Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 369 (2020) 113227

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Towards untrimmed NURBS: CAD embedded reparameterization of trimmed B-rep geometry using frame-field guided global parameterization

René R. Hiemstra[a,*], Kendrick M. Shepherd[b], Michael J. Johnson[b], Lulin Quan[b], Thomas J.R. Hughes[b]

[a] *Institut für Baumechanik und Numerische Mechanik (IBNM), Leibniz Universität Hannover, Germany*
[b] *Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, United States of America*

Received 10 March 2020; accepted 18 June 2020
Available online xxxx

Abstract

The boundary representation or B-rep is the prevalent geometry description in computer aided design (CAD). It combines aspects from explicit and implicit geometry and is incompatible with many downstream applications such as finite element analysis (FEA) and computer aided manufacturing (CAM). We develop a semi-automatic frame-field guided parameterization approach that converts trimmed B-rep geometry to conforming, watertight analysis-suitable NURBS. The resulting geometry description is simultaneously suitable for design as well as analysis and therefore avoids the need for further meshing and geometry-processing. While existing frame-field based global parameterization methods offer the high quality necessary in our application, they suffer from several robustness issues. Current challenges mainly stem from the lack of integrability of the frame-field which causes the computed parameterization to fold-over, particularly near parameterization singularities. We propose specialized constraints that incorporate properties of an analytical solution that resolve the poor behavior near singularities. Furthermore, we present a convenient and efficient solution framework that directly incorporates such constraints into a reduced basis. Improved behavior around parameterization singularities simplifies the extraction of a quadrilateral layout. This layout is subsequently fitted with C^0 NURBS to yield a conforming watertight model that is analysis suitable. The latter is illustrated by performing isogeometric shell modal analysis on several geometric models.

© 2020 Elsevier B.V. All rights reserved.

Keywords: Isogeometric analysis; Model generation; Meshing; Parameterization; Watertight splines

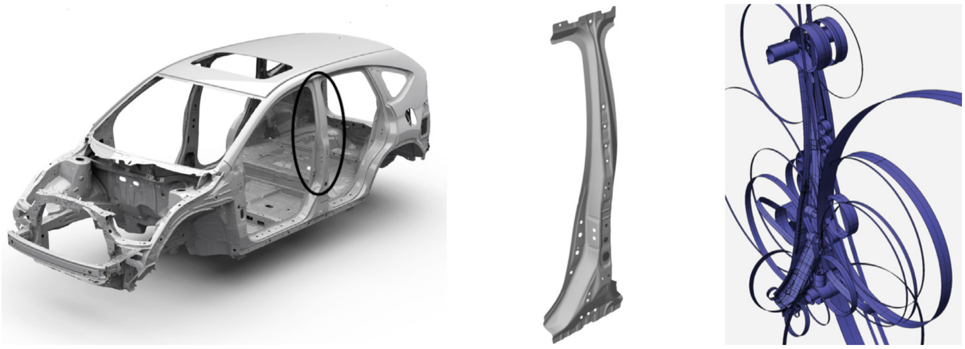
1. Introduction

Originally computer aided design (CAD) was envisioned as an interactive environment to support the product development process as a whole [1,2]. This holistic view included the conceptual design phase, in which shape and form are sought to support design objectives, all the way through to detailed design, with frequent intermediate

[*] Corresponding author.

E-mail addresses: rene.hiemstra@ibnm.uni-hannover.de (R.R. Hiemstra), kendrick@ices.utexas.edu (K.M. Shepherd), mike@ices.utexas.edu (M.J. Johnson), quanlulin@gmail.com (L. Quan), hughes@ices.utexas.edu (T.J.R. Hughes).

https://doi.org/10.1016/j.cma.2020.113227
0045-7825/© 2020 Elsevier B.V. All rights reserved.



(a) CAD model car body (b) Visual representation B-pillar (c) Brep data structure B-pillar

Fig. 1. The mathematical representation in CAD of a structural component of a car, the B-pillar, consist of 1280 trimmed NURBS surfaces that do not form a single conforming or watertight model. (Urick, B. Private communication. 2018).

recourse to engineering analysis. Although CAD has in many ways been very successful, it has fallen short of its original intent to serve as an interactive environment for efficient product development. A complete industry exists nowadays to support a "leaky" design-through-analysis pipeline. According to a report of Sandia National Laboratories in 2005 [3], the time spent creating analysis suitable geometry (which includes many steps, such as de-featuring, geometry decomposition, meshing, and mesh repair) completely dominates the time spent performing analysis. In recent years this trend has worsened because advances in computing power and efficiency of numerical solvers have cut down the analysis times while not much progress has been made to reduce the time spent on geometry pre-processing.

Isogeometric analysis was introduced by Hughes et al. [4] to improve the interoperability between computer aided design and finite element analysis by unifying the underlying technology that supports both disciplines. Its promise is to eliminate the tedious process of geometry repair, feature removal, and mesh generation, while, at the same time, maintaining a single exact representation of geometry at all stages of the design process. Although isogeometric analysis has proven its fidelity as an analysis technology, some key issues, inherited from traditional CAD, have remained.

The prevalent geometry description used in CAD is the so called boundary representation or B-rep. As the name suggests, geometry is represented by its bounding surfaces. There are two major bottlenecks that impede the design-through-analysis process:

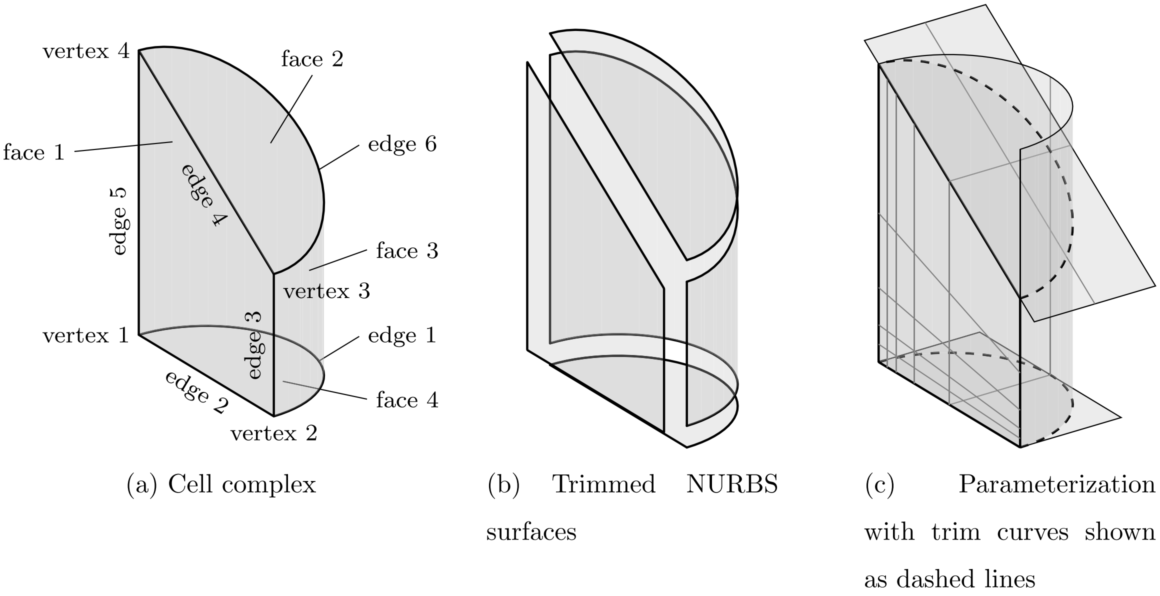
- Apart from shell or boundary element analysis, in general, a three dimensional volumetric parameterization is required to perform finite element analysis. Hence, one challenge is to generate a volumetric description from a boundary representation.
- Boolean operations, ubiquitous in computer-aided design, use a process called trimming that leads to a non-conforming description of geometry that is largely un-editable and incompatible with downstream applications such as finite element analysis, optimization and manufacturing.

