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

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

[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*

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 apply three steps: fitting the outline, 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), which we extend by a regularisation and a precise interpolation of selected data points. The resulting NURBS geometry is ready to apply isogeometric analysis tools for efficient numerical simulation.

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 an vibro-acoustical application.

1. Introduction

In mechanical simulation problems concerning complex-shaped real world objects, conventional geometric measurement methods are often not giving satisfactory results. A remedy in such cases may be a μ-CT scan of the considered object which generates a set of three-dimensional data points organised in a surface triangulation. The question addressed in this paper is how to construct an accurate spline representation of the object described by such a data set,

*Corresponding author
Email address: sandra.marschke@uni-graz.at (Sandra Marschke)

which is ready for a numerical simulation. A solution based on isogeometric analysis requires an accurate fit of the volumetric geometry to the data as well as a suitable numerical method to perform the simulation on the complex geometry. The complete workflow based on a non-linear least-squares problem [1] and isogeometric mortar methods [2] is presented in detail throughout this paper.

Fitting spline geometries is an old problem as splines were soon recognised to have good approximation properties, which are favourable for geometric design [3]. While splines were originally designed for one-dimensional interpolation, their use in geometry approximation is usually based on the solution of a (possibly non-linear) minimization problem.

A crucial part of the fitting process is the parametrisation of the geometry. When the parameter value for each surface point which is fitted to a given data point is fixed, the optimisation of the spline geometry reduces to a quadratic optimisation problem, e.g. [4, 5]. For more precise and flexible results there exist various non-linear optimisation approaches which include the position of the knots [6], the parameter values [7] or different kinds of error projection methods between the data and the fitted surface [8]. 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 [1], where 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 of the spline, we regularise the non-linear least squares problem. 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 [9, 10, 11]. 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, which 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 speed of convergence.

We also mention other approaches to surface fitting which make use of genetic algorithms [12] or adaptive group testing [13]. 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 [14]. 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 informa-

tion. Another related applications is as-built modeling in engineering [15, 16]. Furthermore we would like to emphasise, that the way of constructing multi-patch surfaces of real world CAD models which we present in this paper can be a computationaly cost-saving alternative to techniques like they are presented in [17] leading to $G1$-smoothness of the composite surface or techniques using trimmed NURBS [18, 19] since we avoid effortful adaptation of the spline basis describing the geometry and it also spares us the modification of the numerical method.

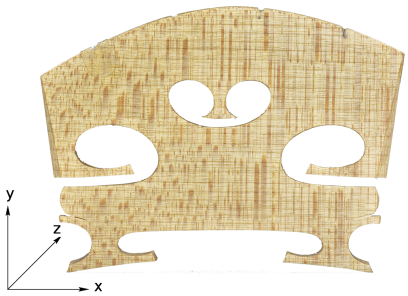


Figure 1: Photo of the scanned violin bridge.

We illustrate our approach by the particular example of a violin bridge, as shown in Figure 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 the acoustic properties of the bridge [20, 21], a highly accurate computational geometry is essential for precise vibro-acoustical simulations.

In this article, 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 [22, 23] and the pre-processing of the generated data. Then a spline volume is fitted to the data using a total least-squares approach [1]. Finally, we solve a vibro-acoustical eigenvalue problem with isogeometric methods [24, 25]. 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 [2].

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, the construction of a two-dimensional multi-patch surface (see Chapter 4.1) and a constrained quadratic optimisation in the third dimension (see Chapter 4.2). In Chapter 5 we point out the capabilities of the algorithm with some detailed examples and

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: the μ-CT scan of the geometry and 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 at a recently developed μ-CT device, a 'v|tome|x s' (GE Sensing & Inspection Technologies GmbH, Ahrensburg, Germany), at the Technical University of Munich and its specific technical setup and mode of operation is explained in more details in the following. For an in-depth background on the physics and mathematics of computed tomography (CT) we refer to [22, 23].

Whereas in clinical CTs, X-ray tube and detector rotate around the patient, in μ-CTs the sample rotates whereas tube and detector stay in fixed positions. The X-ray tube was a 'xs 240 D' direct 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 [22, 23]. The minimum spot size of this tube is 7 µm [26] but goes up to about 10 µm at the tube settings used for the experiment [27]. A 1000x1000 pixel detector 'DXR-250RT' (GE, Germany) was used without further binning of the pixels of 200 µm x 200 µm size. The 'DXR-250RT' 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 but at the same time ensure a stable mounting of the sample. 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 [27].

The volumetric data was converted into a surface mesh in the 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 gray value 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 amount of 16 patches was adopted by a reference geometry which incorporated the main topological features. Each spline patch parametrisation is based on the control points $\mathbf{C}^n_{jkl} = (C^{x,n}_{jkl}, C^{y,n}_{jkl}, C^{z,n}_{jkl})^\top \in \mathbb{R}^3$ and defined as the diffeomorphic image of a spline function $\mathbf{F}_n \colon \widehat{\Omega} \to \mathbb{R}^3$, $\Omega_n = \mathbf{F}_n(\widehat{\Omega})$ 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 $(N^p_{jkl})_{j,k,l}$ of non-uniform rational B-splines (NURBS) of degree p, the parametrised mapping for each patch is given as

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

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

$$B^p_{jkl}(\xi, \eta, \theta) = B^p_j(\xi) B^p_k(\eta) B^p_l(\theta),$$

with one-dimensional spline bases of order p, $B^p_j, B^p_k, B^p_l \colon (0, 1) \to \mathbb{R}$. 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 $p+1+J$, where 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 while keeping the overall length of $\boldsymbol{\zeta}$ at $p + 1 + J$ reduces the smoothness of the basis in one corresponding point [24]. 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

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

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

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 [3, 24].

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.

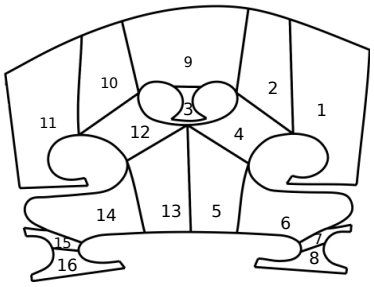


Figure 2: Patch decomposition of the reference geometry.

In our case, we start from a rough initial bridge geometry, which we use to select the patch-decomposition, see Figure 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 is given [28] 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 for each patch, $\mathbf{C}_{jkl} = (C^x_{jkl}, C^y_{jkl}, C^z_{jkl})^\top \in \mathbb{R}^3$, 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 [29] (see Chapter 4.1). And 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 data. Particularly, we have to remove some outliers which arose from measurement errors. Then we can continue to find the vertices that describe the contour of the scanned object. As we are provided with an STL file which contains a surface triangulation of the measured object, we can 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. Manually determining the minimal and maximal z-component $w_{\min}$, $w_{\max}$ of the exact geometry allows us to remove obvious outliers lying outside of this range. Therefore any data points with a z-coordinate outside $[w_{\min}, w_{\max}]$ are removed.

Afterwards we use an orthogonal projection to map 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 outline of the 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 uses 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 center of mass of the data points. The assignment of these data points to the 16 patches of the future spline geometry is displayed in Figure 3.

3.2. Fitting the boundary splines

In this chapter, we describe one of the key procedures: the fitting of the (one-dimensional) boundary splines. In [1], an algorithm was proposed that uses a Gauss-Newton approach to minimise the total least squares error between ordered data and a B-spline curve. In this approach both, the control points and the parameter values at which the spline function values are compared with the data points, appear as optimisation variables. Adapting this idea and incorporating additional constraints, we obtain a method that performs a precise fit of complicated curved shapes which may contain sharp edges. While we solve a single minimisation problem with respect to a pair of optimisation variables, other recent approaches driven by similar ideas require two separate optimisation steps [12]. Some illustrative examples for the applicability of the approach are later presented in Section 5.

