\xi})$ becomes rank deficient.

3.3. Regularising the solution

The procedure described in the previous section shows good results when fitting curves with no major changes in concavity, but once corners or sharp bends are present, twisting of control points occurs. In the following sections, we present some techniques to improve the geometric quality of the data fit. These methods comprise several regularisation terms which can be added to the data fit criterion (1). This is especially helpful in the case of $\mathcal{N}(\boldsymbol{\xi})$ being nearly rank deficient, because this leads to a numerically instable optimisation system. The effect of these regularisation terms are controlled by parameters which have to be tuned to suppress undesirable features on the one hand, without causing numerical instabilities on the other hand.

3.3.1. Interpolating characteristic points

Within the given data points, we can identify characteristic points which should be interpolated precisely. In particular, this includes the start and end points of the boundary spline as well as corners that are present in the geometry. The interpolation of characteristic features of the geometry is important to guarantee that these features are present in the spline model. Therefore, we extend the optimisation problem (1) and introduce the interpolation condition as a constraint. We first present a simple way to interpolate end as well as corner points, based on the elimination of degrees of freedom from the equation system which are fixed by the constraint conditions. Afterwards, we present a more general approach which allows for the interpolation of arbitrary points.

Eliminating corner and end points from the optimization system We first consider the case that a set of corner points $\{(u_i, v_i)\}_{i \in \mathcal{I}_\lambda} \subsetneq \{u_i, v_i\}_{i=1}^I$ shall be interpolated exactly. These points are identified by a subset of indices $\mathcal{I}_\lambda \subsetneq \{1, \dots, I\}$ whose complement we denote by $\mathcal{I}_\lambda^c = \{1, \dots, I\} \setminus \mathcal{I}_\lambda$. A useful property of B-splines is that when a knot ζ_l is repeated p times in the knot vector, the spline interpolates the corresponding control point $(C_j^x, C_j^y)^\top$; see Section 2.2. In this case we have $N_j(\zeta_l) = \delta_{j,l}$, and setting $(C_j^x, C_j^y)^\top = (u_i, v_i)^\top$ for some $i \in \mathcal{I}_\lambda$, the spline curve passes exactly through this data point at the parameter points $\xi_i = \zeta_l$. The same approach works for the start and end points as we consider open knot vectors. Effectively, this means that the curve decomposes into subcurves between two corner points or between a corner and an endpoint. Since the control points and the interpolating parameter values ξ_i for $i \in \mathcal{I}_\lambda$ are fixed they can be eliminated from the minimisation problem (2) as follows. We define $\mathcal{J}_\lambda$ as the subset of all indices $\{1, \dots, J\}$ which correspond to B-splines interpolating the data point $(u_i, v_i)^\top$ for $i \in \mathcal{I}_\lambda$. Moreover, we set $\mathcal{J}_\lambda^c = \{1, \dots, J\} \setminus \mathcal{J}_\lambda$. Note that the described correspondence between interpolated data points $(u_i, v_i)^\top$, $i \in \mathcal{I}_\lambda$, and interpolated control points $\mathbf{C}_j$ with $j \in \mathcal{J}_\lambda$ is obviously one-to-one. We denote this correspondence for $\hat{j} \in \mathcal{J}_\lambda$ and associated $\hat{i} \in \mathcal{I}_\lambda$ by $\hat{i} = \hat{i}(\hat{j})$ and $\hat{j} = \hat{j}(\hat{i})$ respectively such that $N_{\hat{j}}(\xi_i) = \delta_{\hat{i}\hat{j}}$.

The least-squares problem only needs to be solved for the variables C_j^x and C_j^y for all $j \in \mathcal{J}_\lambda^c$, and we thus can eliminate the remaining control points:

$$\sum_{i \in \mathcal{I}_\lambda^c} \left| u_i - \sum_{j \in \mathcal{J}_\lambda^c} C_j^x N_j(\xi_i) - \sum_{\hat{j} \in \mathcal{J}_\lambda} u_{\hat{i}(\hat{j})} N_{\hat{j}}(\xi_i) \right|^2$$

$$+ \sum_{i \in \mathcal{I}_\lambda^c} \left| v_i - \sum_{j \in \mathcal{J}_\lambda^c} C_j^y N_j(\xi_i) - \sum_{\hat{j} \in \mathcal{J}_\lambda} v_{\hat{i}(\hat{j})} N_{\hat{j}}(\xi_i) \right|^2 \longrightarrow \min_{C_j^x, C_j^y \text{ for } j \in \mathcal{J}_\lambda^c}.$$

We introduce the reduced B-spline basis matrix $\tilde{\mathcal{N}}_\lambda^c \in \mathbb{R}^{|\mathcal{I}_\lambda^c| \times |\mathcal{J}_\lambda^c|}$ as

$$\left(\tilde{\mathcal{N}}_\lambda^c \right)_{ij} = N_j(\xi_i), \text{ for } i \in \mathcal{I}_\lambda^c, j \in \mathcal{J}_\lambda^c,$$

and the reduced data vectors

$$\tilde{\mathbf{u}} = \left(u_i - \sum_{\hat{j} \in \mathcal{J}_\lambda} u_{\hat{i}(\hat{j})} N_{\hat{j}}(\xi_i) \right)_{i \in \mathcal{I}_\lambda^c}.$$

The vector $\tilde{\mathbf{v}} \in \mathbb{R}^{|\lambda^c|}$ is defined analogously. Together with the additional notation $\mathbf{C}_\lambda^x = (C_j^x)_{j \in \mathcal{J}_\lambda^c}$ and $\mathbf{C}_\lambda^y = (C_j^y)_{j \in \mathcal{J}_\lambda^c}$ for the control points, the resulting optimality system reads

$$\langle \tilde{\mathbf{u}} - \tilde{\mathcal{N}}_\lambda^c \mathbf{C}_\lambda^x, \tilde{\mathcal{N}}_\lambda^c \boldsymbol{\tau} \rangle_2 = 0 \text{ and}$$
$$\langle \tilde{\mathbf{v}} - \tilde{\mathcal{N}}_\lambda^c \mathbf{C}_\lambda^y, \tilde{\mathcal{N}}_\lambda^c \boldsymbol{\tau} \rangle_2 = 0 \text{ for all } \boldsymbol{\tau} \in \mathbb{R}^{|\mathcal{J}_\lambda^c|}.$$

Therefore, the structure of the optimality system and also the subsequent optimisation with respect to the parameter values $(\xi_i)_{i \in \mathcal{I}_\lambda^c}$ is the same as in the previous subsection. By the modification the system equations decouple into a set of independent subproblems. Since the remaining optimisation parameters $(\xi_i)_{i \in \mathcal{I}_\lambda^c}$ usually do not tend to the interval boundaries (especially not when the penalties presented in Subsection 3.3.2 and 3.3.3 are in force) we do not get into the quandary of $\tilde{\mathcal{N}}_\lambda^c(\boldsymbol{\xi})$ being almost singular.

Constrained optimization for general characteristic points The resulting reduced regularity of the presented method is reasonable for corner points, but might be undesirable for arbitrary characteristic points. For this case we present a second, more general method to incorporate the interpolation of data points. If the data point to be interpolated is not a corner of the geometry, the spline basis will not have the interpolatory property that only one basis function is different from zero at the considered parameter value. The elimination of certain control points from the optimisation in this case is not that obvious. Instead of reducing the number of optimisation variables, we treat the interpolation conditions as equality constraints during the optimisation. The interpolation of a data point $(u_i, v_i)^\top$ at an arbitrary parameter value $\xi = \xi_i$ for $i \in \mathcal{I}_\lambda$ is written as

$$\sum_{j=1}^{J} C_j^x N_j(\xi_i) = u_i \text{ and } \sum_{j=1}^{J} C_j^y N_j(\xi_i) = v_i$$

for all $i \in \mathcal{I}_\lambda$. As in the previous section, we treat the parameter values ξ_i at which we want to interpolate exactly as fixed parameters and exclude them from the list of optimisation variables. The optimisation is only carried out with respect to the parameter values ξ_i with $i \in \mathcal{I}_\lambda^c$. Accordingly, we introduce the two new B-spline basis matrices

$$\left(\mathcal{N}_\lambda^c\right)_{ij} = N_j(\xi_i), \quad i \in \mathcal{I}_\lambda^c, \ j \in \{1, \ldots, J\},$$
$$(\mathcal{N}_\lambda)_{ij} = N_j(\xi_i), \quad i \in \mathcal{I}_\lambda, \ j \in \{1, \ldots, J\},$$

taking into account the classification of basis functions induced by free and fixed parameter values. Additionally we denote the selected data vectors by $\mathbf{u}_\lambda^c = (u_i)_{i \in \mathcal{I}_\lambda^c}$, $\mathbf{u}_\lambda = (u_i)_{i \in \mathcal{I}_\lambda}$ and analogously for $\mathbf{v}$. We assume to have only a few characteristic points, such that $|\mathcal{I}_\lambda| < J$ and $\mathcal{N}_\lambda$ has full rank. Note that $\mathcal{N}_\lambda^c$ also has full rank, as its columns are linearly independent.