The former is commonly referred to as the *volumetric problem*, while the latter as the *surface problem*.

In this work we focus on the surface problem. An example illustrating the severity of the surface problem is depicted in Fig. 1. Complex CAD models are made out of thousands of NURBS patches that are plagued by "dirty geometry", such as slivers, gaps and overlaps. Such a geometry representation is useless in downstream applications such as finite element analysis (FEA) and computer aided manufacturing (CAM) without thorough geometry repair and cleanup.

1.1. Limitations of the boundary representation

A B-rep is defined by a topological structure called a cell complex, a collection of NURBS surfaces and a collection of trim curves, see Fig. 2. The trim curves serve to divide the surfaces into their visible and masked



(a) Cell complex

(b) Trimmed NURBS surfaces

(c) Parameterization with trim curves shown as dashed lines

Fig. 2. Boundary representation (B-rep) of a CAD solid model [5].

regions. The cell complex prescribes how the collection of NURBS surfaces are glued together to form an open or closed poly-surface.

The trim curves of the boundary representation generally do not align with the parametric directions of the surface representation and thus are only represented approximately (and disparately) in both model and parametric space. The B-rep paradigm combines aspects of explicit parametric geometry with implicit geometry, which leads to several problems [5–7]:

1. Many surface operations are prohibited after a Boolean operation (i.e. a subtraction, intersection, or union) has been performed. The reason for this is that an edit could potentially change the topology of the model, which is not allowed by most CAD systems.
2. Major geometric features are introduced that are not aligned with the parametric directions of the underlying surface patches. This can lead to loss of shape control near trim curves for those surface operations that are permitted by the CAD system.
3. Trim curves may be represented slightly differently depending on the geometry kernel of the CAD system. This can lead to loss of accuracy, or worse, to undesirable surface artifacts, like gaps and overlaps, when converting to a common exchange format such as STEP.
4. Trim often leads to non-conforming geometric models that have undesirable gaps and overlaps that pollute downstream processes.

It turns out that trim is not only detrimental for the interoperability with downstream applications, such as finite element analysis and computer aided manufacturing, but it also constitutes a major impediment for computer aided design itself! For these reasons, the Handbook of Solid Modeling [8] advises design engineers to avoid, if possible, Boolean operations. This represents a dilemma, because trimming is among the most used operations in order to model complex models of engineering interest [9]. Riesenfeld, Haimes, and Cohen [2], eminent researchers in the field of computer aided design, urge for a complete reevaluation of present-day tools and practices.

1.2. A new modeling paradigm

Traditionally, meshing has been regarded as a separate process that converts a design model into an analysis model. This process is in general tedious and labor intensive and leads to model inaccuracies. The motivation for this work is to embed re-parameterization techniques directly within CAD, thereby enabling the modeling of watertight conforming CAD geometry that is directly compatible with downstream applications such as FEA and CAM. In this sense we share a common philosophy with the recent work of Urick et al. [10]. Our methodology is based on recent advances in topological vector field design and processing [11], quadrilateral meshing [12] and watertight spline representations [13]. Before we explore the state of the art in existing technologies we briefly summarize the workflow, which is illustrated in Fig. 3.

Assume that a given B-rep is of sufficient quality such that a linear triangulation can be obtained. Features such as B-rep vertices and boundaries are extracted directly from the B-rep and serve as input for the computation of a frame-field (see Fig. 3d). Additional information, such as the principal curvature directions obtained directly from the B-rep, can also be included. The frame-field is then used as a guide to obtain a high quality quadrilateral parameterization. We are interested in the coarse quad-layout structure, which is obtained by tracing the separatrices, that is, the parameterization iso-lines that emanate from singular points. The coarse quad-layout is used as a template to build a spline space. Each face in the layout corresponds to a conforming NURBS patch. If desired, these can be smoothly connected using technology such as T-splines [14] with improved treatment of extraordinary points [15]. Finally the spline surface is fitted to the original boundary representation to obtain a conforming and editable representation of the geometry.

1.3. Prior work

Quadrilateral surface meshing has been a topic of intense research for many decades. Recently, the field has seen a rejuvenation with many new and original contributions from the graphics community. For an overview we refer the reader to [16].

Meshing algorithms can be classified into local and global methods. Local meshing algorithms (e.g. [17,18]) are generally efficient, robust and are able to fit local features accurately. However, they do not provide much control over mesh singularities, which can lead to highly irregular unstructured meshes. Furthermore, even the most robust methods are generally “quad-dominant,” meaning they require a small number of triangles. In contrast, global algorithms [12,19–24] are capable of generating highly regular parameterizations containing few mesh singularities. Many are able to accurately capture both local and global geometric and topological features [25–27]. Furthermore, some methods provide interactive control over placement of singularities [28] and control over global patch layout [29,30]. In this work we focus solely on global parameterization methods because these methods provide the type of regular quadrilateral patch layouts that can be used as a basis for building analysis suitable splines. We do note that global parameterization is still an active area of research and obtaining robust and efficient implementations is challenging.

1.3.1. Frame-field guided re-parameterization

Let $M = (V, E, F)$ denote an orientable open or closed linear surface *triangulation*. The sets V, E and F refer to its *vertices* v_i, *edges* e_{ij} and *faces* f_{ijk}, respectively. A *cut-graph* G of M is a subset of the edges in E such that $M \setminus G$ is a simply connected mesh. We consider a special cut-graph of M that also extends to a prescribed set of topologically admissible cone-singularities $p \in M$ [31], derived from a frame-field $\mathcal{F}$ on M. The *fundamental domain* $M' = \left(V', E', F'\right)$ is a topological disk consisting of all faces in F ($F' \equiv F$) and can be straightforwardly obtained by cutting M along G [32].