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

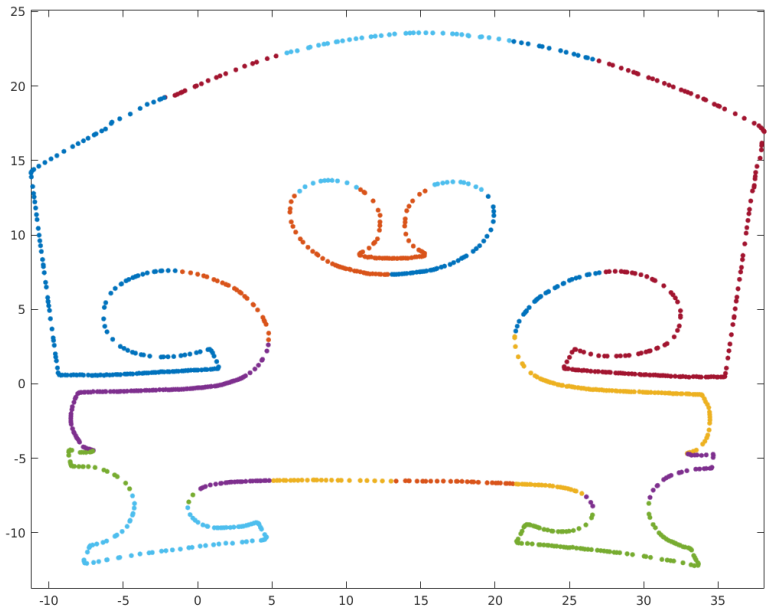


Figure 3: Distribution of the data points to the 16 different patches of the computational geometry. The assignment of the data points to a patch can be distinguished by the different colours of data points of neighboring patches.

points $\{\xi_i\}_{i=1}^I$ with $0 = \xi_1 < \xi_2 < \cdots < \xi_I = 1$ and J control points $(C_j^x, C_j^y)^\top$ that minimise the quadratic distance between the curve evaluated at the parametric points and the data:

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

We remark that by ordered we mean that the data points are ordered in the sense of 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.

To keep the article selfcontained, we shortly recapitulate the applied method and then extend it by incorporating several constraints. For further details, we refer to the original article [1].

Introducing the *B-spline basis matrix* which contains all J 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:

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

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 with a linear optimality system, which we can solve easily. The optimality conditions in this case are

$$\langle \mathbf{u} - \mathcal{N}\mathbf{C}^x, \mathcal{N}\boldsymbol{\tau}\rangle_2 = 0 \text{ and} \tag{2}$$

$$\langle \mathbf{v} - \mathcal{N}\mathbf{C}^y, \mathcal{N}\boldsymbol{\tau}\rangle_2 = 0 \text{ for all } \boldsymbol{\tau} \in \mathbb{R}^I. \tag{3}$$

The solution to the optimality system then can be writen as

$$\mathbf{C}^x = \mathcal{N}^+(\boldsymbol{\xi})\,\mathbf{u} \text{ and } \mathbf{C}^y = \mathcal{N}^+(\boldsymbol{\xi})\,\mathbf{v}$$

where $\mathcal{N}^+$ denotes the Moore Penrose inverse $\mathcal{N}^+ = (\mathcal{N}^\top\mathcal{N})^{-1}\mathcal{N}^\top$ of the matrix $\mathcal{N}$, see [30].

After the elimination of the control points we can consider the reduced problem

$$\|\mathbf{r}(\boldsymbol{\xi})\|_2^2 = \left\|\mathbf{u} - \mathcal{N}(\boldsymbol{\xi})\,\mathcal{N}^+(\boldsymbol{\xi})\,\mathbf{u}\right\|_2^2 + \left\|\mathbf{v} - \mathcal{N}(\boldsymbol{\xi})\,\mathcal{N}^+(\boldsymbol{\xi})\,\mathbf{v}\right\|_2^2 \longrightarrow \min_{\boldsymbol{\xi}} . \tag{4}$$

Here we have set $\mathbf{r}(\boldsymbol{\xi}) = \left(\mathbf{r}^x(\boldsymbol{\xi})^\top, \mathbf{r}^y(\boldsymbol{\xi})^\top\right)^\top$ with $\mathbf{r}^x(\boldsymbol{\xi}) = \left(I - \mathcal{N}(\boldsymbol{\xi})\,\mathcal{N}^+(\boldsymbol{\xi})\right)\mathbf{u}$ and $\mathbf{r}^y(\boldsymbol{\xi}) = \left(I - \mathcal{N}(\boldsymbol{\xi})\,\mathcal{N}^+(\boldsymbol{\xi})\right)\mathbf{v}$. For simplicity, subsequently we use the notation $r(\boldsymbol{\xi}) = \|\mathbf{r}(\boldsymbol{\xi})\|_2^2$. 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 $\{\xi_i\}_{i=1}^I$. Similar methods like [31] minimising (4) for the optimal control points and the optimal parameter values use an BFGS approach. In the following we present a minimisation scheme based on a Gauss-Newton approach.
To iteratively minimise $\mathbf{r}(\boldsymbol{\xi})$ as stated in (4), we linearize $\boldsymbol{r}$ and consider the quadratic least squares problem

$$\|\mathbf{r}(\boldsymbol{\xi}_k) + J_{\mathbf{r}}(\boldsymbol{\xi}_k)\left(\boldsymbol{\xi}_{k+1} - \boldsymbol{\xi}_k\right)\|_2^2 \longrightarrow \min_{\boldsymbol{\xi}_{k+1}},$$

leading to the iterative Gauss-Newton scheme:

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

Here, as above, $J_{\mathbf{r}}^{+}\left(\boldsymbol{\xi}_k\right)$ denotes the Moore Penrose inverse of the Jacobi matrix $J_{\mathbf{r}}\left(\boldsymbol{\xi}_k\right)$ for the parameter vector $\boldsymbol{\xi}_k$ at iteration step k. To evaluate $J_{\mathbf{r}}^{+}$ we need to compute $J_{\mathbf{r}}\left(\boldsymbol{\xi}\right)$ with the entries $J_{\boldsymbol{r}}\left(\boldsymbol{\xi}\right)_{i,j} = \partial r_i/\partial\xi_j$, which requires closer examination of the residual vector $\mathbf{r}\left(\boldsymbol{\xi}\right)$.

Using the definition of the Moore Penrose inverse, it is straightforward to show that the Matrix $P_{\mathrm{rg}(\mathcal{N})} = \mathcal{N}\mathcal{N}^{+}$ is symmetric and has the property that $P_{\mathrm{rg}(\mathcal{N})}^2 = P_{\mathrm{rg}(\mathcal{N})}$. Moreover, the ranges of $\mathcal{N}$ and $P_{\mathrm{rg}(\mathcal{N})}$ coincide, i.e. we have $\mathrm{rg}(\mathcal{N}) = \mathrm{rg}(P_{\mathrm{rg}(\mathcal{N})})$. Therefore, $P_{\mathrm{rg}(\mathcal{N})}$ is the orthogonal projection from $\mathbb{R}^I \to \mathrm{rg}(\mathcal{N}) \subset \mathbb{R}^I$. Note that $P_{\mathrm{rg}(\mathcal{N})^\perp} = I - P_{\mathrm{rg}(\mathcal{N})}$ is the orthogonal projection onto the orthogonal complement $\mathrm{rg}(\mathcal{N})^\perp$ of $\mathrm{rg}(\mathcal{N})$.

With this on hand, we can rewrite $r(\boldsymbol{\xi}) = \|P_{\mathrm{rg}(\mathcal{N})^\perp}\mathbf{u}\|_2^2 + \|P_{\mathrm{rg}(\mathcal{N})^\perp}\mathbf{v}\|_2^2$. So, to compute the Jacobian $J_{\mathbf{r}}$, we need to be able to evaluate $\partial P_{\mathrm{rg}(\mathcal{N})^\perp}/\partial\xi_i$. We use the following Lemma which was shown in [1, 32]:

Lemma 1. *Let $A = A(\boldsymbol{\xi}) \in \mathbb{R}^{I\times J}$ be a differentiable matrix valued function and let $A^- \in \mathbb{R}^{J\times I}$ be such that $AA^-A = A$ and $(AA^-)^\top = AA^-$ holds.(I.e. A^- is a generalised inverse for A. For example, the Moore Penrose inverse A^+ of A has the desired properties). Then we can compute the derivatives of the orthogonal projection $P_{(\mathrm{rg}A)^\perp} = I - AA^-$ as*

$$\frac{\partial P_{(\mathrm{rg}A)^\perp}}{\partial\xi_j} = -P_{(\mathrm{rg}A)^\perp}\frac{\partial A}{\partial\xi_j}A^- - \left(P_{(\mathrm{rg}A)^\perp}\frac{\partial A}{\partial\xi_j}A^-\right)^\top .$$