With this on hand, we can write the constrained optimisation problem for fixed parameter vector $\boldsymbol{\xi}$ as

$$\|\mathbf{u}_\lambda^c - \mathcal{N}_\lambda^c \mathbf{C}^x\|_2^2 + \|\mathbf{v}_\lambda^c - \mathcal{N}_\lambda^c \mathbf{C}^y\|_2^2 \longrightarrow \min_{\mathbf{C}^x, \mathbf{C}^y} \tag{3a}$$

$$\text{s.t. } \left((\mathcal{N}_\lambda \mathbf{C}^x)^\top, (\mathcal{N}_\lambda \mathbf{C}^y)^\top\right)^\top = (\mathbf{u}_\lambda^\top, \mathbf{v}_\lambda^\top)^\top. \tag{3b}$$

Problem (3) is a quadratic optimisation problem with linear constraints. As the terms concerning $\mathbf{C}^x$ and $\mathbf{C}^y$ are not coupled, we can consider them independently. We present the elimination of $\mathbf{C}^x$ and note that $\mathbf{C}^y$ meet the analogous equations. The corresponding optimality system is given by

$$\begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathcal{N}_\lambda^c & (\mathcal{N}_\lambda)^\top \\ \mathcal{N}_\lambda & 0 \end{pmatrix} \begin{pmatrix} \mathbf{C}^x \\ \boldsymbol{\lambda}^x \end{pmatrix} = \begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathbf{u}_\lambda^c \\ \mathbf{u}_\lambda \end{pmatrix}. \tag{4}$$

We denote the upper left block of the saddle point equation by

$$A = (\mathcal{N}_\lambda^c)^\top \mathcal{N}_\lambda^c.$$

Note that under the previous assumption that $\mathcal{N}(\boldsymbol{\xi})$ has maximal rank, $\mathcal{N}_\lambda^c(\boldsymbol{\xi})$ will also have maximal rank, and hence A will be regular. With $\mathcal{N}_\lambda$ being surjective, the saddle point matrix of (4) is regular; see Boffi et al. (2013, Theorem 3.2.1). We eliminate the control points using the Schur complement $S = -\mathcal{N}_\lambda A^{-1} (\mathcal{N}_\lambda)^\top$:

$$\mathbf{C}^x = \left(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathcal{N}_\lambda\right) A^{-1} (\mathcal{N}_\lambda^c)^\top \mathbf{u}_\lambda^c - A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathbf{u}_\lambda, \tag{5}$$

where $I_J \in \mathbb{R}^{J \times J}$ denotes the identity matrix. Analogous to (2) this yields the reduced optimisation problem for the parameter vector $\boldsymbol{\xi} = (\xi_i)_{i \in \mathcal{I}_\lambda^c}$:

$$\|\boldsymbol{r}_\lambda(\boldsymbol{\xi})\| = \|\mathbf{r}_\lambda^x(\boldsymbol{\xi})\|_2^2 + \|\mathbf{r}_\lambda^y(\boldsymbol{\xi})\|_2^2 \to \min_{\boldsymbol{\xi}},$$

where

$$\mathbf{r}_\lambda^x(\boldsymbol{\xi}) = \left(I_{|\mathcal{I}_\lambda^c|} - \mathcal{N}_\lambda^c \left(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathcal{N}_\lambda\right) A^{-1}(\mathcal{N}_\lambda^c)^\top\right) \mathbf{u}_\lambda^c$$
$$- \mathcal{N}_\lambda^c A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathbf{u}_\lambda \tag{6}$$

and $\mathbf{r}_\lambda^y(\boldsymbol{\xi})$ is defined analogously.

The matrices $\mathcal{N}_\lambda^c$ and $\mathcal{N}_\lambda$ and consequently also A and S depend on the parameter vector $\boldsymbol{\xi}$ and therefore contribute to the Jacobian $\mathbf{J}_{\mathbf{r}_\lambda}$. To set up the Gauss-Newton scheme, we further explore the structure of (5) and show that (Borges and Pastva, 2002, Theorem 1) can be applied to calculate the Jacobian of certain terms occurring in the residual term (6).

Lemma 1. *The linear map*

$$P_V = \mathcal{N}_\lambda^c \left(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathcal{N}_\lambda\right) A^{-1} (\mathcal{N}_\lambda^c)^\top$$

is the orthogonal projection in $\mathbb{R}^{|\mathcal{I}_\lambda^c|}$ onto the subspace

$$V = \{\mathbf{v} = \mathcal{N}_\lambda^c \mathbf{u} \in \text{rg}(\mathcal{N}_\lambda^c) : \mathcal{N}_\lambda \mathbf{u} = 0\}$$
$$= \text{rg}\left(\mathcal{N}_\lambda^c|_{\ker(\mathcal{N}_\lambda)}\right).$$

Proof. First we rewrite the operator in terms of two Moore Penrose inverses (denoted by the superscript $^+$)

$$\mathcal{N}_\lambda^{\mathsf{c}}\big(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1}\mathcal{N}_\lambda\big)A^{-1}(\mathcal{N}_\lambda^{\mathsf{c}})^\top = \tag{7}$$
$$\mathcal{N}_\lambda^{\mathsf{c}}(\mathcal{N}_\lambda^{\mathsf{c}})^+ - \mathcal{N}_\lambda^{\mathsf{c}}A^{-1}(\mathcal{N}_\lambda)^\top\big(\mathcal{N}_\lambda^{\mathsf{c}}A^{-1}(\mathcal{N}_\lambda)^\top\big)^+ .$$

Now a direct calculation shows that P_V is idempotent, i.e., $P_V^2 = P_V$, and symmetric. It therefore is the orthogonal projection in $\mathbb{R}^{|\mathcal{I}_\lambda^{\mathsf{c}}|}$ onto its range $\mathrm{rg}(P_V)$ and it remains to show that $\mathrm{rg}(P_V) = V$.

Obviously $\mathrm{rg}(P_V) \subset \mathrm{rg}(\mathcal{N}_\lambda^{\mathsf{c}})$ holds. Moreover, for $\mathbf{v} = P_V\mathbf{w}$ and

$$\mathbf{u} = (\mathcal{N}_\lambda^{\mathsf{c}})^+\mathbf{w} - A^{-1}(\mathcal{N}_\lambda)^\top\big(\mathcal{N}_\lambda^{\mathsf{c}}A^{-1}(\mathcal{N}_\lambda)^\top\big)^+\mathbf{w},$$

we have $\mathbf{v} = \mathcal{N}_\lambda^{\mathsf{c}}\mathbf{u} \in \mathrm{rg}(P_V)$ and $\mathcal{N}_\lambda\mathbf{u} = 0$, which shows $\mathrm{rg}(P_V) \subset V$.