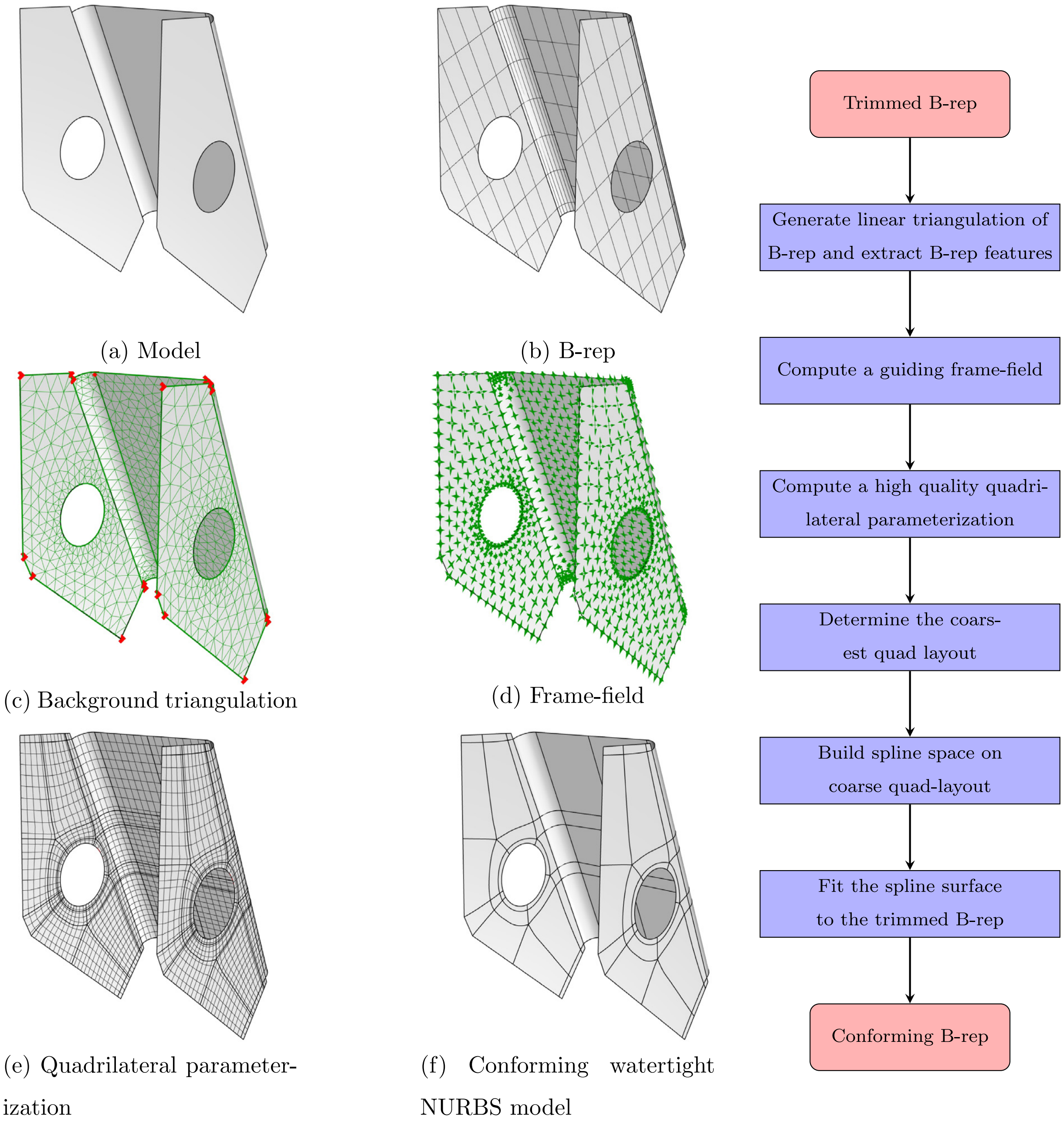
A *quadrilateral surface parameterization* of M is defined by two parameterization functions, denoted by

$$\{\ \Psi_i \in C^0(M'),\ i = 0, 1\ \}. \tag{1}$$

When carefully prescribed these functions define a quadrilateral grid on M when sampled at a suitably chosen set of iso-parameter lines. Particularly, if their values at cone-singularities are integers and transitions across the edges in G are seamless and of integer value, then these functions define an integer grid map [12,16,33–36]. Although there is progress [37] existence of arbitrary parameterizations is yet an unsolved problem.

Typically, these methods proceed by minimizing an energy

$$E(\Psi) := \sum_{i=0}^{1} \tfrac{1}{2} \int_{M'} |F_i - \nabla \Psi_i|^2 dA \tag{2}$$


(a) Model

(b) B-rep

(c) Background triangulation

(d) Frame-field

(e) Quadrilateral parameterization

(f) Conforming watertight NURBS model

Trimmed B-rep
Generate linear triangulation of B-rep and extract B-rep features
Compute a guiding frame-field
Compute a high quality quadrilateral parameterization
Determine the coarsest quad layout
Build spline space on coarse quad-layout
Fit the spline surface to the trimmed B-rep
Conforming B-rep

Fig. 3. Proposed workflow of obtaining a watertight conforming representation from a B-rep. The B-rep consisting out of trimmed NURBS patches in (b) is triangulated (c) and feature points and lines are extracted. Subsequently a frame-field (d) is computed that follows the prescribed features. Guided by the frame-field a quadrilateral parameterization is computed (e). Finally, the coarse layout structure of the model parameterization is determined and a new, conforming and watertight NURBS model (f) is constructed that fits the original B-rep within a specified tolerance.

subject to transition constraints along G and integer constraints, $\Psi_i(p) \in \mathbb{Z}$, at cone-singularities $p \in M$. Here $\{\, F_i \in \mathcal{F},\ i = 0, 1 \,\}$ denote the linear independent components of a frame-field that have been computed a-priori. There are several difficulties with this approach:

1. The vector fields $F_i, \ i = 0, 1$ may be far from being integrable (curl-free), in which case the computed parameterization functions typically exhibit undesirable behavior such as folds and large deviations from the original, smooth fields.
2. Mixed integer programming problems are NP-complete, and thus difficult and expensive to solve.
3. The two coordinates of the parameterization may overlap near parameterization singularities.

A complete, robust implementation of frame-field guided global parameterization will have to deal with the above issues. Previous work in [31,35,38] has focused on curl-free frame-fields that improve the first item. Other methods, for example [39], eliminate the use of integer constraints. [40] presents a post-processing strategy to deal with ill-behavior around parameterization singularities and other undesirable artifacts.

1.4. Contributions and outline

The contributions of this paper are summarized as follows:

1. We present a global quadrilateral surface meshing pipeline to re-parameterize B-rep geometry to conforming watertight NURBS geometry. Our approach builds on many aspects such as frame-fields (Section 2), global parameterization (Section 3), layout extraction, and NURBS fitting (Section 5). Several examples show how re-parameterized NURBS models can be used directly within structural shell analysis codes (Section 6.2).
2. We introduce specialized constraints in Section 3.2.4 that lead to well defined behavior near parameterization singularities. The computed parameterization is guaranteed to be inversion-free in the one-ring around a singularity by locally incorporating properties of an analytical solution. This means that parameterization separatrices can be traced effectively, without the need for additional post-processing, leading to simplified layout extraction (Section 5.1).
3. In Section 4 we present a convenient and efficient framework for handling linear constraints in general optimization problems. Constraints are built into a reduced basis using a special null-space method. The basis satisfies the prescribed linear constraints exactly and is maximally sparse, thereby preserving the sparsity of the original problem. The final linear system of equations is symmetric, positive definite and can be efficiently solved using Cholesky or the conjugate gradient method.

The first contribution is aimed towards the development of computer aided design techniques that yield analysis suitable geometric models, by construction, throughout the process of design. Success of the proposed technology depends on the solution of several open challenges discussed in the previous sub-section. Our second and third contribution completely resolve challenge (3) of those listed in Section 1.3.1. Future work relating to the remaining challenges and other interesting avenues for research are discussed in the conclusion, see Section 7.

2. Computation of frame-fields

In this section we briefly review the computation of a guiding frame-field on surface triangulations, feature alignment and extraction of frame-field topology by a process that is called frame-field matching. The approach is based on previous advances in topological vector field design and processing [11] and, in particular, the works in [36,41].

2.1. Preliminaries

Input of the proposed methodology is a valid B-rep computer aided design model. It is assumed that the model is of good quality such that a valid conforming triangulation can be extracted with relative ease. In practice this means that it has a valid cell-complex, manifold structure, and does not have sliver surfaces nor undesirable gaps that exceed the matching tolerance of the CAD system. A B-rep model that does not satisfy these conditions requires manual/semi-automatic clean-up using embedded CAD-tools or dedicated software such as CADfix.[1] An example of a valid B-rep is shown in Fig. 4. B-rep features, such as G^0 boundary curves, G^1 feature lines and sharp corners are automatically extracted. Subsequently, a conforming triangulation $M = (V, E, F)$ is computed that accurately

[1] See https://www.iti-global.com/cadfix.