The Lemma states that we can calculate the derivatives of $P_{\mathrm{rg}(\mathcal{N})^\perp}$ when we can calculate the projection $P_{\mathrm{rg}(\mathcal{N})^\perp}$ itself and the Moore Penrose inverse or some other generalized inverse of $\mathcal{N}$. Using the QR-decomposition[33] of $\mathcal{N}$, we have

$$\mathcal{N} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}\begin{bmatrix} R \\ 0 \end{bmatrix}\Pi^\top , \tag{5}$$

where Π is a permutation matrix, $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \in \mathbb{R}^{I\times I}$ is orthogonal with $Q_1 \in \mathbb{R}^{I\times J}$ and $Q_2 \in \mathbb{R}^{I\times(I-J)}$ and $R \in \mathbb{R}^{J\times J}$ is upper triangular where $J = \mathrm{rank}(\mathcal{N})$ as assumed above. From this we easily get

$$\mathcal{N}^{+} = \Pi R^{-1}Q_1^\top . \tag{6}$$

It directly follows from (5), (6) that

$$P_{\mathrm{rg}(\mathcal{N})} = Q_1Q_1^\top , \qquad P_{\mathrm{rg}(\mathcal{N})^\perp} = Q_2Q_2^\top . \tag{7}$$

Applying Lemma 1 to our case and taking into account equations (6) and (7), we get

$$\frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial\xi_j} = -Q_2Q_2^\top\frac{\partial\mathcal{N}}{\partial\xi_j}\Pi R^{-1}Q_1^\top - \left(Q_2Q_2^\top\frac{\partial\mathcal{N}}{\partial\xi_j}\Pi R^{-1}Q_1^\top\right)^\top .$$

With this, the Jacobian

$$J_{\mathbf{r}} = \begin{pmatrix} \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_1}(\mathbf{u}), \ldots, \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_I}(\mathbf{u}) \\ \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_1}(\mathbf{v}), \ldots, \frac{\partial P_{\mathrm{rg}(\mathcal{N})^\perp}}{\partial \xi_I}(\mathbf{v}) \end{pmatrix} \tag{8}$$

can be calculated and the Gauss-Newton scheme can be performed.

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, we found necessary 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). The effect of these regularisation terms are controlled by parameters which have to be tuned to suppress undesirable features on the one hand, without disturbing the quality of the data fit on the other hand. To construct these regularisers, we make use of a-priori information mainly obtained from the known reference model [28]. We shall, however, also present ideas how to set up the regularisation terms if this information was not available.

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 and corners that are present in the geometry. When the start and end points are interpolated, the set of boundary splines automatically forms a closed curve. 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 optimization problem (1) and introduce the interpolation condition as a constraint. We first present a simple way to interpolate the end points as well as corners, based on the elimination of degrees of freedom from the equation system. Then we present a more general approach which allows for the interpolation of arbitrary points.

Eliminating corner and end points from the optimization system. We consider first 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 $\mathcal{I}_\lambda \subsetneq \{1, \ldots, I\}$. A useful property of B-splines is that when a knot ζ_l in the knot vector is occurs p times, the spline geometry interpolated the corresponding control point$(C_j^x, C_j^y)^\top$, see Section 2.2. In this case $N_j(\zeta_l) = \delta_{j,l}$ 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. With this approach, both the control points and the interpolating parameter values ξ_i for $i \in \mathcal{I}_\lambda$ are fixed and no longer act as optimisation variables in the analogous problem to (4). They can be eliminated as follows.

We define $\mathcal{I}_\lambda^c = \{1,\dots,I\} \setminus \mathcal{I}_\lambda$ as the complement of the index set $\mathcal{I}_\lambda$ corresponding to the data points. Furthermore we define $\mathcal{J}_\lambda$ as the subset of all indices $\{1,\dots,J\}$ which correspond to the interpolatory B-splines, due to a repeated knot within the knot vector. Using these B-splines, we shall interpolate 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 $\hat{\jmath} \in \mathcal{J}_\lambda$ and associated $\hat{\imath} \in \mathcal{I}_\lambda$ by $\hat{\imath} = \hat{\imath}(\hat{\jmath})$ and $\hat{\jmath} = \hat{\jmath}(\hat{\imath})$ respectively such that $B_{\hat{\jmath}}(\xi_{\hat{\imath}}) = \delta_{i\hat{\jmath}}$.

Fixing the parameters $\xi_{\hat{\imath}}$ for $\hat{\imath} \in \mathcal{I}_\lambda$ is trivial as the parameter vector is only updated in the outer iteration. To adapt the scheme, we thus have a closer look at the inner iteration, where the parameter vector $\boldsymbol{\xi}$ is fixed. 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 can eliminate the remaining control points:

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

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 adapted data vector

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

where $\tilde{\mathbf{v}} \in \mathbb{R}^{|\lambda^c|}$ is defined analogously to $\tilde{\mathbf{u}}$. By using this and additionally also the 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

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

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 and the algorithmic treatment can be adopted.

Constrained optimization for general characteristic points. A key requirement of the presented method is that the underlying spline space as repeated knots at the points of interpolation. The resulting reduced regularity is reasonable for corner points, but might be undesired 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 an 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 optimization variables. The optimization is only carried out with respect to the parameter values ξ_i with $i \in \mathcal{I}_\lambda^c$. We introduce the adapted B-spline basis matrices

$$\begin{aligned}(\mathcal{N}_\lambda^c)_{ij} &= N_j(\xi_i)\,, \quad i \in \mathcal{I}_\lambda^c\,,\ j \in \{1,\dots,J\}\,,\\ (\mathcal{N}_\lambda)_{ij} &= N_j(\xi_i)\,, \quad i \in \mathcal{I}_\lambda\,,\ j \in \{1,\dots,J\}\,.\end{aligned}$$

and the selected data vectors $\mathbf{u}_\lambda^c = (u_i)_{i\in\mathcal{I}_\lambda^c}$, $\mathbf{u}_\lambda = (u_i)_{i\in\mathcal{I}_\lambda}$ 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.

Then we can write the constrained optimisation problem for fixed parameter vector $\boldsymbol{\xi}$ as

$$\|\mathbf{u}_\lambda^c - \mathcal{N}_\lambda^c \mathbf{C}^x\|_2^2 + \|\mathbf{v}_\lambda^c - \mathcal{N}_\lambda^c \mathbf{C}^y\|_2^2 \longrightarrow \min_{\mathbf{C}^x,\mathbf{C}^y} \tag{9a}$$

$$\text{s.t. } \left((\mathcal{N}_\lambda \mathbf{C}^x)^\top, (\mathcal{N}_\lambda \mathbf{C}^y)^\top\right)^\top = (\mathbf{u}_\lambda^\top, \mathbf{v}_\lambda^\top)^\top\,. \tag{9b}$$

Problem (9) is a quadratic optimisation problem with linear constraints. The corresponding optimality system is given by

$$\begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathcal{N}_\lambda^c & (\mathcal{N}_\lambda)^\top \\ \mathcal{N}_\lambda & 0 \end{pmatrix} \begin{pmatrix} \mathbf{C}^x \\ \boldsymbol{\lambda}^x \end{pmatrix} = \begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathbf{u}_\lambda^c \\ \mathbf{u}_\lambda \end{pmatrix} \text{ and} \tag{10a}$$

$$\begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathcal{N}_\lambda^c & (\mathcal{N}_\lambda)^\top \\ \mathcal{N}_\lambda & 0 \end{pmatrix} \begin{pmatrix} \mathbf{C}^y \\ \boldsymbol{\lambda}^y \end{pmatrix} = \begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathbf{v}_\lambda^c \\ \mathbf{v}_\lambda \end{pmatrix}. \tag{10b}$$