On the other hand let $\mathbf{v} \in V$. From $\mathbf{v} = \mathcal{N}_\lambda^{\mathsf{c}}\mathbf{u}$ and $\mathcal{N}_\lambda\mathbf{u} = 0$, we get $\mathcal{N}_\lambda^{\mathsf{c}}(\mathcal{N}_\lambda^{\mathsf{c}})^+\mathbf{v} = \mathbf{v}$ by the properties of the Moore Penrose inverse. Furthermore, we have

$$\mathcal{N}_\lambda^{\mathsf{c}}A^{-1}\mathcal{N}_\lambda^\top(\mathcal{N}_\lambda A^{-1}\mathcal{N}_\lambda)^{-1}\mathcal{N}_\lambda A^{-1}(\mathcal{N}_\lambda^{\mathsf{c}})^\top\mathcal{N}_\lambda^{\mathsf{c}}\mathbf{u} = 0,$$

because $A^{-1}(\mathcal{N}_\lambda^{\mathsf{c}})^\top\mathcal{N}_\lambda^{\mathsf{c}} = I$ and $\mathcal{N}_\lambda\mathbf{u} = 0$. From this it follows that $P_V\mathbf{v} = \mathbf{v}$ and consequently $V \subset \mathrm{rg}(P_V)$. □

We now consider the subspace $\ker(\mathcal{N}_\lambda) \subset \mathbb{R}^J$ since we assume that $\mathcal{N}_\lambda$ has maximal rank $|\mathcal{I}_\lambda|$ and that $|\mathcal{I}_\lambda| < J$. The orthogonal projection $P_{\ker(\mathcal{N}_\lambda)}$ onto $\ker(\mathcal{N}_\lambda)$ in $\mathbb{R}^J$ is then given by

$$P_{\ker(\mathcal{N}_\lambda)} = I - (\mathcal{N}_\lambda)^+\mathcal{N}_\lambda .$$

See Ben-Israel and Greville (2003). To treat the projection P_V onto V within the framework described in Borges and Pastva (2002, Theorem 1) we define the matrix B,

$$B = \mathcal{N}_\lambda^{\mathsf{c}}\big(I - \mathcal{N}_\lambda^+\mathcal{N}_\lambda\big) \in \mathbb{R}^{|\mathcal{I}_\lambda^{\mathsf{c}}|\times J}, \tag{8}$$

and we obtain

$$V = \mathrm{rg}(\mathcal{N}_\lambda^{\mathsf{c}}|_{\ker(\mathcal{N}_\lambda)}) = \mathrm{rg}(\mathcal{N}_\lambda^{\mathsf{c}} \cdot P_{\ker(\mathcal{N}_\lambda)}) = \mathrm{rg}\big(\mathcal{N}_\lambda^{\mathsf{c}}\big(I - (\mathcal{N}_\lambda)^+\mathcal{N}_\lambda\big)\big) = \mathrm{rg}(B) .$$

We set

$$B^- = (\mathcal{N}_\lambda^{\mathsf{c}})^+ - A^{-1}\mathcal{N}_\lambda^\top S^{-1}\mathcal{N}_\lambda(\mathcal{N}_\lambda^{\mathsf{c}})^+, \tag{9}$$

which is a generalised inverse of B.

For the matrices B and B^-, defined in (8) and (9), the conditions of Borges and Pastva (2002, Theorem 1) are satisfied. Applying the theorem to $P_{\mathrm{rg}(B)^\perp} = (I - BB^-)$ yields

$$\frac{\partial P_{(\mathrm{rg}B)^\perp}}{\partial \xi_i} = -P_{(\mathrm{rg}B)^\perp}\frac{\partial B}{\partial \xi_i}B^- - \Big(P_{(\mathrm{rg}B)^\perp}\frac{\partial B}{\partial \xi_i}B^-\Big)^\top . \tag{10}$$

Note that $I - BB^-$ is exactly the matrix in the first expression on the right hand side of (6). To calculate the Jacobian for the Gauss-Newton scheme, the first term in the residual vector $\mathbf{r}_\lambda(\boldsymbol{\xi})$ can be treated exactly as described in Section 3.2. Note also that the parameter values ξ_i for $i \in \mathcal{I}_\lambda$ do not appear as optimisation variables and therefore their derivatives are not considered in the computation of the Jacobian. For $i \in \mathcal{I}_\lambda^{\mathsf{c}}$, we get $\frac{\partial}{\partial \xi_i}\mathcal{N}_\lambda = 0$ for the derivative because $\mathcal{N}_\lambda$ does only depend on the interpolating parameter values in $\mathcal{I}_\lambda$. Therefore, we get

$$\frac{\partial B}{\partial \xi_i} = \frac{\partial \mathcal{N}_\lambda^{\mathsf{c}}}{\partial \xi_i}(I - \mathcal{N}_\lambda^+\mathcal{N}_\lambda) \tag{11}$$

for all $i \in \mathcal{I}_\lambda^{\mathsf{c}}$.

The term remaining in equation (6) is $\mathcal{N}_\lambda^{\mathsf{c}}A^{-1}\mathcal{N}_\lambda^\top S^{-1}$, whose partial derivatives can be computed using standard differentiation rules.

Example of a constrained fit The significance of precisely interpolating corners as well as end points is visualised in Fig. 4. It shows how a patch boundary with two kinks is fitted with and without the additional constraint. In the left picture in Fig. 4 we note instabilities in the fitted spline since the Jacobian is ill-conditioned around the point where the curve is non-smooth. We can clearly recognise an improved fit of the corners once the constraint is used, resulting in a better overall fit.
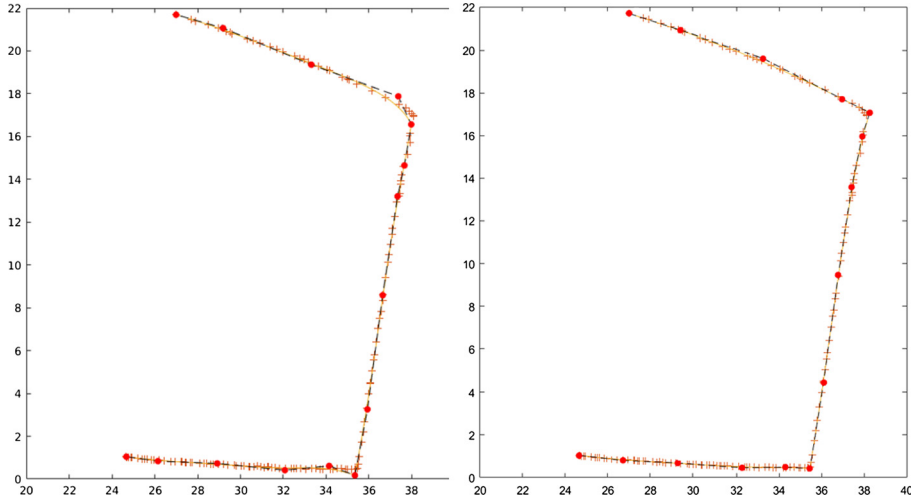
Fig. 4. Left: spline fit without constraints. Right: spline fit with end and corner point constraints. The red bullets indicate the control points of the fitted spline (which is visualised by a yellow line). The data points to be approximated are indicated by red crosses. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

3.3.2. Control point distances penalty

A major problem occurring in curve fitting is that sometimes control points tend to oscillate or accumulate. Hence $\mathcal{N}(\boldsymbol{\xi})$ might become nearly rank deficient. This can be avoided by penalising the distance between adjacent control points relative to the curve length. So, setting

$$L = \sum_{i=1}^{I} \left\| \begin{pmatrix} u_i \\ v_i \end{pmatrix} - \begin{pmatrix} u_{i-1} \\ v_{i-1} \end{pmatrix} \right\|_2^2,$$

and denoting the control point distances by $\boldsymbol{\Delta}_j = \mathbf{C}_j - \mathbf{C}_{j-1}$, $j = 2, ..., J$ and fixing desirable distances $\overline{\boldsymbol{\Delta}}_j$, we can define the regularisation term

$$p_1\left(\mathbf{C}^x, \mathbf{C}^y, \alpha\right) = \frac{\alpha}{L} \sum_{j=2}^{J} \|\boldsymbol{\Delta}_j - \overline{\boldsymbol{\Delta}}_j\|^2. \tag{12}$$

Here, once again, one possibly can make use of a similar, pre-calculated NURBS geometry with control points $\{\mathbf{C}_j^{\text{ref}}\}_{j=1}^{J}$ by setting $\overline{\boldsymbol{\Delta}}_j = \mathbf{C}_j^{\text{ref}} - \mathbf{C}_{j-1}^{\text{ref}}$. Otherwise one needs to decide upon an eligible spacing, the simplest one – though not always effective and applicable – would be a uniform distribution. Usually α is in the range $[0, 10]$.

Fig. 5 illustrates the relevance of this regulariser. For this strongly curved outline we get oscillating control points due to overfitting. The regularisation term p_1 can prevent this undesirable behaviour in this example and yields a curve that matches the exact outline.

3.3.3. Control point miss-match penalty

The regularisation term $p_1(\mathbf{C}^x, \mathbf{C}^y, \alpha)$ reduces overfitting, but in some cases the quality of the approximation is insufficient under the influence of p_1 alone. For these cases, we introduce a second regularisation term since we observed that the joint use of two regularisation terms with smaller parameters results in an improved fit. An example is shown in Figs. 6a and 6b, where the original data fit without any additional penalties yields a self-intersecting curve. Applying the regularisation $p_1(\mathbf{C}^x, \mathbf{C}^y, \alpha)$ results in a curve which is smooth but with a significant los in the accuracy of the approximation.