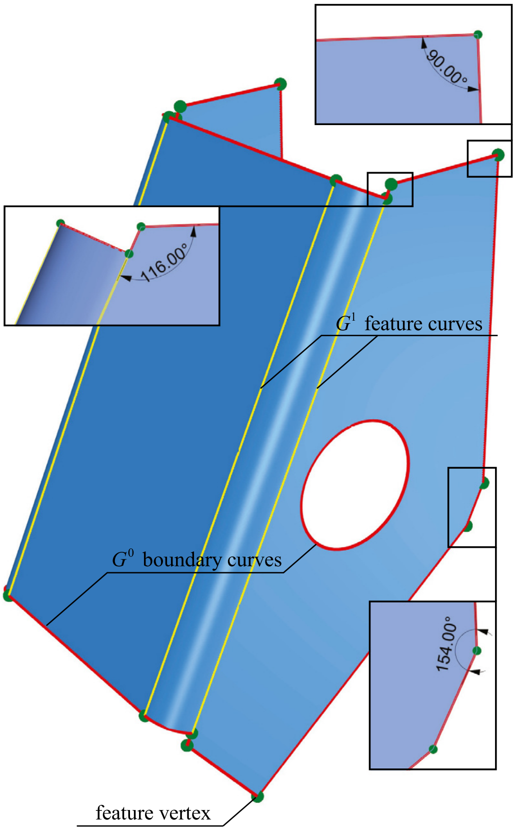


Fig. 4. Features are automatically derived from the B-rep. These include angles at feature vertices, G^0 feature curves such as boundaries and G^1 feature curves between B-rep faces.

captures the main features of the B-rep geometry. The discrete spaces used to represent frame-fields as well as the parameterization functions are based on linear interpolation of vertex-data on a triangulation. All computed fields are ultimately pulled back to the actual B-rep surface geometry, and hence, even a fairly coarse linear approximation of parameterization fields does not negatively effect the geometrical error made in the re-parameterization process (see Fig. 5).

2.1.1. Implementation

This work follows the approach proposed in [36] in which a frame-field is represented by a cross-field in a particular metric. First, a smooth metric is computed by minimizing a standard discrete Dirichlet energy over the space of symmetric positive definite tensor fields. In contrast to [36] we use a vertex-based discretization for the unknown tensor fields, instead of a face-based discretization. Our implementation of cross-fields follows that of [41]. We refer to these works for additional details regarding theory and implementation and highlight here several details important for our application.

Remark 2.1. The main advantage of a vertex-based representation over a face-based representation is that feature, boundary and corner constraints can be strongly applied with ease. One of the main advantages of the face-based approach, compared with vertex-based fields, is that consistent field tracing is easily implemented, see e.g. [39].

2.2. Feature constraints

Fig. 6 illustrates the type of constraints that are implemented at feature vertices. Locally, the behavior of the frame-field is controlled via a so-called feature angle, denoted by θ_{feature}. Given a feature angle, the frame-

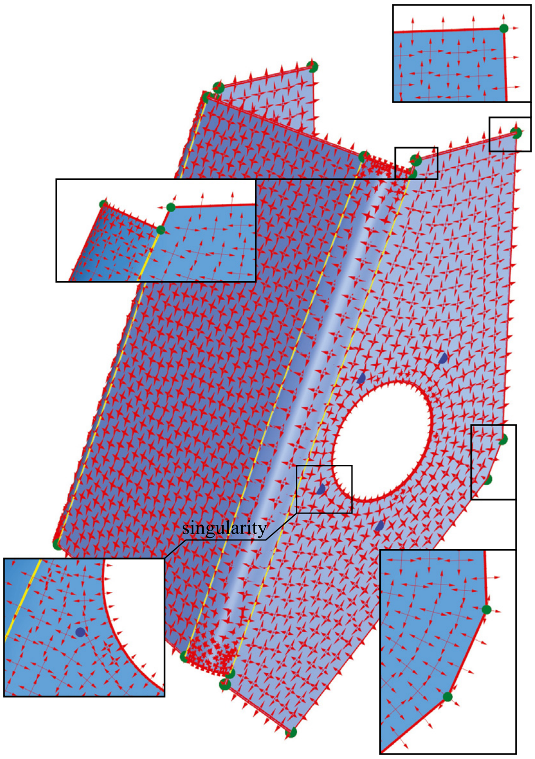
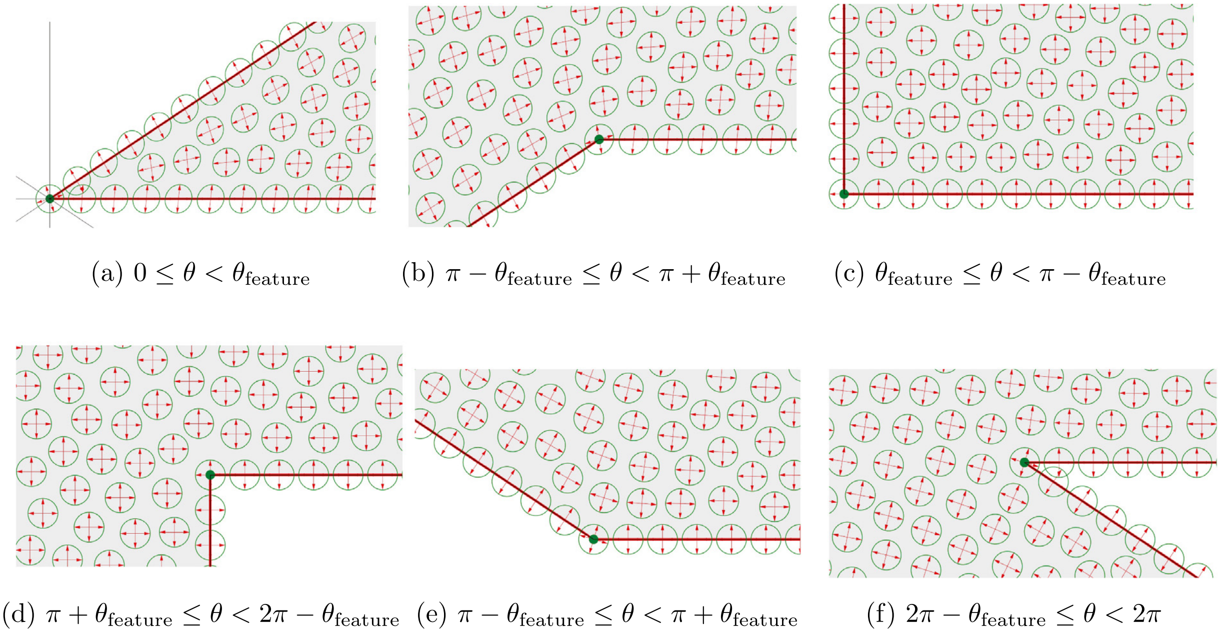


Fig. 5. The frame-field is constrained to satisfy G^0 and G^1 feature curve alignment and alignment at corners. Note that angles close to 90 deg are treated differently than angles close to 180 deg.