We consider one of the two uncoupled equations and define the saddle point matrix

$$\mathcal{A} = \begin{pmatrix} (\mathcal{N}_\lambda^c)^\top \mathcal{N}_\lambda^c & (\mathcal{N}_\lambda)^\top \\ \mathcal{N}_\lambda & 0 \end{pmatrix}$$

and denote the upper left block of it by $A = (\mathcal{N}_\lambda^{\mathrm{c}})^\top \mathcal{N}_\lambda^{\mathrm{c}}$. Note that under the previous assumption that $\mathcal{N}(\boldsymbol{\xi})$ has maximal rank also $\mathcal{N}_\lambda^{\mathrm{c}}(\boldsymbol{\xi})$ is be of maximal rank and hence A is be regular. With $\mathcal{N}_\lambda$ being surjective, also the saddle point matrix $\mathcal{A}$ is regular, see [34, Theorem 3.2.1]. We use the Schur complement of the saddle point matrix $\mathcal{A}$ given by

$$S = -\mathcal{N}_\lambda A^{-1} (\mathcal{N}_\lambda)^\top$$

to express the solutions $\mathbf{C}^x$ and $\mathbf{C}^y$ of (10) as

$$\mathbf{C}^x = \big(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathcal{N}_\lambda\big) A^{-1} (\mathcal{N}_\lambda^{\mathrm{c}})^\top \mathbf{u}_\lambda^{\mathrm{c}} - A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathbf{u}_\lambda, \tag{11a}$$

and

$$\mathbf{C}^y = \big(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathcal{N}_\lambda\big) A^{-1} (\mathcal{N}_\lambda^{\mathrm{c}})^\top \mathbf{v}_\lambda^{\mathrm{c}} - A^{-1}(\mathcal{N}_\lambda)^\top S^{-1} \mathbf{v}_\lambda\,, \tag{11b}$$

where $I_J \in \mathbb{R}^{J\times J}$ denotes the identity matrix. We here only give the solution formula for the optimal control point vectors and not for the associated multipliers $\boldsymbol{\lambda}^x$ and $\boldsymbol{\lambda}^y$. For details about the Langrange multipliers see [35].

With (11), we can eliminate the control points as optimisation variables and can define the reduced functional which depends only on the parameter vector $\boldsymbol{\xi} = (\xi_i)_{i\in\mathcal{I}_\lambda^{\mathrm{c}}}$ analogous to (4). Inserting (11) into (9a) we face the problem of minimising

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

with respect to $\boldsymbol{\xi}$ where

$$\begin{aligned}\mathbf{r}_\lambda^x(\boldsymbol{\xi}) = &\Big(I_{|\mathcal{I}_\lambda^{\mathrm{c}}|} - \mathcal{N}_\lambda^{\mathrm{c}}\big(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1}\mathcal{N}_\lambda\big) A^{-1}(\mathcal{N}_\lambda^{\mathrm{c}})^\top\Big)\mathbf{u}_\lambda^{\mathrm{c}} \\ &- \mathcal{N}_\lambda^{\mathrm{c}} A^{-1}(\mathcal{N}_\lambda)^\top S^{-1}\mathbf{u}_\lambda.\end{aligned} \tag{12}$$

The corresponding expression for $\mathbf{r}_\lambda^y(\boldsymbol{\xi})$ is defined analogously. The matrices $\mathcal{N}_\lambda^{\mathrm{c}}$ 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 (11) and show that Lemma 1 can be applied to calculate the Jacobian of certain terms occurring in the residual term (12). We present the calculation for (11a), as (11b) follows analogously. Note that we can rewrite the operator from (12) to which we would like to apply Lemma 1 in the following form

$$\begin{aligned}\mathcal{N}_\lambda^{\mathrm{c}}\big(I_J + A^{-1}(\mathcal{N}_\lambda)^\top S^{-1}\mathcal{N}_\lambda\big) A^{-1}(\mathcal{N}_\lambda^{\mathrm{c}})^\top = \\ \mathcal{N}_\lambda^{\mathrm{c}}(\mathcal{N}_\lambda^{\mathrm{c}})^+ - \mathcal{N}_\lambda^{\mathrm{c}} A^{-1}(\mathcal{N}_\lambda)^\top \big(\mathcal{N}_\lambda^{\mathrm{c}} A^{-1}(\mathcal{N}_\lambda)^\top\big)^+.\end{aligned} \tag{13}$$

Here the superscript $^+$ denotes the Moore Penrose inverse. As a preliminary step we show that (13) is actually a projection:

Lemma 2. *The linear map*

$$P_V = \mathcal{N}_\lambda^{\mathrm{c}}(\mathcal{N}_\lambda^{\mathrm{c}})^+ - \mathcal{N}_\lambda^{\mathrm{c}} A^{-1}(\mathcal{N}_\lambda)^\top \big(\mathcal{N}_\lambda^{\mathrm{c}} A^{-1}(\mathcal{N}_\lambda)^\top\big)^+$$

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

$$\begin{aligned} V &= \{\mathbf{v} = \mathcal{N}_\lambda^c \mathbf{u} \in \mathrm{rg}(\mathcal{N}_\lambda^c) \,:\, \mathcal{N}_\lambda \mathbf{u} = 0\} \\ &= \mathrm{rg}\Big(\mathcal{N}_\lambda^c|_{\ker(\mathcal{N}_\lambda)}\Big). \end{aligned}$$

Proof. A direct calculation shows that P_V as defined above is idempotent, i.e. $P_V^2 = P_V$, and symmetric. It therefore is the orthogonal projection in $\mathbb{R}^{|\mathcal{I}_\lambda^c|}$ onto its range $\mathrm{rg}(P_V)$. Obviously $\mathrm{rg}(P_V) \subset \mathrm{rg}(\mathcal{N}_\lambda^c)$ holds. Moreover, with

$$\mathbf{u} = (\mathcal{N}_\lambda^c)^+ \mathbf{w} - A^{-1}(\mathcal{N}_\lambda)^\top \big(\mathcal{N}_\lambda^c A^{-1}(\mathcal{N}_\lambda)^\top\big)^+ \mathbf{w}\,, \mathbf{w} \in \mathbb{R}^{|\mathcal{I}_\lambda^c|}\,,$$

we have $\mathbf{v} = \mathcal{N}_\lambda^c \mathbf{u} \in \mathrm{rg}(P_V)$ and

$$\begin{aligned} \mathcal{N}_\lambda \mathbf{u} &= \mathcal{N}_\lambda (\mathcal{N}_\lambda^c)^+ \mathbf{w} - \mathcal{N}_\lambda A^{-1}(\mathcal{N}_\lambda)^\top \big(\mathcal{N}_\lambda^c A^{-1}(\mathcal{N}_\lambda)^\top\big)^+ \mathbf{w} \\ &= \mathcal{N}_\lambda A^{-1}(\mathcal{N}_\lambda^c)^\top \mathbf{w} - \mathcal{N}_\lambda A^{-1}\mathcal{N}_\lambda^\top \big(\mathcal{N}_\lambda A^{-1} A A^{-1} \mathcal{N}_\lambda^\top\big)^{-1} \mathcal{N}_\lambda A^{-1}(\mathcal{N}_\lambda^c)^\top \mathbf{w} \\ &= \mathcal{N}_\lambda A^{-1}(\mathcal{N}_\lambda^c)^\top \mathbf{w} - \mathcal{N}_\lambda A^{-1}(\mathcal{N}_\lambda^c)^\top \mathbf{w} = 0. \end{aligned}$$

Thus, we get $\mathrm{rg}(P_V) \subset V$.

On the other hand let $\mathbf{v} \in V$. We show that then $P_V \mathbf{v} = \mathbf{v}$ holds. With $\mathbf{v} = \mathcal{N}_\lambda^c \mathbf{u}$ and $\mathcal{N}_\lambda \mathbf{u} = 0$, we get $\mathcal{N}_\lambda^c (\mathcal{N}_\lambda^c)^+ \mathbf{v} = \mathcal{N}_\lambda^c (\mathcal{N}_\lambda^c)^+ \mathcal{N}_\lambda^c \mathbf{u} = \mathcal{N}_\lambda^c \mathbf{u} = \mathbf{v}$ by the properties of the Moore Penrose inverse $(\mathcal{N}_\lambda^c)^+$. On the other hand, we have

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

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

We 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 [30]. Note that $\mathcal{N}_\lambda$ has linearly independent rows by assumption, so the Moore Penrose inverse is given as $(\mathcal{N}_\lambda)^+ = \mathcal{N}_\lambda^\top (\mathcal{N}_\lambda \mathcal{N}_\lambda^\top)^{-1}$. We want to treat the projection P_V onto V within the framework described in Lemma (1) for this we define the matrix B

$$B = \mathcal{N}_\lambda^c \left(I - \mathcal{N}_\lambda^+ \mathcal{N}_\lambda\right) \in \mathbb{R}^{|\mathcal{I}_\lambda^c| \times J}. \tag{14}$$

and we obtain

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

We now investigate this matrix further to subseqently apply Lemma 1, in order to evaluate the derivative of the operator in (12). We set

$$B^- = (\mathcal{N}_\lambda^c)^+ - A^{-1}\mathcal{N}_\lambda^\top S^{-1} \mathcal{N}_\lambda (\mathcal{N}_\lambda^c)^+. \tag{15}$$