The second regularisation term forces the control points towards a suitable set of reference control points. This also prevents the self-intersection of the curve and allows us to use a smaller regularisation constant, yielding a better approximation. The reference control points for this can either be chosen, or, as in this case the control points of the available reference geometry can be used: $\mathbf{C}_j^{\text{ref}} = (C_j^{\text{ref},x}, C_j^{\text{ref},y})^\top$, $j \in \{1, ..., J\}$. The new regulariser then is of the following form:

$$p_2\left(\mathbf{C}^x, \mathbf{C}^y, \beta\right) = \beta \sum_{j=1}^{J} \left\| \mathbf{C}_j^{\text{ref}} - \mathbf{C}_j \right\|_2^2. \tag{13}$$

In our case the regularisation factor $\beta > 0$ was usually in the range $[0, 0.1]$ but it should be remarked that β needs to be chosen with special care since the similar geometry may have slight, but notable differences in curvature and dimension or size. The improved precision of the geometry with both regularisation terms can be seen in Fig. 6c.
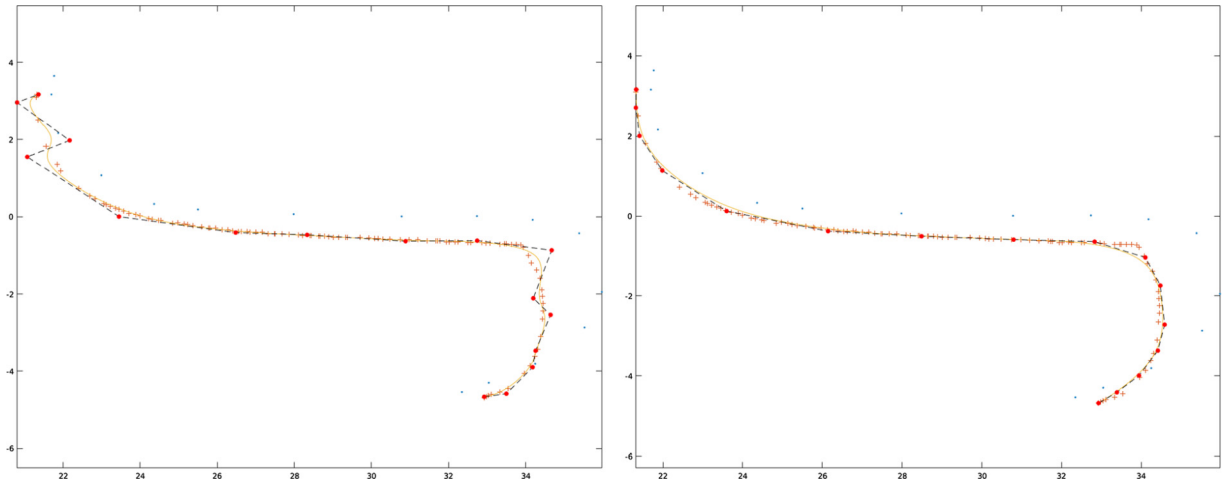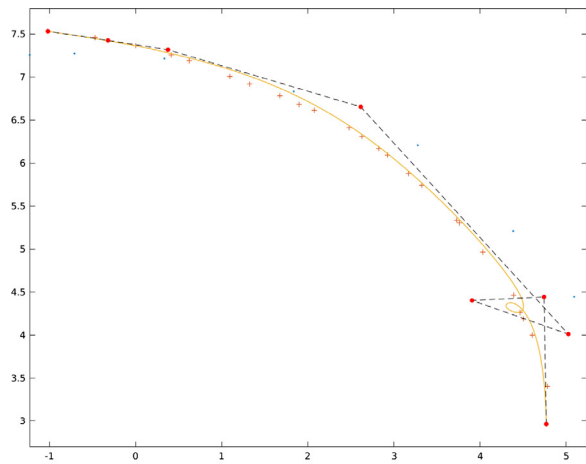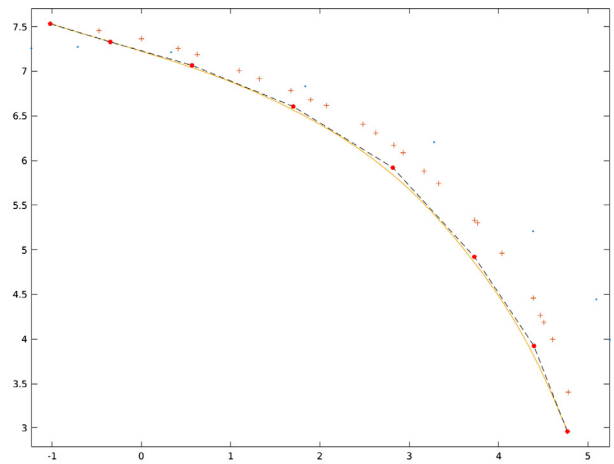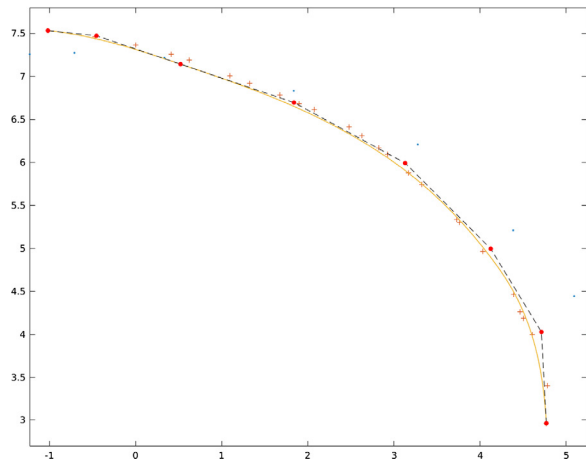
Fig. 5. Left: spline fit without penalising the distance of the control points. Right: spline fit with penalizing the distance of the control points via the term $p_1(\mathbf{C}^x, \mathbf{C}^y, \alpha)$ in (12) with $\alpha = 4$. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

(a) Spline-fit with no regularisation
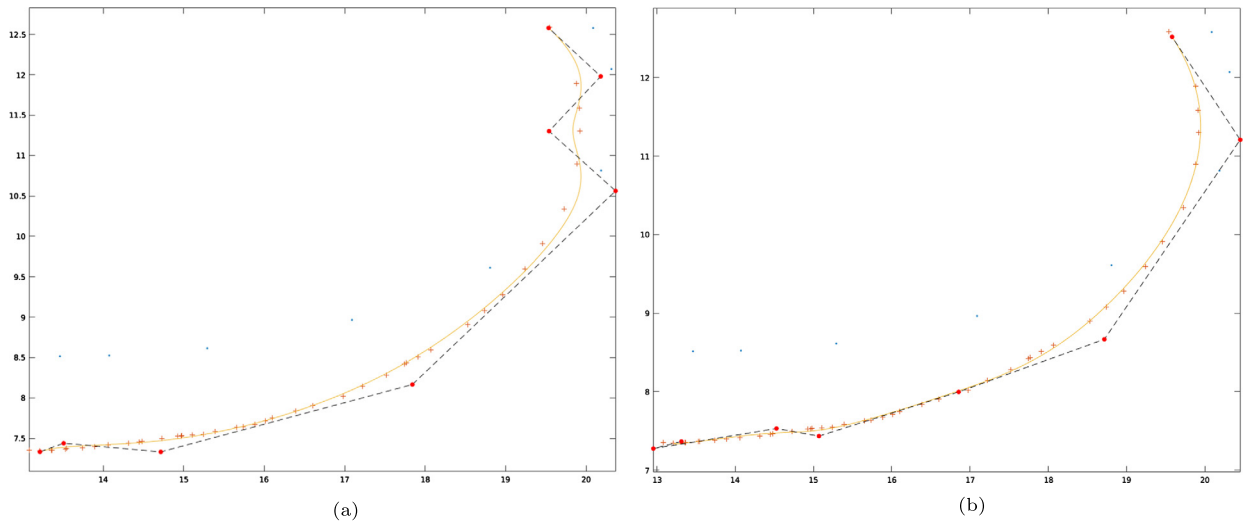
(b) Spline fit, regularised using only p_1 with $\alpha = 10$.

(c) Spline fit, regularised using both terms, p_1 and p_2, with $\alpha = 7.5, \beta = 0.1$.