(a) $0 \leq \theta < \theta_{\text{feature}}$ (b) $\pi - \theta_{\text{feature}} \leq \theta < \pi + \theta_{\text{feature}}$ (c) $\theta_{\text{feature}} \leq \theta < \pi - \theta_{\text{feature}}$

(d) $\pi + \theta_{\text{feature}} \leq \theta < 2\pi - \theta_{\text{feature}}$ (e) $\pi - \theta_{\text{feature}} \leq \theta < \pi + \theta_{\text{feature}}$ (f) $2\pi - \theta_{\text{feature}} \leq \theta < 2\pi$

Fig. 6. Alignment constraints close to feature vertices in green. The feature angle θ_{feature}, set to 45 deg, provides precise control over the behavior at a feature vertex. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

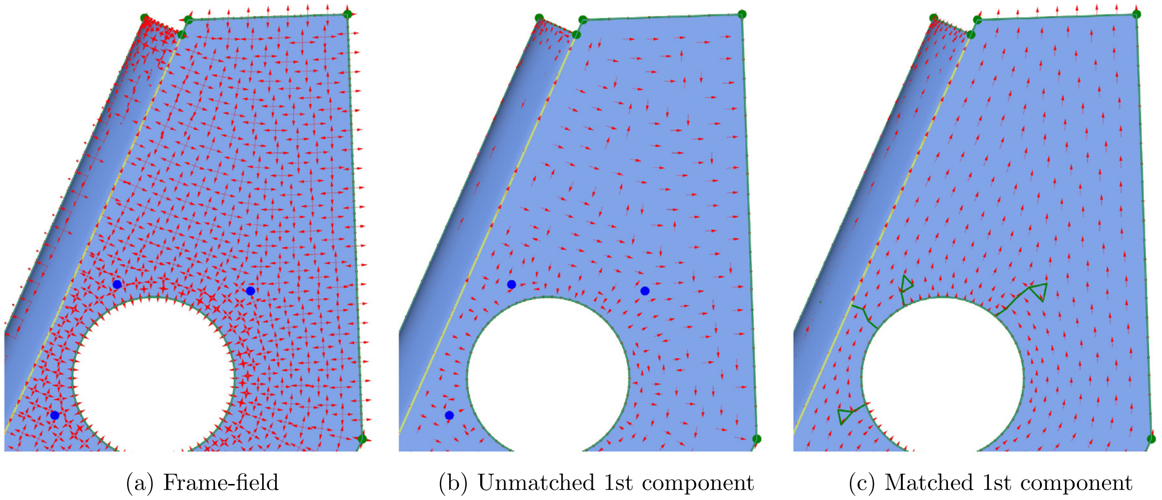

(a) Frame-field (b) Unmatched 1st component (c) Matched 1st component

Fig. 7. Frame-field matching. (b) The four individual vectors at every vertex are not ordered consistently. To obtain the individual components of a frame-field we have to consistently match them across the simplicial surface. The matched first component are shown in Figure (c). The second component can be obtained similarly. Note that the individual vector fields are continuous everywhere except across the edges of the cut-graph.

field vectors are defined, up to a possible scaling. If the scaling is unknown we choose the vectors to have unit-length. This uniquely defines the metric tensor and cross-field vector at these vertices and these values are used as boundary alignment constraints in the optimization problem.

2.3. *Frame-field matching*

The individual components of a frame-field are not necessarily ordered consistently. The process that recovers the 4 individual components of a cross- or frame-field is called *matching* and is illustrated in Fig. 7. Figure (a) shows a frame-field with singularities of index $-1/4$. The first, unordered, component is shown in Figure (b). The first linearly independent component of the frame-field, recovered after performing matching, is depicted in Figure (c). The process consists of the following steps

1. Identify faces that contain a frame-field singularity.
2. Compute the fundamental domain M' obtained by slicing M along edges of a cut-graph G from all interior boundary components, including the faces with singularity, to the exterior boundary in such a way that M' is homeomorphic to a disk.[2]
3. Compute a vertex-spanning-tree: a tree data-structure that connects a start vertex to every other vertex in the fundamental domain M' along a single unique path.
4. Compute matching difference between frame-field vectors located at adjacent vertices. This yields an integer corresponding to every edge e_{ij} that measures the number of $\pi/4$ rotations necessary to rotate the vector at vertex v_i to smoothly align with the vector at vertex v_j.
5. The cumulative sums of the matching differences along the paths of the vertex-spanning-tree yield the rotation of every frame-field vector to align smoothly with the frame-field vector at the start vertex.

For further reading on direction field matching we refer to [11]. For general references on computational geometry and mesh processing algorithms, including topics such as *fundamental domain*, *cut-graphs*, *shortest path algorithms*, and *vertex spanning-trees*, we refer to [32,42–44].

[2] Here we assumed that we are working with open surfaces of genus 0 with k boundary components. For general surfaces one has to compute a basis for the 1st cohomology group of the surface and use these to slice the domain to obtain disk-topology.

3. Configuration space of admissible parameterization functions

The two linearly independent components of the frame-field , $\{ F_i \in \mathcal{F}, \; i = 0, 1 \}$, provide topological and alignment constraints for the computation of a global, conforming parameterization. In particular, after the matching process we know

- position and index of the parameterization singularities
- which component is tangent along the boundary of M
- the change in component across edges of the cut-graph G.

In this section we use this information to construct a discrete configuration space of admissible parameterization functions that has the appropriate topological and alignment constraints built-in to the space. Subsequently, we use this space to compute a parameterization that minimizes the energy functional $E(\cdot)$ in (2).

3.1. Preliminaries

Consider the following discrete space

$$S^h(M') := \left\{ \sum_{j=1}^{|V'|} X_j \phi_j(x) \mid X_j \in \mathbb{C}, \; j = 1, \ldots, |V'| \right\} \tag{3}$$

Here $\{ \phi_j(x), \; j = 1, \ldots, |V'|, \; x \in M \}$ denote the classical linear hat functions defined on the fundamental domain M'. Note that the degrees of freedom are defined as complex numbers. Suppose $\Psi^h \in S^h$. Its real and imaginary components will be referred to as $\Psi_0^h := \mathrm{Re}\left(\Psi^h\right)$ and $\Psi_1^h := \mathrm{Im}\left(\Psi^h\right)$.

The configuration space of admissible parameterization functions is a subspace of S^h, denoted and defined by

$$S_c^h := \{ \Psi^h \in S^h \; : \; C(\Psi^h) = 0 \} \tag{4a}$$

Here $C(\cdot)$ denotes a set of linear constraints. In the coming sections we shall discuss each of the following constraints in more detail

$$C(\Psi^h) \Leftrightarrow \begin{cases} \text{Remove non-trivial kernel from } E(\cdot) \\ \text{Continuity over cuts in the cut-graph } G \\ \text{Tangential boundary conditions} \\ \text{Template constraints near singularities} \\ \text{Path constraints between singularities} \end{cases} \tag{4b}$$

We consider parameterization as a flow problem, similar to the setting of [22]. The configuration space S_c^h, with additional integer constraints, can also be identified as the space of integer grid maps [12,16,33–36]. In contrast to these works, we do not require that the value of the parameterization at cone singularities are integral, alleviating the need for a mixed-integer optimization. Instead, we allow user-prescribed path constraints that align iso-parameter values between multiple parameterization singularities. A disadvantage of this approach is that mesh extraction is more involved because now tracing iso-parameter lines involves tracing of *floating-point*, rather than *integral*, values.

3.2. Parameterization constraints

In this sub-section we discuss each of the constraints in (4b) in more detail. Subsequently, we show how to build the constraints into a reduced (sparse) basis such that they are satisfied by construction. This enables fast solvers such as Cholesky or the conjugate gradient method to be used to solve the resulting matrix problem.

3.2.1. Constraints that remove the kernel

The energy in (2) is not definite on the space of functions S^h. Elements of S^h are defined up to a constant. In order to obtain a unique solution it is important that this constant is set a priori, thereby removing the non-trivial kernel from the resulting set of matrix equations. A simple way to achieve this is by specifying a constraint on the average

$$\int_{\Omega} \Psi_i^h dx = 0, \quad i = 0, 1. \tag{5}$$

3.2.2. Continuity constraints across cuts

The two linearly independent vector fields $\{ F_i \in \mathcal{F},\ i = 0, 1 \}$, which are obtained from the frame-field after the matching process, are continuous everywhere except across edges of the cut-graph, see Fig. 7. It is here where the components switch direction. Consequently, special continuity constraints along edges of the cut-graph are necessary to couple the two components of the parameterization, $\{ \Psi_i^h,\ i = 0, 1 \}$, such that their tangent component is continuous across the cuts

$$\frac{\partial \Psi^h}{\partial t}(x_-) - (1i)^{k_{\text{match}}} \frac{\partial \Psi^h}{\partial t}(x_+) = 0 \quad \forall x \in G. \tag{6}$$

Here $k_{\text{match}} \in \{ 0, 1, 2, 3 \}$ is an integer that prescribes the change in coordinate direction across an edge of the cut-graph. We refer to [11] for additional details on this subject.