We show that B^- is a generalised inverse for B satisfying the two conditions $BB^-B = B$ and $BB^- = (BB^-)^\top$, which are necessary for the application of Lemma 1. Using the relations $(\mathcal{N}_\lambda^{\mathrm{c}})^+\mathcal{N}_\lambda^{\mathrm{c}} = I$ and $\mathcal{N}_\lambda\mathcal{N}_\lambda^+\mathcal{N}_\lambda = \mathcal{N}_\lambda$ we see

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

and

$$BB^-B = \mathcal{N}_\lambda^{\mathrm{c}}(I - \mathcal{N}_\lambda^+\mathcal{N}_\lambda)^2 = \mathcal{N}_\lambda^{\mathrm{c}}(I - \mathcal{N}_\lambda^+\mathcal{N}_\lambda) = B,$$

as $I - \mathcal{N}_\lambda^+\mathcal{N}_\lambda$ is a projection. Moreover, a direct computation shows that $BB^- = P_V$ and therefore symmetric.

With the definition (14) and the corresponding generalised inverse B^- in (15), the conditions of Lemma 1 are satisfied. Applying the lemma to $P_{\mathrm{rg}(B)^\perp} = (I - BB^-)$ then yields

$$\frac{\partial P_{(\mathrm{rg}B)^\perp}}{\partial \xi_i} = -P_{(\mathrm{rg}B)^\perp}\frac{\partial B}{\partial \xi_i}B^- - \left(P_{(\mathrm{rg}B)^\perp}\frac{\partial B}{\partial \xi_i}B^-\right)^\top. \tag{16}$$

Note that $I - BB^-$ is exactly the matrix in the first expression on the right hand side of(12). To calculate the Jacobian for the Gauss-Newton scheme, the first term in the residuum 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 variable and therefore their derivatives are not considered in the computation of the Jacobian. For $i \in \mathcal{I}_\lambda^{\mathrm{c}}$, the derivative $\frac{\partial}{\partial \xi_i}\mathcal{N}_\lambda = 0$ because $\mathcal{N}_\lambda$ does only depend on the interpolating parameter values in $\mathcal{I}_\lambda$. Therefore, we get

$$\frac{\partial B}{\partial \xi_i} = \frac{\partial \mathcal{N}_\lambda^{\mathrm{c}}}{\partial \xi_i}(I - \mathcal{N}_\lambda^+\mathcal{N}_\lambda) \tag{17}$$

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

The term remaining in equation (12) after subtracting (13) is $\mathcal{N}_\lambda^{\mathrm{c}}A^{-1}\mathcal{N}_\lambda^\top S^{-1}$. Its partial derivatives can be computed using

$$\frac{\partial}{\partial \xi}(L^{-1}) = -L^{-1}\frac{\partial L}{\partial \xi}L^{-1}$$

for the derivative of the inverse of a regular matrix L with respect to a parameter ξ.

Example of a constrained fit. The significance of precisely interpolating corners as well as the end points is easily seen with an example. In Figure 4 a patch boundary with two kinks is fitted with and without the additional constraint. We clearly see an improved fit of the corners once the constraint is used, resulting in a better overall fit.

3.3.2. Control point distances penalty

A major problem occurring in curve fitting is that sometimes control points tend to oscillate or accumulate. This can be avoided by penalizing the distance

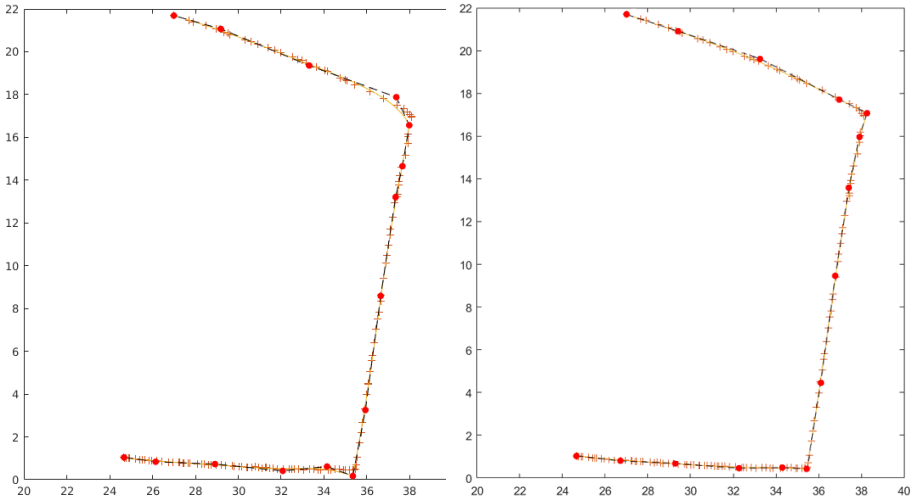


Figure 4: Left: spline fit without constraints, right: spline fit with end and corner point constraints. The red bullets indicate the control points of the fitted spline (which is visualised by a yellow line). The data points to be approximated are indicated by red crosses.

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

$$p_1\left(\boldsymbol{C}^x, \boldsymbol{C}^y, \alpha\right) = \frac{\alpha}{L} \sum_{j=2}^{J} \left\| \boldsymbol{\Delta}_j - \overline{\boldsymbol{\Delta}}_j \right\|^2 . \tag{18}$$

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 think here about an eligible spacing, the simplest one — though not always effective and applicable — would be a uniform distribution. A usual value for α is from the range $[0, 10]$.

Figure 5 illustrates the relevance of this regulariser. For this strongly curved outline we get oscillating control points. While the distance between the data points and the curve is minimal, the curve is clearly overfitting and the distance between the exact outline and the curve is too large. The regularization term p_1 can prevent this undesirable behaviour in this example and yields a curve that matches the exact outline.

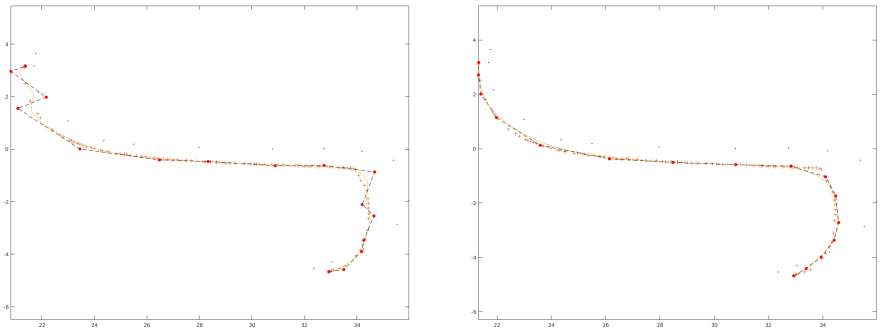


Figure 5: Left: spline fit without penalizing the distance of the control points, right: spline fit with penalizing the distance of the control points via the term $p_1(\boldsymbol{C}^x, \boldsymbol{C}^y, \alpha)$ in (18) with $\alpha = 4$.

3.3.3. Control point miss-match penalty

The regularisation term $p_1(\boldsymbol{C}^x, \boldsymbol{C}^y, \alpha)$ reduced 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 as we noted that the joint use of two regularisation terms with smaller parameters results in an improved fit. An example is shown in Figure 6a and 6b , where the original data fit without any additional penalties yields a self-intersecting curve. Applying the regularisation $p_1(\boldsymbol{C}^x, \boldsymbol{C}^y, \alpha)$ results in a smooth curve, which however significantly lost the accuracy of the approximation.