Fig. 6. Comparison of the fitted spline curves with different regularisation terms applied. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

Fig. 7. Illustration of the advantage of optimising the parameter vector. Figure (a) shows the spline fit without optimising the parameter vector and (b) with optimising the parameter vector only at the first iteration which significantly smoothness the spline curve. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

Fig. 6 shows an example where the error of the data fit is significantly reduced by applying this regulariser such that the control points are forced into the direction of the data points.

In summary we suggest to minimise the modified least squares problem which we get by adding the terms p_1, p_2 to equation (3a). In the same manner as before, inserting the solution for the control points which we get for fixed ξ then again leads to a (now modified) residual term $\tilde{r}_\lambda$. The optimality system stays nearly the same, except for some obvious modifications in (4). Specifically a unit matrix is used for the term $p_1\left(C^x, C^y, \alpha\right)$, a unit block diagonal matrix needs to be inserted for the term $p_2\left(C^x, C^y, \beta\right)$ and the corresponding necessary changes in the Jacobian matrix $J_{\tilde{r}_\lambda}(\xi)$ (insertion of additional zero lines) have to be made.

We compared the behaviour of the optimisation algorithm for different start configurations of the parameter vector. More precisely, we compared starting with a uniform parameter vector against starting with a parameter vector parameterised according to the affine angle parameterisation (Foley and Nielson, 1989). Both versions needed a comparable number of iterations to converge to a steady state. Between 10 to 15 iterations per spline fit were needed, depending on the complexity of the geometry of the underlying data. With this approach we did not experience any case of non-convergence. Fig. 7 illustrates the benefit of the optimisation of the parameter vector ξ. It is clearly visible that the fitted spline curve tends to oscillate if we only optimise the control points and not the parameter vector; see Figs. 7 (a) and (b). Even though an optimisation of ξ only in the first iteration step of the optimisation can significantly improve the data fit, there are still some obstructive kinks present in the control net.

We conclude with Fig. 8, where one can see the positive influence of the regularisation terms on the accuracy of the spline fit for one of the boundary splines. The estimated error is plotted logarithmically.

4. Construction of the fitted NURBS volume

With the fit of univariate splines to the outline of the back of the violin as described in Chapter 3, the main step to a precise geometry description is completed. However, this does not yet capture all of the important geometry features, as the front of the bridge shows a complex curvature. In this chapter, we first use the spline curves of the outline to construct a multi-patch spline surface of the back side. The spline surface can then trivially be extended to the third dimension, to yield a complete volume representation of the bridge. The control points of the front side are subsequently fitted to match the z-direction of the μ-CT scan. Compared to the outline, less data points are available, so we need to consider a more robust fit.

4.1. Creating the multi-patch spline surface using Coon's patch

Each of the constructed spline curves is a boundary of one of the 16 patches. To create the two-dimensional patch, we apply a Coon's patch (Coons, 1967), using the Matlab implementation of the NURBS toolbox (Spink et al., 2017). It takes four spline curves passing through four corner points and constructs two surfaces between each of the two pairs of oppositely arranged splines. Furthermore a third surface interpolating the four corner points is constructed by bilinear interpolation. The three surfaces are then blended into one blended surface which is the surface resulting from the Coon's patch. Since the quality of the constructed Coon's patch strongly depends on the quality of the parametrization of the interpolated spline
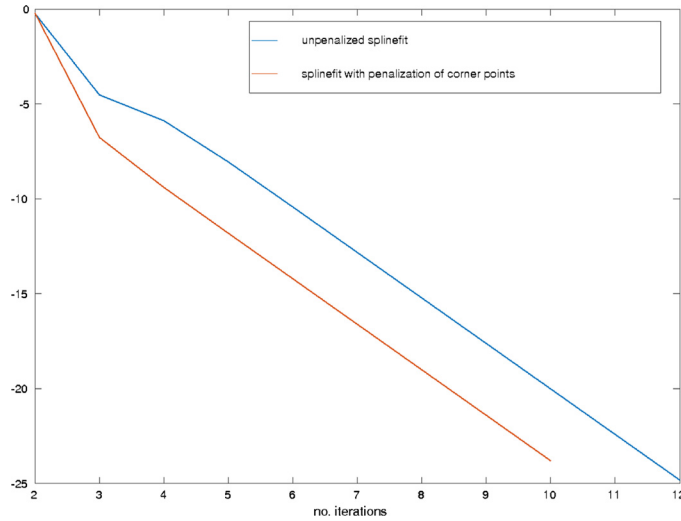
Fig. 8. Estimated error for the spline fit of the example illustrated in Fig. 9, Section 5. The blue line shows the evolution of the estimated error of the optimiser without any regularization terms, the orange line shows the improved evolution of the estimated error when we interpolate corner points. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

curves, there might be situations where the Coon's patch might fail to give a useful surface construction. For instance, it can happen that a part of the resulting surface lies outside the bounding curves. In this case there is a large variation of other planar parametrisation methods available. For example, a good survey can be found in Xu et al. (2011) or Gravesen et al. (2014), the latter one with a special focus on parametrisations in the context of shape optimisation problems. In Gravesen et al. (2014) one also finds an explanation of the spring model, which is a standard alternative for the construction of inner control points. Furthermore, one can find approaches of improved uniformness in Xu et al. (2013) and Nian et al. (2016).

Due to the presence of 16 interfaces, at which neighbouring patches touch, not all of the patch boundaries are fitted spline curves. Besides two exceptions which were inherited from the reference geometry, the internal interfaces are planar. As such, they can simply be constructed based on the end points of the spline curves. Thanks to the strict interpolation of the end points as described in Section 3.3.1, there are no gaps between two neighbouring patches. With this consecutive procedure of inheriting interface splines from one patch to the other we get a watertight multi-patch domain and avoid modification of the fitted spline basis as it would be necessary when trimming NURBS surfaces (Schmidt et al., 2012; Kim et al., 2010).

As a result, we obtain the B-spline-surfaces

$$S(\xi, \eta) = \sum_{j=1}^{J} \sum_{k=1}^{K} N_j(\xi) \cdot N_k(\eta) \cdot \begin{pmatrix} C_{j,k}^x \\ C_{j,k}^y \end{pmatrix} \tag{14}$$

for each of the 16 patches.

4.2. Fitting of NURBS volumes

The final step to obtain a precise spline geometry of the violin bridge is to extrude the spline surface to a three-dimensional solid. We build the spline surface for the front and the back side of the violin bridge, which can then be linearly connected to form a three-dimensional geometry. The back side of the violin bridge is flat and lies within the x-y-plane, so it requires no data fitting. The front side has the same outline in the x-y-plane, but it has a complicated curvature in the z-direction that needs to be fitted. Compared to the fit of the boundary curves, this is less involved as we do not need to change the x and y components and thus do not need to optimise the parameter values. Instead, only the z-component C_{jkL}^z of the control points $\mathbf{C}_{jkL} = (C_{jkL}^x, C_{jkL}^y, C_{jkL}^z)^\top$ of the front surface needs to be optimised, such that the curved spline surface matches the z-components $\{w_i\}_{i=1}^{I}$ of the data points, i.e., we solve

$$\min_{C_{jkL}^z} \sum_{i=1}^{I} \left(\sum_{j=1}^{J} \sum_{k=1}^{K} N_j(\xi_i) N_k(\eta_i) \cdot C_{jkL}^z - w_i \right)^2 \tag{15}$$

The parameter values (ξ_i, η_i) belonging to a data point (u_i, v_i, w_i) are such that the x and y coordinates of the surface match the ones of the data point. They can be computed by inverting equation (14) at $\{u_i, v_i\}$ using a Gauss-Newton
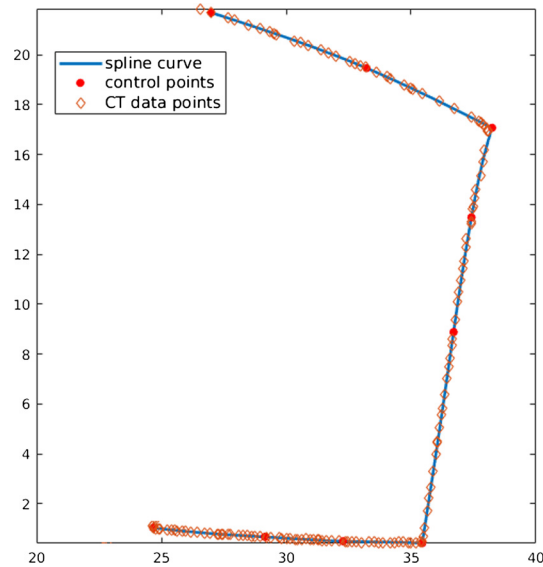
Fig. 9. The blue line show the spline fit for v_1-curve of patch 1 incorporating the characteristic properties of the data points (orange diamonds) like corner points. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

approach. Once the curved spline geometry of the front side has been fitted, it can be linearly connected to the back side to form the three-dimensional geometry.