3.2.3. Tangential boundary constraints

Recall that under the metric computed in Section 2, components of the frame-field were aligned to various surface features, including the boundary. Let $\partial M_{\text{tangential}^i}$ denote the set of feature curves and boundaries that $F_i,\ i = 0, 1$ was forced to align with. To align the parameterization iso-lines with these feature curves and boundaries we have to add additional constraints that express this tangency relation

$$\frac{\partial}{\partial t} \Psi_i^h(x) = 0 \quad \forall x \in \partial M_{\text{tangential}^i}. \tag{7}$$

These constraints can be applied strongly for every edge on the feature curve or boundary.

3.2.4. Template constraints at singularities

The solution of the frame-field is always under-resolved in the vicinity of frame-field singularities, no matter the mesh-size (see, e.g. [45]). With mesh refinement this unresolved area shrinks. However, the character of the solution is independent of the mesh-size and therefore behavior around singularities remains problematic. Minimization of the energy in (2), without a proper treatment around singularities, generally leads to local fold-overs in the Ψ_0^h-and Ψ_1^h- coordinates of the parameterization. In [40] this problem is addressed in a post-processing step. We address this issue directly within the solver by incorporating properties of an analytical solution near singularities.

Singularity template

Consider a singularity with valence[3] ν. We wish to construct a conformal mapping, $G_\nu \ :\ \mathbb{C} \mapsto \mathbb{C}$, that is rotationally symmetric, with angle $\theta_\nu = 2\pi/\nu$, about the singularity located at origin $\mathbf{0} \in \mathbb{C}$. The mapping has a branch-cut aligned with the real axis. Let coordinates $y \in \mathbb{C}$ be parameterized in terms of polar coordinates such that $y = r\cos(\theta) + i \cdot r\sin(\theta)$, with angle $\theta \in [0, 2\pi)$ and radius $r \in \mathbb{R}^+$. The domain of the mapping can be divided into ν sections. Point $y \in \mathbb{C}$ is located in section $k(y) = \lfloor \theta(y)/\theta_\nu \rfloor$. With these definitions, the conformal mapping G_ν can be constructed as

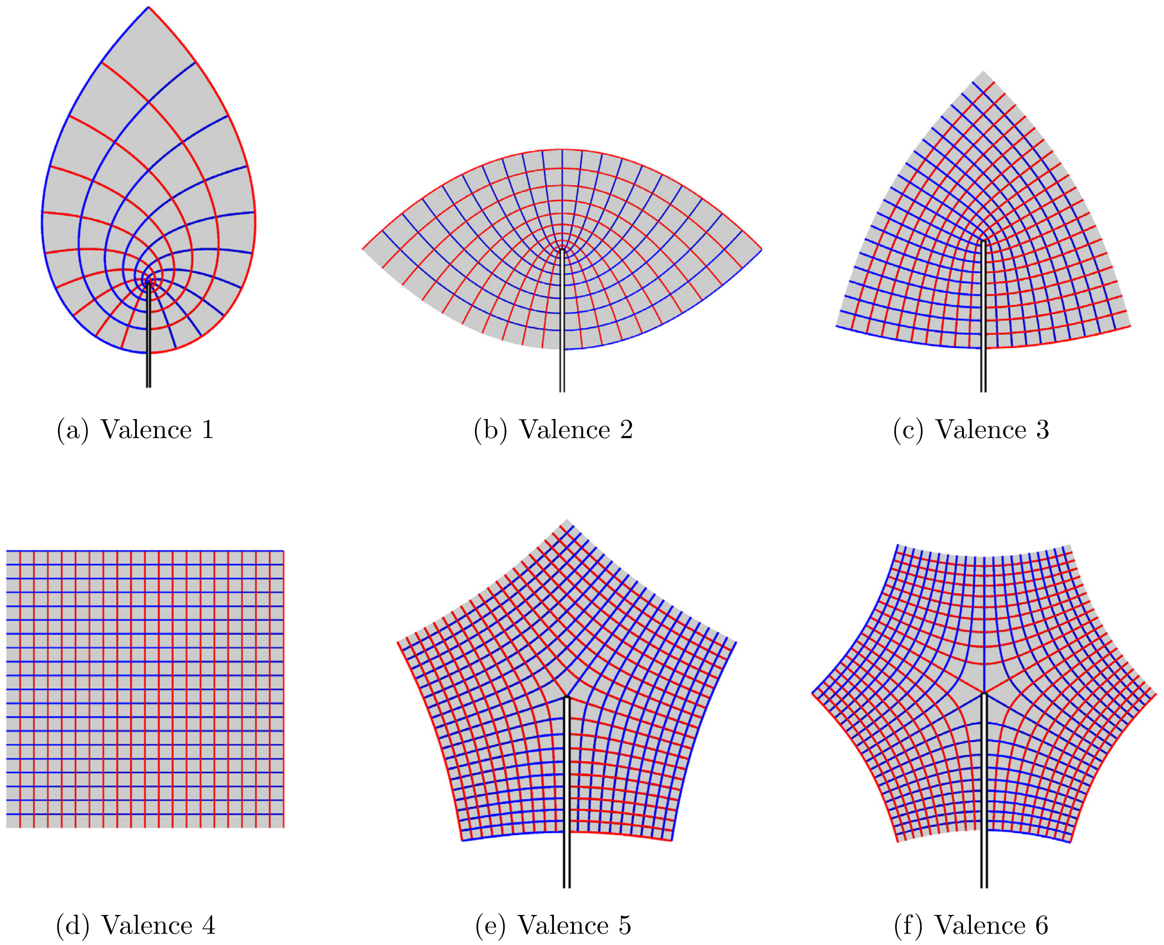
$$G_\nu(y) = \exp\left(i \cdot k(y)\pi/2\right) \left((y) \exp\left(-i \cdot 2k(y)\pi/\nu\right)\right)^{\nu/4} \tag{8}$$

The conformal mapping $G_\nu(\cdot)$ is fixed for a given valence ν and is constructed only once. For this reason we call this a *singularity template*. When considered in the form

$$G_\nu(y) = u(y) - v(y) \cdot i. \tag{9}$$

we can distinguish between the two separate coordinates of the mapping. Fig. 8 shows several examples of $G_\nu(y)$ corresponding to different mesh valence numbers. The blue iso-lines correspond to the u-coordinate while the red iso-lines correspond to the v-coordinate. As can be observed in the figures, the mappings are smooth everywhere except across the branch-cut. In practical applications we only consider valence 3 and 5; these are the type of singularities that emanate from the frame-field computation.

[3] Recall that valence and singularity index of a frame-field are related by: index = 1 − valence/4.


(a) Valence 1

(b) Valence 2

(c) Valence 3

(d) Valence 4

(e) Valence 5

(f) Valence 6

Fig. 8. Conformal mappings around singularities of different types. Note that the mapping is continuous everywhere except at the *branch cut*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Application of template constraints

To improve the behavior of the numerical solution we constrain it in the 1-ring neighborhood $\mathcal{N}_{\text{1-ring}}$ around each face containing a singularity to interpolate an analytical solution of a valence three or five conformal mapping, see Fig. 9. To achieve this, the 1-ring neighborhood is first flattened using triangle flattening [28]. Let $x \in \mathbb{C}$ denote coordinates of $\mathcal{N}_{\text{1-ring}}$ with origin o, chosen as the barycentric mean of the face containing the singularity, and consider the two-parameter family of mappings $g_\nu[R,S] : \mathbb{C} \mapsto \mathbb{C}$, defined by

$$g_\nu[R,S](x) = RG_\nu(x-o) + S. \tag{10}$$

The two complex degrees of freedom $R, S \in \mathbb{C}$ or, equivalently, the 4 real degrees of freedom $a, b, s, t \in \mathbb{R}$ such that

$$S = a + b \cdot i \qquad \text{(Translation)} \tag{11}$$

$$R = s + t \cdot i \qquad \text{(Rotation and scaling)} \tag{12}$$

are not known a priori and should follow from the solution to the global parameterization problem.