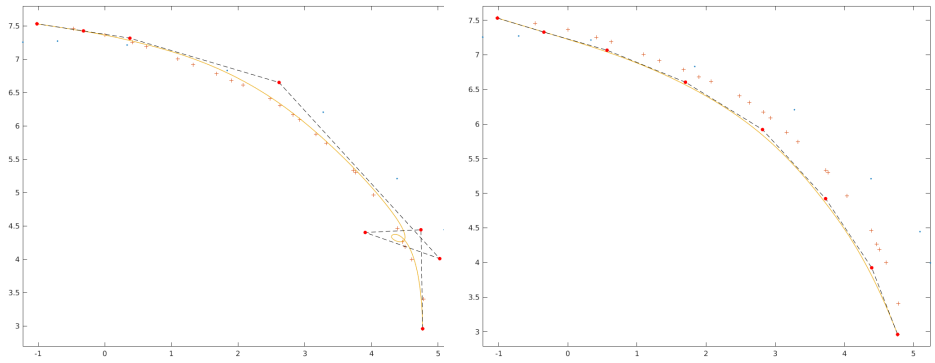
The second regularisation term forces the control points towards a suitable set of reference control points. This also prevents the self-intersection of the curve and allows us to use a smaller regularisation constant, yielding a better approximation. The reference control points for this can either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

$$p_2\left(\boldsymbol{C}^x, \boldsymbol{C}^y, \beta\right) = \beta \sum_{j=1}^{J} \left\| \mathbf{C_j^{ref}} - \mathbf{C_j} \right\|_2^2 . \tag{19}$$

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/size. The improved precision of the geometry with both regularisation terms can be seen in Figure 6c.

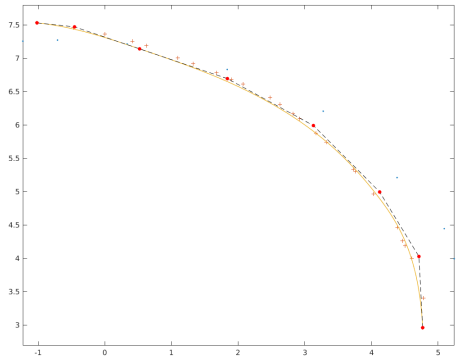
Figure 6 shows an example where the error of the data fit gets significantly reduced by applying this regulariser such that the control points are forced into the direction of the data points.

In summary we suggest to minimise the modified least squares problem which



(a) Spline-fit with no regularization

(b) Spline fit, regularised using only p_1 with $\alpha = 10$.



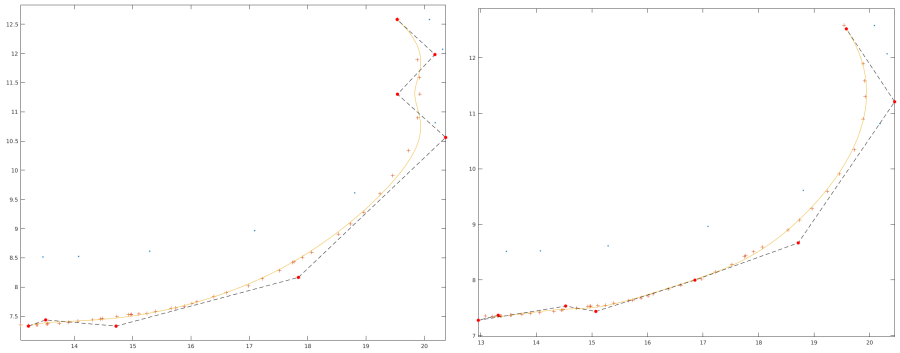
(c) Spline fit, regularised using both terms, p_1 and p_2, with $\alpha = 7.5, \beta = 0.1$.

Figure 6: Comparison of the fitted spline curves with different regularization terms applied.

we get by adding the terms p_1, p_2 to equation (9a). In the same manner as before, inserting the solution for the control points which we get for fixed $\boldsymbol{\xi}$ then leads again to a (now modified) residual term $\tilde{\boldsymbol{r}}_\lambda$. The optimality system stays nearly the same, just some obvious modifications in (10), a unit matrix for the term $p_1\left(\boldsymbol{C}^x, \boldsymbol{C}^y, \alpha\right)$, and a unit block diagonal matrix for the term $p_2\left(\boldsymbol{C}^x, \boldsymbol{C}^y, \beta\right)$ need to be inserted and the corresponding necessary changes in the Jacobian matrix $J_{\tilde{\mathbf{r}}_\lambda}\left(\boldsymbol{\xi}\right)$ (insertion of additional zero lines) have to be done.

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 parameterization [36]. Both versions needed a comparable number of iterations to converge to a steady state, leading between 10 to 15 iterations per spline fit, depending on the complexity of the geometry of the underlying data. With this approach we did not experience any

case of non-convergence. Figure 7 illustrates the benefit of the optimisation of the parameter vector $\boldsymbol{\xi}$. 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 Figure 7 (a) and (b)). Even though a partial optimisation of $\boldsymbol{\xi}$ 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.



(a) (b)

Figure 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 smoothens the spline curve.

We conclude with Figure 8, where one can see the positive influence of the regularisation terms on the rate of convergence for one of the boundary splines. The rate of convergence 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 Coon's patch [29], using the

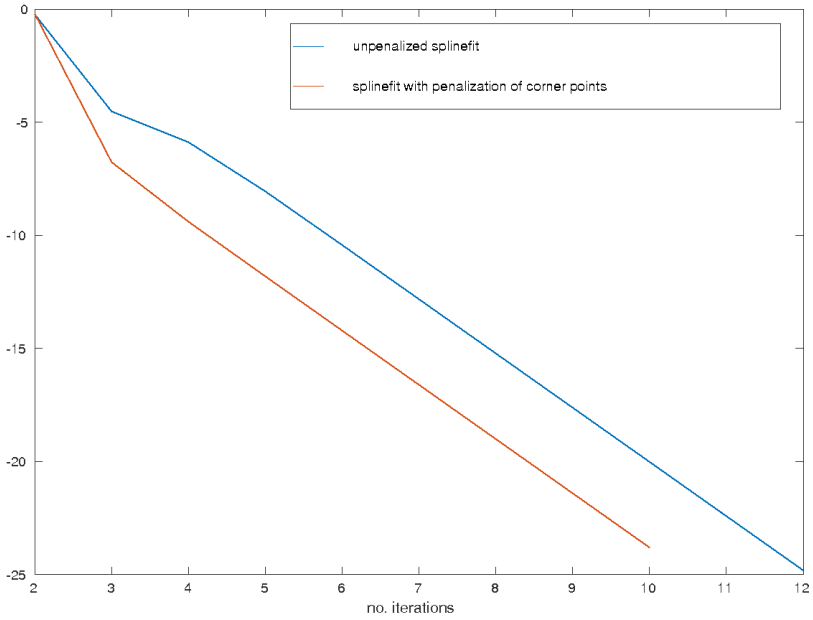


Figure 8: estimated error for the spline fit of the example illustrated in Figure 9, Section 5. The blue line shows the evolution of the estimated error of the optimiser without any regularization terms, the orange line shows the improved evolution of the estimated error when we interpolate corner points.

Matlab implementation of the NURBS toolbox [37]. When using Coon's patch for complex geometries, it can happen that a part of the resulting surface lies outside the boundary curves. Due to the suitable decomposition into 16 subdomains, we did not encounter such problems. However, in some cases Coon's patch yielded an irregular parametrisation of the interior control points, which required a manual correction.

Due to the presence of 16 interfaces, at which neighboring 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 neighboring patches. With this consecutive procedure of inheriting interface splines from one patch to the other we get a continuous multi-patch domain and avoid modification of the fitted spline basis like it would be necessary when trimming NURBS surfaces [18, 19].

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

$$S(\xi,\eta) = \sum_{j=1}^{J}\sum_{k=1}^{K} N_j(\xi)\cdot N_k(\eta)\cdot \begin{pmatrix} C^x_{j,k} \\ C^y_{j,k} \end{pmatrix} \tag{20}$$

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, therefore requires no data fitting. The front size has the same outline in the x-y-plane, but 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^z_{jkL} of the control points $\mathbf{C}_{jkL} = (C^x_{jkL}, C^y_{jkL}, C^z_{jkL})^\top$ 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

$$\min_{C^z_{jkL}} \sum_{i=1}^{I} \left(\sum_{j=1}^{J} \sum_{k=1}^{K} N_j(\xi_i) N_k(\eta_i) \cdot C^z_{jkL} - w_i \right)^2 \tag{21}$$

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 (20) at $\{u_i, v_i\}$ using a Gauss-Newton 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 (21) 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

$$\begin{aligned} \min_{C^z_{jkl}} \; & \|S(\cdot,\cdot;C^z)\|^2_{L^2} + \alpha \, \|\nabla_{(x,y)} S(\cdot,\cdot;C^z)\|^2_{L^2}, \\ & \text{s.t. } S(\xi_i, \eta_i; C^z) \geq w_i, \quad i = 1, \dots I, \end{aligned}$$

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 Figure 2 for the numbering of the patches. For all plots in this chapter the lengths are measured in millimeters.