We note that directly solving the least-squared problem (15) did not show satisfying results, due to overfitting and insufficient data quality. Some data points that are actually in the interior of the solid geometry were identified by the CT-scan to be on the surface. Instead of manually identifying these outliers, we fit a smooth surface which is lying above all data points. Rather than minimising the least-squares error, we use the condition that the surfaces lies above the data points as a constraint and simply minimise a norm which enforces a smooth surface. With this inequality condition, the outliers in the interior of the geometry are no longer relevant.

More precisely, we solve

$$\min_{C^z_{jkl}} \left\| S(\cdot, \cdot; C^z) \right\|^2_{L^2} + \alpha \left\| \nabla_{(x,y)} S(\cdot, \cdot; C^z) \right\|^2_{L^2},$$

$$\text{s.t.} \ S(\xi_i, \eta_i; C^z) \geq w_i, \quad i = 1, \ldots I,$$

where $S(\xi, \eta; C^z) = \sum_{j=1}^{J} \sum_{k=1}^{K} N_j(\xi) N_k(\eta) C^z_{jkL}$ is the z-component of the parametric surface representation and $\alpha \in \{0, 1\}$ is a patch-dependent regularization parameter. In our case $\alpha = 1$ was required for patches $6, 7, 8, 14, 15, 16$, where the surface data point were particularly sparse. The quadratic programming problem can be solved with an out-of-the-box solver, e.g., using `quadprog` in Matlab.

5. Examples

In this section we discuss a computational example from the geometry fit for the μ-CT scan data of the violin bridge. As explained in Chapter 1, we divided its spline geometry into 16 patches; see also Fig. 2 for the numbering of the patches. For all plots in this chapter the lengths are measured in millimeters.

In Figs. 9 and 10 one can see the different steps of the surface fit exemplified for patch 1. First, Fig. 9 shows the data as well as the spline fit for the v_1-curve of this patch after the fitting process like it is describe in Chapter 3.2. One can see that end- as well as edge points are met precisely thanks to the constraints on this characteristic control points; see Chapter 3.3.1 for implementational details. Also we remark that we get a reasonable spacing of the control points along the spline curve due to the regularisation term penalizing the distance of the control points; see Chapter 3.3.2.

Fig. 10 then shows the complete patch surface constructed from the four fitted boundary splines (for the u_1, u_2, v_1, v_2-directions respectively). The little red crosses show the data points of the measured μ-CT data superposed with the patch boundary. As one can see we benefit here from the preparations we took during the construction of the boundary splines. Despite the complicated curvature of the patch we do not get any intersecting splines inside the surface.

In Fig. 11, one can see how the final multi-patch surface geometry of the complete violin bridge looks after the fitting procedure has been performed for all 16 patches and the interfaces between the patches have been constructed as described in Chapter 4.1. Finally, Fig. 12 shows the 3D-geometry after the fitting of the third dimension has been performed; see Chapter 4.2. The distribution of the scan data to the patches which was used for the fit of the curved surface is indicated
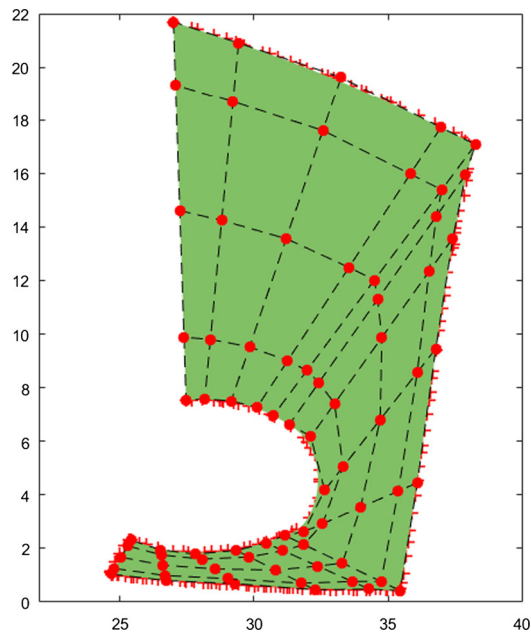
Fig. 10. Spline surface fitted to the data of patch 1. The red bullets represent the control net of the patch. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)
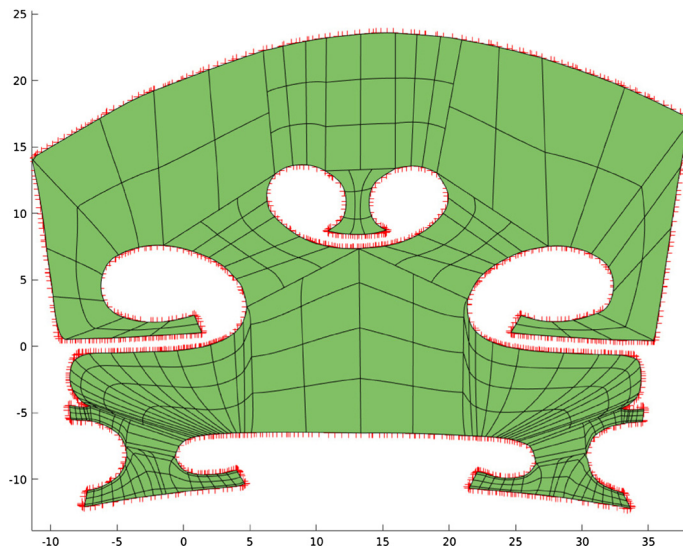

Fig. 11. The complete fitted 2D surface-geometry. The data points are indicated by orange crosses. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

by the different colours of the data points and one can see that the vast majority of the data points are approximated very well.

6. An application: vibro-acoustical simulations for a violin bridge

Being the first filter in the pathway of energy transmission from the vibrating string to the violin corpus, the violin bridge has a significant influence on the tonal behaviour and the sound of an instrument (Woodhouse, 2005; Cremer and Allen, 1984). On the other hand, the influence of certain geometric modifications on the filter properties of the bridge is not very well understood. An accurate finite element model can therefore be quite valuable to establish guidelines which may lead to designs with certain desirable tonal properties.

With the precise geometry at hand, we are ready to run accurate vibro-acoustical simulation for the bridge and solve the eigenvalue problem of linear elasticity:
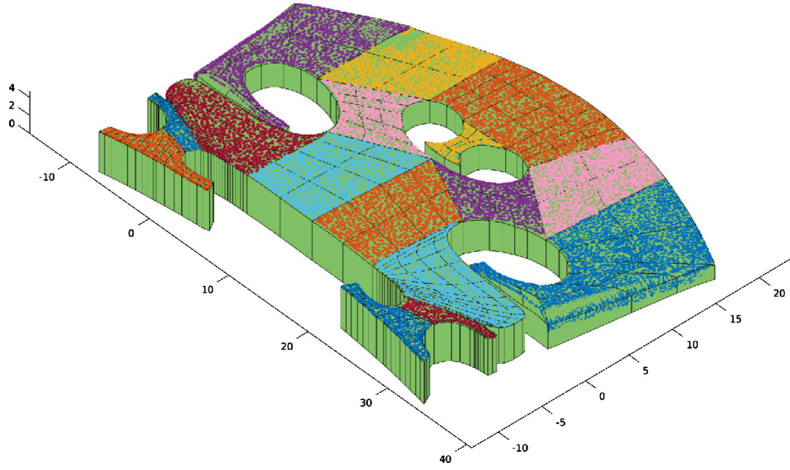
Fig. 12. The 3D geometry fitted to the data points. The data points are coloured patch-wise. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

$$-\operatorname{div}\boldsymbol{\sigma}(\mathbf{u}) = \lambda\rho\mathbf{u},$$

where $\rho > 0$ is the constant mass-density, $\boldsymbol{\sigma}(\mathbf{u}) = \mathbb{C}\boldsymbol{\varepsilon}(\mathbf{u})$ the stress tensor for the orthotropic material and $\boldsymbol{\varepsilon}(\mathbf{u}) = (\nabla\mathbf{u} + \nabla\mathbf{u}^\top)/2$ is the infinitesimal strain tensor. The orthotropic stiffness tensor is given by

$$\mathbb{C} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 & 0 \\ A_{31} & A_{32} & A_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{yz} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{zx} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{xy} \end{pmatrix},$$

with the shear moduli G_{xy}, G_{yz}, G_{zx} and the entries A_{ij} depending on the elastic moduli E_x, E_y, E_z and the Poisson's ratios $\nu_{xy}, \nu_{yz}, \nu_{zx}$. The exact formula for A_{ij} can be found in Rand and Rovenski (2007, Chapter 2.4).