Consider the set of m global vertex coordinates $x_{\iota_k} \in \mathcal{N}_{\text{1-ring}},\ k = 1,\dots,m$ and consider the local coordinates $y_k = x_{\iota_k} - o,\ k = 1,\dots,m$. The constraints on the global parameterization functions $\Psi_i,\ i = 0,1$ can then be stated as

$$\begin{pmatrix} \Psi_0(x_{\iota_1}) \\ \Psi_0(x_{\iota_2}) \\ \vdots \\ \Psi_0(x_{\iota_n}) \\ \hline \Psi_1(x_{\iota_1}) \\ \Psi_1(x_{\iota_2}) \\ \vdots \\ \Psi_1(x_{\iota_n}) \end{pmatrix} = \left(\begin{array}{cccc} 1 & 0 & u(y_1) & v(y_1) \\ 1 & 0 & u(y_2) & v(y_2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & u(y_n) & v(y_n) \\ \hline 0 & 1 & -v(y_1) & u(y_1) \\ 0 & 1 & -v(y_2) & u(y_2) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & -v(y_n) & u(y_n) \end{array}\right) \begin{pmatrix} a \\ b \\ s \\ t \end{pmatrix} \tag{13}$$

Here $\Psi_i(x_{\iota_k}),\ i = 0, 1$ denote the two parameterization functions evaluated at vertex position x_{ι_k}. The matrix in (13), denoted by $\hat{\mathsf{B}} \in \mathbb{R}^{m\times 4}$, is designed such that the parameterization functions $\Psi_i(x),\ i = 0, 1$ interpolate, by construction, the chosen conformal mapping at a selection of vertex positions $x_{\iota_k},\ k = 1, \dots, m$ near the singularity

$$\Psi_0(x_{\iota_k}) = \mathrm{Re}(g_\nu[R, S](x_{\iota_k})) \tag{14}$$

$$\Psi_1(x_{\iota_k}) = \mathrm{Im}(g_\nu[R, S](x_{\iota_k})) \tag{15}$$

The matrix $\hat{\mathsf{B}}$ is fixed for a singularity of valence ν and set of local vertex coordinates $y_k,\ k = 1, \dots, m$ and is determined in a pre-processing step to the global parameterization.

It can be checked that $\hat{\mathsf{B}}$ is of full rank. Hence, there exists a matrix $\hat{\mathsf{C}} \in \mathbb{R}^{m-4\times m}$ such that

$$\hat{\mathsf{C}}\hat{\mathsf{B}} = \mathsf{0}. \tag{16}$$

Matrix $\hat{\mathsf{C}}$ can be interpreted as the linear constraints acting on the vertex values $\Psi_i(x_{\iota_k}),\ i = 0, 1,\ k = 1, \dots, m$ such that they interpolate the chosen conformal mapping in the one-ring around the singularity. Matrix $\hat{\mathsf{C}}$ is computed using the null-space machinery presented in the next section.

3.2.5. Path constraints

An iso-line is a curve parameterized by one of the coordinate functions, while keeping the other one constant, at least when not traversing a cut of the cut-graph. Suppose $\gamma\ :\ [a, b] \mapsto M$ is a path on M that does not traverse an edge of the associated cut-graph G. By the fundamental theorem of calculus we have

$$\int_\gamma \mathrm{d}\Psi_i^h = \Psi_i^h(\gamma(b)) - \Psi_i^h(\gamma(a)) \tag{17}$$

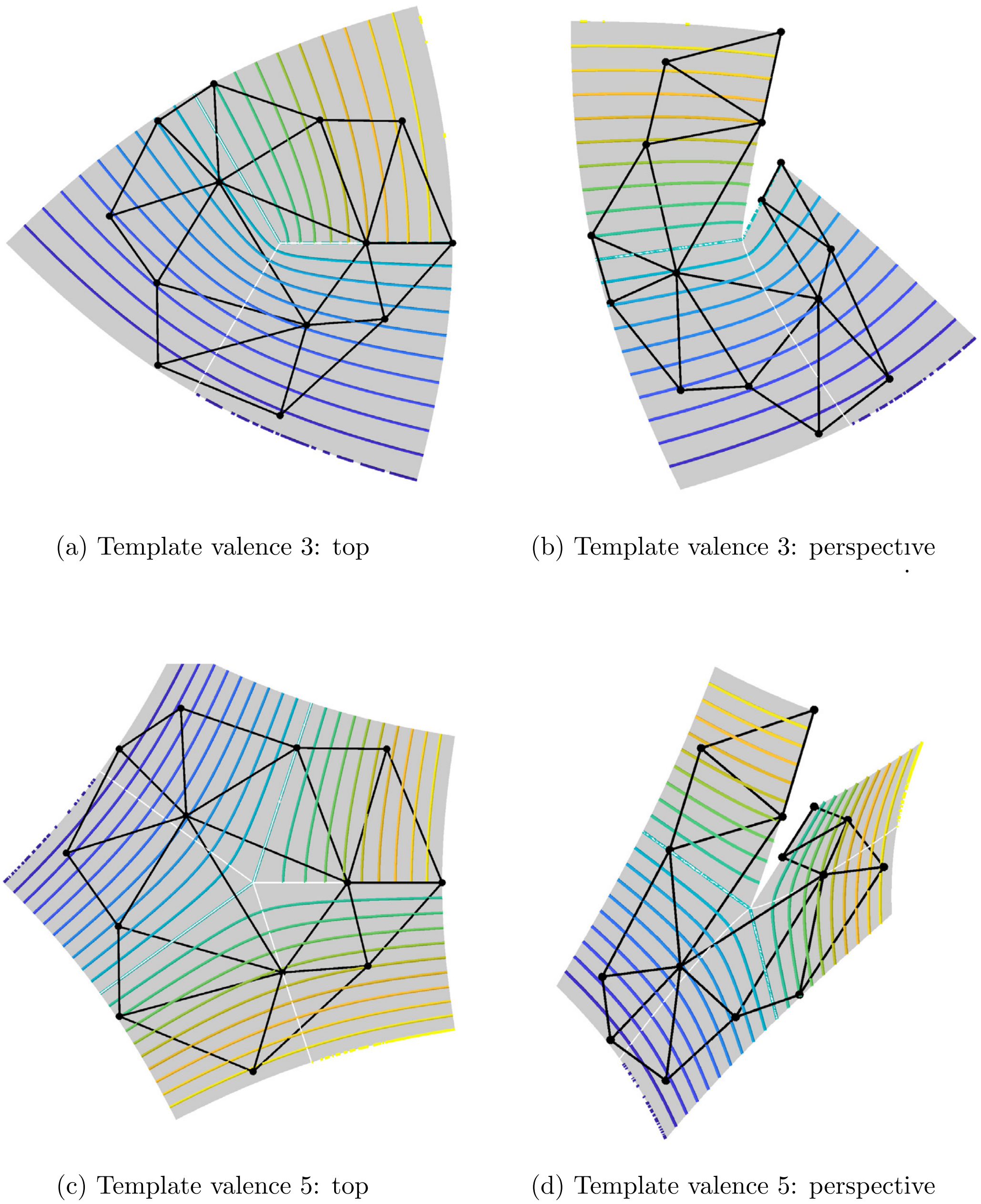
Hence, if we wish that the parameterization coordinate Ψ_i^h has the same value at $\gamma(b)$ as it has at $\gamma(a)$ then the flux over γ should cancel. If curve γ crosses an edge of G then a similar result holds, but we need to take into account the switch in component direction.