In Figure 9 to Figure 10 one can see the different steps of the surface fit examplified for patch 1. First, Figure 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).

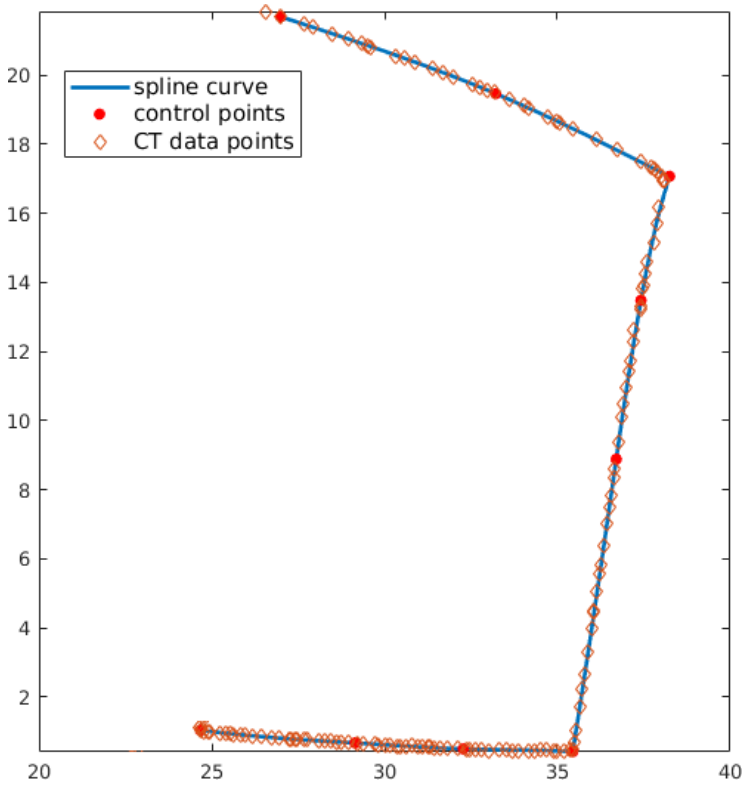


Figure 9: The blue line show the spline fit for v_1-curve of patch 1 incorporating the characteristic properties of the data points (orange diamonds) like corner points.

Figure 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 are showing the data points of the measured μ-CT data superposing with the patch boundary. As one can see we benefit here from the preparations we took during the construction of the boundary splines. Despite of the complicated curvature of the patch we do not get any intersecting splines inside the surface.

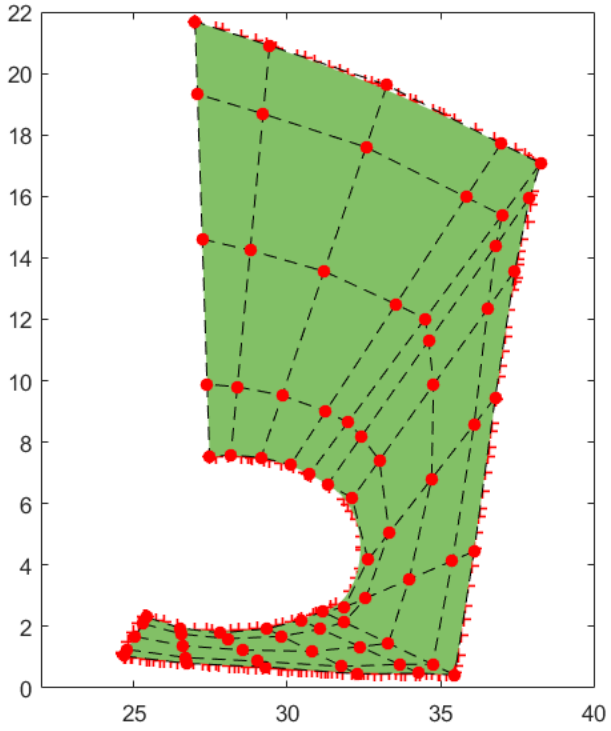


Figure 10: Spline surface fitted to the data of patch 1. The red bullets visualize the control net of the patch.

In Figures 11, one can see how the final multi-patch surface geometry of the complete violin bridge looks like 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, Figure 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 by the different colors of the data points and one can see that the vast majority of the data points got approximated very well.

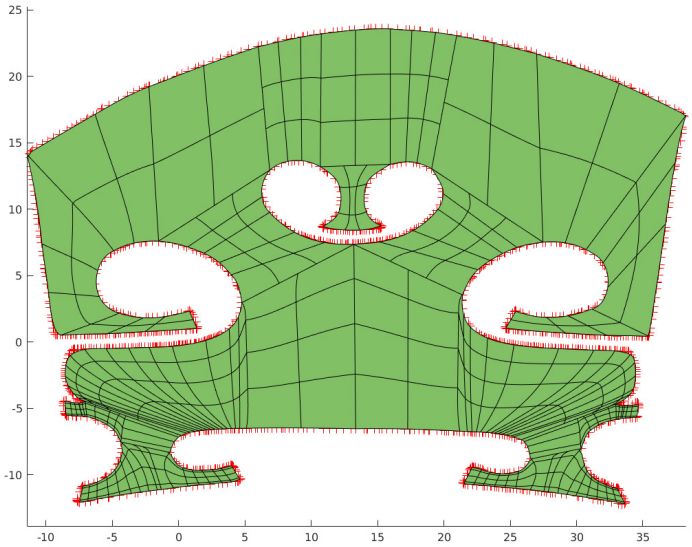


Figure 11: The complete fitted 2D surface-geometry. The data points are indicated by orange crosses.

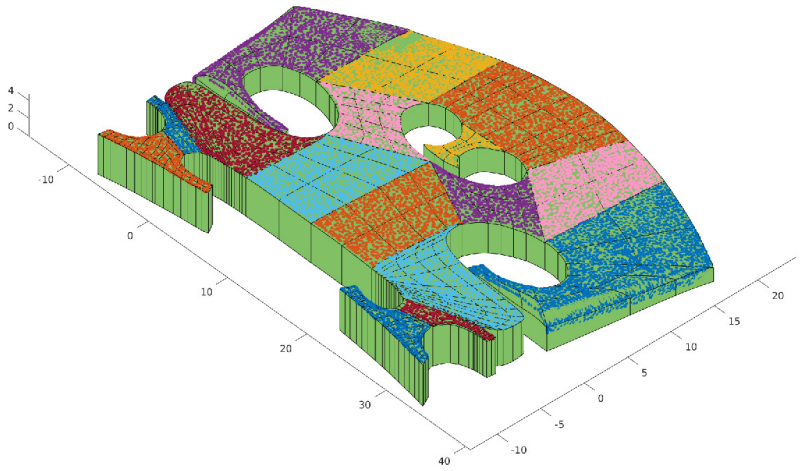


Figure 12: The 3D geometry fitted to the data points. The data points are colored patch-wise.

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 [21, 38]. On the other hand, the influences of certain geometric modifications onto the filter properties of the bridge are 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:

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

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 $\boldsymbol{\varepsilon}(\mathbf{u}) = (\nabla\mathbf{u} + \nabla\mathbf{u}^\top)/2$ 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 [39, Chapter 2.4].

The idea of isogeometric analysis [24, 40] 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 [2, 28] to discretize the fitted geometry. The trivariate 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. $\gamma_i = \partial\Omega_s \cap \partial\Omega_m$) and 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 [2].

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

$$a(\mathbf{u}_h,\mathbf{v}_h) + b(\mathbf{v}_h,\boldsymbol{\mu}_h) = \lambda_h\, m(\mathbf{u}_h,\mathbf{v}_h), \quad \mathbf{v}_h \in \mathbf{V}_h,$$
$$b(\mathbf{u}_h,\boldsymbol{\tau}_h) = 0, \quad \boldsymbol{\tau}_h \in \mathbf{M}_h.$$

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., $\lambda_h = \omega_h^2$. Figure 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.