The idea of isogeometric analysis (Cottrell et al., 2009; Höllig, 2003) is to use the same discretization for the FE analysis as it was used for the discretization, i.e., NURBS. To flexibly handle the multi-patch geometry of the violin bridge, we used an isogeometric mortar domain decomposition (Brivadis et al., 2015; Horger et al., 2017) to discretize the fitted geometry. The tri-variate spline space $V_n = \operatorname{span}\{N^p_{j,k,l}\}$ is considered for each patch Ω_n and the broken space $\mathbf{V}_h = \prod_{n=1}^{16} V_n^3$ is weakly coupled on each of the 16 interfaces. We note that the mesh size for the analysis needs to be significantly smaller than for the geometry description, so one additional step of uniform h-refinement (where additional knots are added to enrich spline spaces) is performed. Also the polynomial degree in all directions is raised to the same level, namely three. For each interface γ_i the two adjacent domains are labelled as one slave and one master domain, i.e., $\gamma_i = \partial\Omega_s \cap \partial\Omega_m$. The coupling space M_n is set as the trace space of the spline spaces on the slave domain and $\mathbf{M}_h = \prod_{n=1}^{16} M_n^3$. On several selected cross points an appropriate local degree reduction is performed to guarantee stability (Brivadis et al., 2015).

We use the standard bilinear forms for mortar methods in linear elasticity

$$a(\mathbf{u}, \mathbf{v}) = \sum_{n=1}^{16} \int_{\Omega_n} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}), \quad m(\mathbf{u}, \mathbf{v}) = \sum_{n=1}^{16} \int_{\Omega_n} \rho\,\mathbf{u}^\top\mathbf{v},$$

$$b(\mathbf{v}, \boldsymbol{\tau}) = \sum_{\tilde{n}=1}^{16} \int_{\gamma_{\tilde{n}}} [\mathbf{v}]_{\tilde{n}}^\top \boldsymbol{\tau},$$

where $[\mathbf{v}]_{\tilde{n}} = \mathbf{v}_s|_{\gamma_{\tilde{n}}} - \mathbf{v}_m|_{\gamma_{\tilde{n}}}$ denotes the jump across the interface $\gamma_{\tilde{n}}$. Then the discretization of the vibro-acoustical eigenvalue problem is given by $(\mathbf{u}_h, \boldsymbol{\mu}_h) \in \mathbf{V}_h \times \mathbf{M}_h$, $\lambda_h \in \mathbb{R}$, such that

$$a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, \boldsymbol{\mu}_h) = \lambda_h\, m(\mathbf{u}_h, \mathbf{v}_h), \quad \mathbf{v}_h \in \mathbf{V}_h,$$

$$b(\mathbf{u}_h, \boldsymbol{\tau}_h) = 0, \quad \boldsymbol{\tau}_h \in \mathbf{M}_h.$$
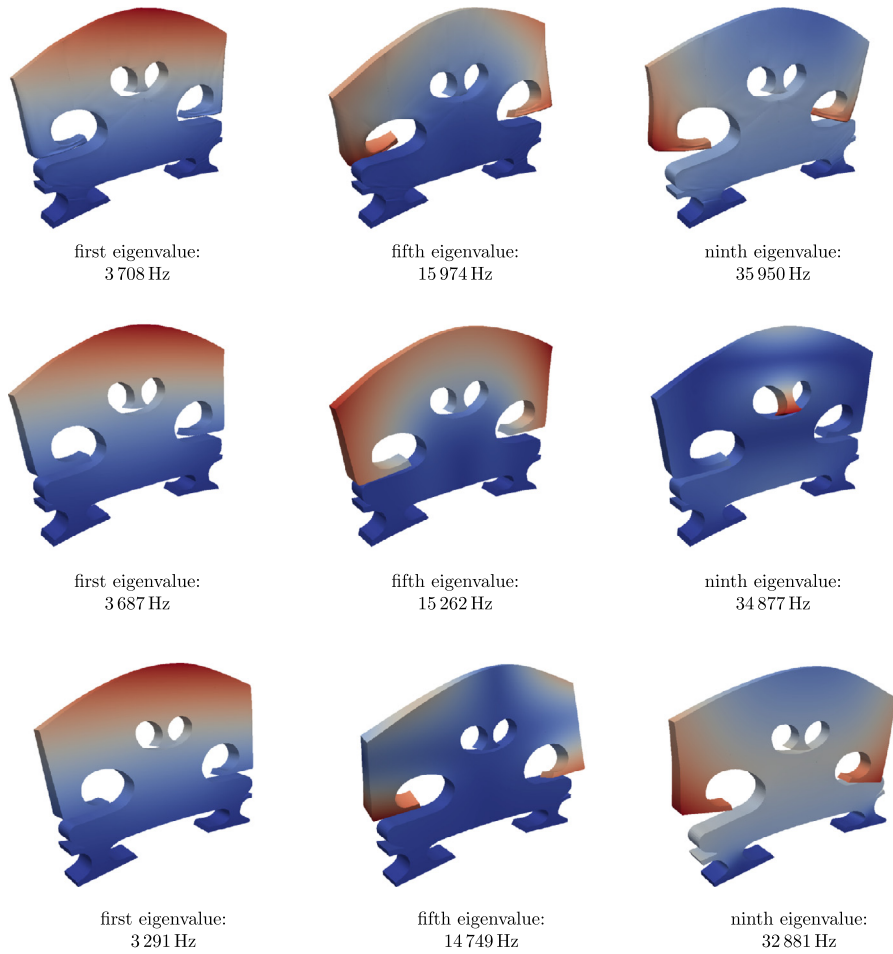
Fig. 13. Visualisation of the three eigenmodes with the associated eigenvalue. Top row: three-dimensional spline fit; Middle row: fitted outline with a linear thickness; Bottom row: reference geometry. All geometries share the same size. (For interpretation of the colours in the figure, the reader is referred to the web version of this article.)

Table 1
Comparison of the eigenvalues for the reference and the fitted geometry.

Eigenvalue	Linear fit reference geometry	Linear fit 2D-CT-geometry	3D CT-geometry
1st	3 291 Hz	3 687 Hz	3 708 Hz
2nd	5 889 Hz	6 118 Hz	6 957 Hz
3rd	11 606 Hz	11 208 Hz	12 494 Hz
4th	14 552 Hz	14 165 Hz	15 730 Hz
5th	14 748 Hz	15 262 Hz	15 974 Hz
6th	21 536 Hz	21 737 Hz	23 401 Hz
7th	26 903 Hz	26 287 Hz	29 136 Hz
8th	27 710 Hz	30 621 Hz	31 730 Hz
9th	32 882 Hz	34 877 Hz	35 950 Hz
10th	34 128 Hz	36 947 Hz	37 140 Hz

The primal solution $\mathbf{u}_h$ approximates the eigenmode, the Lagrange multiplier $\boldsymbol{\mu}_h$ is an approximation for the surface tension $\boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n}$ along the interfaces and the eigenvalue λ_h is an approximation to the square of the angular frequency, i.e., $\lambda_h = \omega_h^2$. Fig. 13 shows three eigenmodes for the reference geometry (top row) compared to the same three eigenmodes of the fitted CT-geometry with a linear thickness (middle row) and the fully fitted three-dimensional geometry which was reconstructed from the μ-CT scan (bottom row). The reference geometry was scaled linearly to have the same overall size as the precise geometry. We can see that the geometry has a significant effect on the eigenvalues and on the displacement of the eigenfunctions. The first ten eigenvalues for the different geometries are compared in Table 1.

7. Conclusion

We have presented the workflow to perform numerical simulations for a violin bridge with a precise geometry description, based on a μ-CT scan. Creating a suitable multi-patch geometry required three main steps – the spline fit of the outline, the extension to a two-dimensional geometry and a spline fit in the third dimension. During the spline fit, special care was taken to avoid overfitting and to meet characteristics, e.g., corners. The final numerical results show the high sensitivity of the vibro-acoustical problem to changes on the geometry.