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

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

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

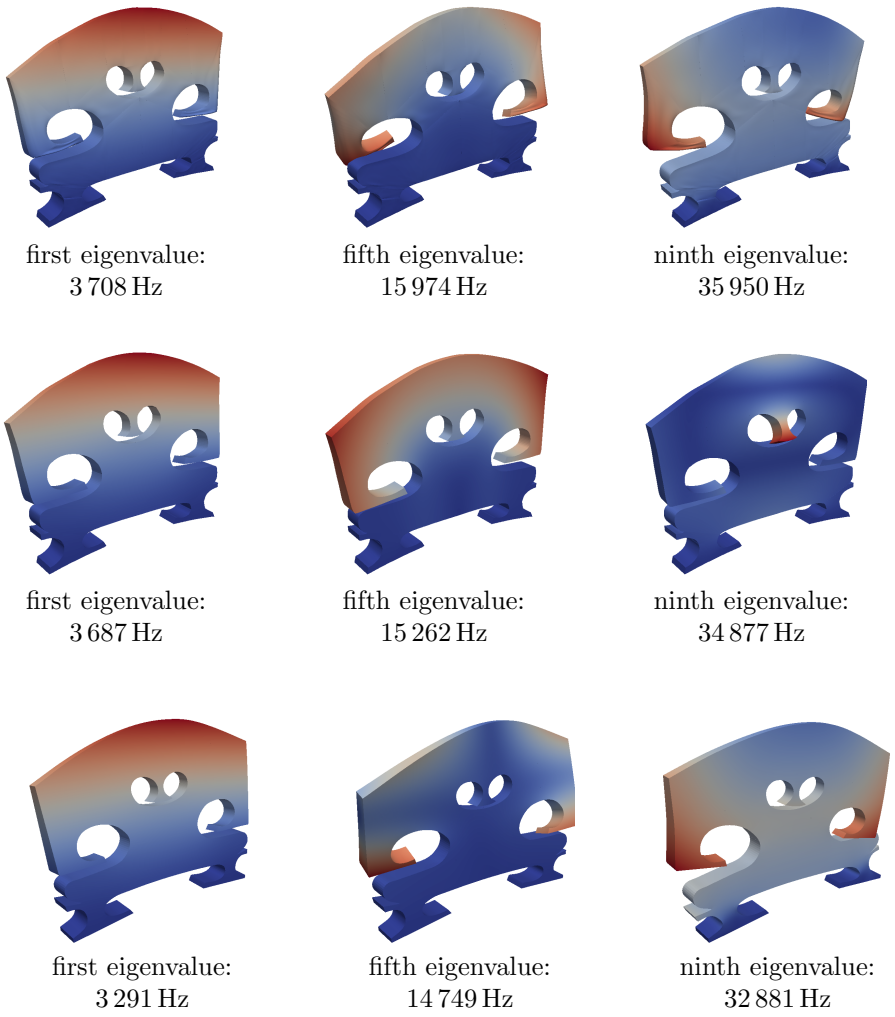


Figure 13: Visualization of the three eigenmodes with the associated eigenvalue. Top row: three-dimensional spline fit; Middle row: fitted outline with a linear thickness; Bottom row: reference geometry. All geometries share the same size.

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 with respect to the geometry.

Acknowledgement

We gratefully acknowledge the support by the International Research Training Group IGDK 1754 Optimization and Numerical Analysis for Partial Differential Equations with Nonsmooth Structures, funded by the German Research Council (DFG) and the Austrian Science Fund (FWF):[W 1244-N18].

The authors thank Markus Muhr and Barbara Wohlmuth for helpful suggestions and fruitful discussions and Andrew Finnigan and Pia Klaembt for providing us with an excelent violin bridge for the μ-CT scan.

References

[1] C. F. Borges, T. Pastva, Total least squares fitting of Bzier and B-spline curves to ordered data, Computer Aided Geometric Design 19 (4) (2002) 275–280.

[2] E. Brivadis, A. Buffa, B. Wohlmuth, L. Wunderlich, Isogeometric mortar methods, Computer Methods in Applied Mechanics and Engineering 284 (2015) 292–319.

[3] L. Schumaker, Spline Functions: Basic Theory, 3rd Edition, Cambridge University Press, Cambridge, 2007.

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

[5] L. Piegl, W. Tiller, The NURBS Book, Springer, 1997.

[6] Y. Zhang, J. Cao, Z. Chen, X. Li, X.-M. Zeng, B-spline surface fitting with knot position optimization, Computers & Graphics 58 (2016) 73 – 83.

[7] V. Weiss, L. Andor, G. Renner, T. Varady, Advanced surface fitting techniques, Computer Aided Geometric Design 19 (1) (2002) 19 – 42.

[8] P. Bo, R. Ling, W. Wang, A revisit to fitting parametric surfaces to point clouds, Computers & Graphics 36 (5) (2012) 534 – 540, shape Modeling International (SMI) Conference 2012.

[9] J. Carnicer, J. Delgado, J. Peña, Progressive iteration approximation and the geometric algorithm, Computer-Aided Design 44 (2) (2012) 143 – 145.

[10] C. Deng, H. Lin, Progressive and iterative approximation for least squares B-spline curve and surface fitting, Computer-Aided Design 47 (2014) 32 – 44.

[11] M. Liu, B. Li, Q. Guo, C. Zhu, P. Hu, Y. Shao, Progressive iterative approximation for regularized least square bivariate B-spline surface fitting, Journal of Computational and Applied Mathematics 327 (2018) 175 – 187.

[12] A. Gàlvez, A. Iglesias, J. Puig-Pey, Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction, Information Sciences 182 (1) (2012) 56 – 76.

[13] A. N. Ravari, H. D. Taghirad, Reconstruction of B-spline curves and surfaces by adaptive group testing, Computer-Aided Design 74 (2016) 32 – 44.

[14] O. Grove, K. Rajab, L. A. Piegl, From CT to NURBS: Contour fitting with B-spline curves, Computer-Aided Design and Applications 7 (sup1) (2010) 1–19.

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

[16] V. Patraucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, C. Haas, State of research in automatic as-built modelling, Advanced Engineering Informatics 29 (2) (2015) 162 – 171.

[17] A. Seiler, D. Gromann, B. Jttler, Spline surface fitting using normal data and norm-like functions, Computer Aided Geometric Design 64 (2018) 37 – 49.

[18] R. Schmidt, R. Wchner, K.-U. Bletzinger, Isogeometric analysis of trimmed nurbs geometries, Computer Methods in Applied Mechanics and Engineering 241-244 (2012) 93 – 111.

[19] H.-J. Kim, Y.-D. Seo, S.-K. Youn, Isogeometric analysis with trimming technique for problems of arbitrary complex topology, Computer Methods in Applied Mechanics and Engineering 199 (45) (2010) 2796 – 2812.

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

[21] J. Woodhouse, On the "bridge hill" of the violin, Acta Acustica united with Acustica 91 (1) (2005) 155–165.

[22] A. C. Kak, S. M., Principles of Computerized Tomographic Imaging, IEEE Press, New York, 1988.

[23] T. Buzug, Computed Tomography. From Photon Statisticsto Modern Cone-Beam CT, Springer-Verlag, Heidelberg, 2008.

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

[25] L. Beirão Da Veiga, A. Buffa, G. Sangalli, R. Vásquez, Mathematical analysis of variational isogeometric methods, Acta Numerica 23 (2014) 157–287.

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

[27] J. Rueckel, M. Stockmar, F. Pfeiffer, J. Herzen, Spatial resolution characterization of a X-ray microCT system, Applied Radiation and Isotopes 94 (2014) 230–234.

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

[29] S. A. Coons, Surfaces for computer-aided design of space forms, Tech. rep., Massachusetts Institute of Technology (1967).

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

[31] A. Ebrahimi, G. Barid Loghmani, Shape modeling based on specifying the initial B-spline curve and scaled BFGS optimization method, Multimedia Tools and Applications 77 (23) (2018) 30331–30351.

[32] G. Golub, V. Pereyra, The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, SIAM Journal on Numerical Analysis 10 (1973) 413–432.

[33] G. H. Golub, C. F. Van Loan, Matrix Computations, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.

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

[35] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.

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

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

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

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

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