Declaration of competing interest

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Acknowledgement

References

Beirão Da Veiga, L., Buffa, A., Sangalli, G., Vásquez, R., 2014. Mathematical analysis of variational isogeometric methods. Acta Numer. 23, 157–287.

Ben-Israel, A., Greville, T.N., 2003. Generalized Inverses: Theory and Applications. CMS Books in Mathematics. Springer.

Bo, P., Ling, R., Wang, W., 2012. A revisit to fitting parametric surfaces to point clouds. In: Shape Modeling International (SMI) Conference 2012. Comput. Graph. 36 (5), 534–540.

Boffi, D., Brezzi, F., Fortin, M., 2013. Mixed Finite Element Methods and Applications. Springer.

Borges, C.F., Pastva, T., 2002. Total least squares fitting of Bézier and B-spline curves to ordered data. Comput. Aided Geom. Des. 19 (4), 275–280.

Brivadis, E., Buffa, A., Wohlmuth, B., Wunderlich, L., 2015. Isogeometric mortar methods. Comput. Methods Appl. Mech. Eng. 284, 292–319.

Buchegger, F., Jüttler, B., 2017. Planar multi-patch domain parameterization via patch adjacency graphs. In: Isogeometric Design and Analysis. Comput. Aided Des. 82, 2–12. https://doi.org/10.1016/j.cad.2016.05.019.

Buzug, T., 2008. Computed Tomography. From Photon Statistics to Modern Cone-Beam CT. Springer-Verlag, Heidelberg.

Carnicer, J., Delgado, J., Peña, J., 2012. Progressive iteration approximation and the geometric algorithm. Comput. Aided Des. 44 (2), 143–145.

Coons, S.A., 1967. Surfaces for computer-aided design of space forms. Tech. Rep. Massachusetts Institute of Technology.

Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y., 2009. Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley.

Cremer, L., Allen, J., 1984. The Physics of the Violin. MIT Press.

Deng, C., Lin, H., 2014. Progressive and iterative approximation for least squares B-spline curve and surface fitting. Comput. Aided Des. 47, 32–44.

Dierckx, P., 1995. Curve and Surface Fitting with Splines. Clarendon Press.

Dimitrov, A., Golparvar-Fard, M., 2014. In: Robust NURBS Surface Fitting from Unorganized 3D Point Clouds for Infrastructure As-Built Modeling. Computing in Civil and Building Engineering, 2014. American Society of Civil Engineers, pp. 81–88.

Falini, A., Jüttler, B., 2019. Thb-splines multi-patch parameterization for multiply-connected planar domains via template segmentation. J. Comput. Appl. Math. 349, 390–402. https://doi.org/10.1016/j.cam.2018.08.011.

Fletcher, N.H., Rossing, T., 1998. The Physics of Musical Instruments, 2nd edition. Springer-Verlag, New York.

Foley, T.A., Nielson, G.M., 1989. Knot selection for parametric spline interpolation. In: Lyche, T., Schumaker, L. (Eds.), Mathematical Methods in Computer Aided Geometric Design. Academic Press, pp. 261–272.

Gálvez, A., Iglesias, A., Puig-Pey, J., 2012. Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. Inf. Sci. 182 (1), 56–76.

Gravesen, J., Evgrafov, A., Nguyen, D.-M., Nørtoft, P., 2014. Planar parametrization in isogeometric analysis. In: Mathematical Methods for Curves and Surfaces. 2012. Springer, Berlin Heidelberg, pp. 189–212.

Grove, O., Rajab, K., Piegl, L.A., 2010. From CT to NURBS: contour fitting with B-spline curves. Comput-Aided Des. Appl. 7 (sup1), 1–19.

Han, X.-F., Jin, J.S., Wang, M.-J., Jiang, W., Gao, L., Xiao, L., 2017. A review of algorithms for filtering the 3d point cloud. Signal Process. Image Commun. 57, 103–112. https://doi.org/10.1016/j.image.2017.05.009.

Höllig, K., 2003. Finite Element Methods with B-Splines. Frontiers in Applied Mathematics. SIAM.

Horger, T., Wohlmuth, B., Wunderlich, L., 2017. Reduced basis isogeometric mortar approximations for eigenvalue problems in vibroacoustics. In: Benner, P., Ohlberger, M., Patera, A., Rozza, G., Urban, K. (Eds.), Model Reduction of Parametrized Systems. Springer International Publishing, Cham, pp. 91–106.

Kak, A.C., Slaney, M., 1988. Principles of Computerized Tomographic Imaging. IEEE Press, New York.

Kim, H.-J., Seo, Y.-D., Youn, S.-K., 2010. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. Comput. Methods Appl. Mech. Eng. 199 (45), 2796–2812.

Liu, M., Li, B., Guo, Q., Zhu, C., Hu, P., Shao, Y., 2018. Progressive iterative approximation for regularized least square bivariate B-spline surface fitting. J. Comput. Appl. Math. 327, 175–187.

Nian, X., Chen, F., 2016. Planar domain parameterization for isogeometric analysis based on Teichmüller mapping. Comput. Methods Appl. Mech. Eng. 311, 41–55.

Patraucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., Haas, C., 2015. State of research in automatic as-built modelling. Adv. Eng. Inform. 29 (2), 162–171.

Piegl, L., Tiller, W., 1997. The NURBS Book. Springer.

Rakotosaona, M.-J., Barbera, V.L., Guerrero, P., Mitra, N.J., Ovsjanikov, M., 2019. Pointcleannet: learning to denoise and remove outliers from dense point clouds. arXiv:1901.01060 [abs].

Rand, O., Rovenski, V., 2007. Analytical Methods in Anisotropic Elasticity: with Symbolic Computational Tools. Birkhäuser.

Ravari, A.N., Taghirad, H.D., 2016. Reconstruction of B-spline curves and surfaces by adaptive group testing. Comput. Aided Des. 74, 32–44.

Rueckel, J., Stockmar, M., Pfeiffer, F., Herzen, J., 2014. Spatial resolution characterization of a X-ray microCT system. Appl. Radiat. Isot. 94, 230–234.

Schall, O., Belyaev, A., Seidel, H., 2005. Robust filtering of noisy scattered point data. In: Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics. 2005, pp. 71–144.

Schmidt, R., Wüchner, R., Bletzinger, K.-U., 2012. Isogeometric analysis of trimmed nurbs geometries. Comput. Methods Appl. Mech. Eng. 241–244, 93–111.

Schumaker, L., 2007. Spline Functions: Basic Theory, 3rd edition. Cambridge University Press, Cambridge.

Seiler, A., Großmann, D., Jüttler, B., 2018. Spline surface fitting using normal data and norm-like functions. Comput. Aided Geom. Des. 64, 37–49.

Singhal, A., Grande, J., Zhou, Y., 2013. Micro/nano-CT for visualization of internal structures. Micros. Today 21 (2), 16–22.

Spink, M., Claxton, D., de Falco, C., Vazquez, R., 2017. The NURBS toolbox. https://octave.sourceforge.io/nurbs/, package Version: 1.3.13.

Wang, J., Xu, K., Liu, L., Cao, J., Liu, S., Yu, Z., Gu, X.D., 2013. Consolidation of low-quality point clouds from outdoor scenes. In: Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing. SGP '13. Eurographics Association, pp. 207–216. https://doi.org/10.1111/cgf.12187.

Weiss, V., Andor, L., Renner, G., Varady, T., 2002. Advanced surface fitting techniques. Comput. Aided Geom. Des. 19 (1), 19–42.

Woodhouse, J., 2005. On the "bridge hill" of the violin. Acta Acust. Acust. 91 (1), 155–165.

Xiao, S., Kang, H., Fu, X.-M., Chen, F., 2018. Computing iga-suitable planar parameterizations by polysquare-enhanced domain partition. Comput. Aided Geom. Des. 62, 29–43. https://doi.org/10.1016/j.cagd.2018.03.008.

Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2011. Parameterization of computational domain in isogeometric analysis: methods and comparison. Comput. Methods Appl. Mech. Eng. 200 (23), 2021–2031. https://doi.org/10.1016/j.cma.2011.03.005.

Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2013. Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method. J. Comput. Phys. 200 (252), 275–289. https://doi.org/10.1016/j.jcp.2013.06.029.

Zhang, Y., Cao, J., Chen, Z., Li, X., Zeng, X.-M., 2016. B-spline surface fitting with knot position optimization. Comput. Graph. 58, 73–83.