Fig. 10 shows an example of a parameterization before and after the path constraints have been added between parameterization singularities. In this work we allow the user to specify path constraints using a graphical user interface embedded within the CAD system. First, the user selects two singularities that need to be matched, as well as a topological path connecting the two. A default path could be the shortest path on M given by Dijkstra's algorithm. Then, the most likely component of the parameterization is chosen automatically based on the minimum of the computed fluxes, $\int_\gamma \mathrm{d}\Psi_i^h,\ i = 0, 1$, see Fig. 11. Finally, the new path constraints are used in the next parameterization solve leading to the result shown in Fig. 10.

4. Sparse null-space solution methodology

Minimization of the Dirichlet energy in (2) within the configuration space of admissible parameterization functions S_c^h leads to a saddle point problem of the following form

$$\begin{pmatrix} \mathsf{A} & \mathsf{C}^T \\ \mathsf{C} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \tag{18}$$



(a) Template valence 3: top

(b) Template valence 3: perspective

(c) Template valence 5: top

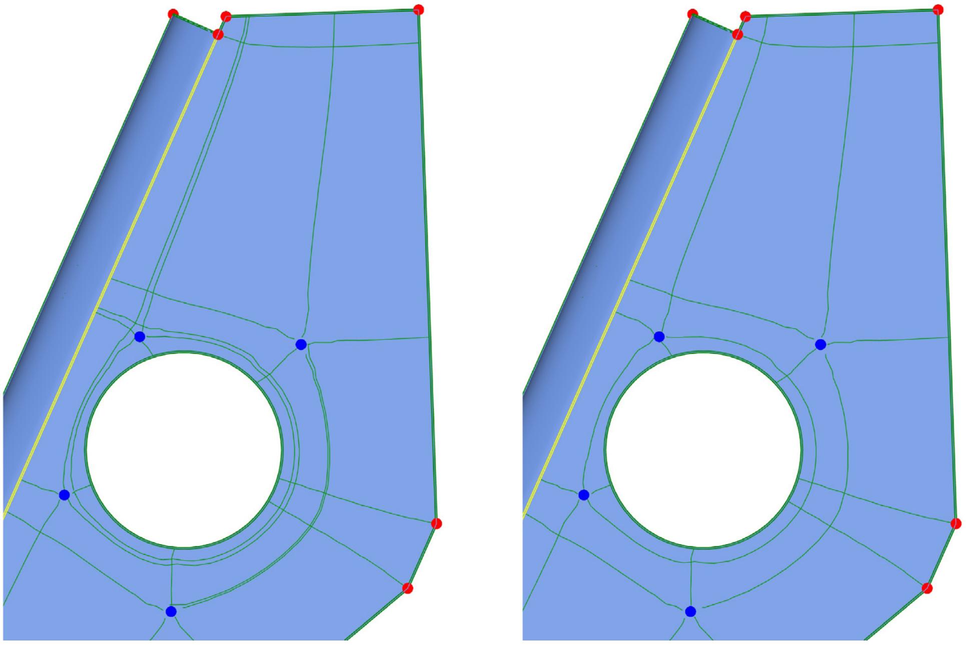
(d) Template valence 5: perspective

Fig. 9. Interpolation on the cut-mesh of the Ψ_0^h-coordinate of a conformal mapping with a valence 3 and 5 singularity. The Ψ_1^h-coordinate is similarly interpolated. Note that the cut-path to the face that contains the singularity coincides with the branch-cut of the conformal mapping. These constraints are applied within the parameterization such that locally, around the singularity, the parameterization is well defined and inversion-free.

Here $\mathsf{A} \in \mathbb{R}^{N\times N}$ is a symmetric positive semi-definite matrix[4] associated with the quadratic functional. Matrix $\mathsf{C} \in \mathbb{R}^{M\times N}$, assumed to be of full rank,[5] collects the constraints. $\mathbf{x} \in \mathbb{R}^N$ denotes the vector with degrees of freedom corresponding to the parameterization functions and $\mathbf{p} \in \mathbb{R}^M$ are Lagrange multipliers. By formulating the

[4] N equals $2|V'|$.

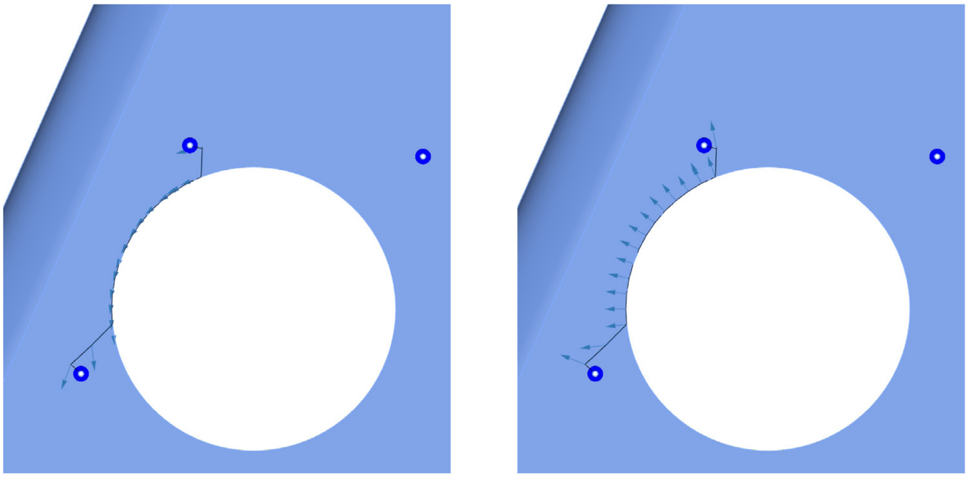
[5] This assumption simplifies the discussion. The solution strategy we propose does not need C to be of full rank. In practice, we allow constraints that are linearly dependent.



(a) Without path constraints

(b) With path constraints

Fig. 10. Path constraints can be added by the user leading to improved layouts.



(a) $\nabla\Psi_0^h$ along a path connecting two singularities

(b) $\nabla\Psi_1^h$ along a path connecting two singularities

Fig. 11. The total flux of both components of the parameterization is computed along a path connecting two singularities. Component Ψ_0^h has a smaller flux across the path than Ψ_1^h. It thus makes sense to place a zero-flux condition on component Ψ_0^h of the parameterization.

problem as a saddle point problem we have lost the positive (semi)-definiteness of the unconstrained problem. This is disadvantageous because it limits the use of fast direct and iterative solvers such as Cholesky factorization and the conjugate gradient method, respectively. Furthermore, instead of reducing the effective number of degrees of freedom with M, due to the presence of M linear constraints, we have actually increased their total by M Lagrange multipliers.

By constructing an explicit basis for the null-space of the constraint matrix C we can, instead, solve a direct problem that involves inverting a symmetric positive definite matrix. The key ingredient of a null-space method [46] is the availability of a matrix $\mathsf{Z} \in \mathbb{R}^{N \times N-M}$, of full rank, that satisfies

$$\text{range}\,(\mathsf{Z}) = \ker\left(\mathsf{C}\right) \tag{19}$$

In other words, the columns of Z span the null-space of C. The saddle-point problem can then be solved for $\mathbf{x}$ by first solving

$$\mathsf{Z}^T \mathsf{A}\mathsf{Z}\mathbf{y} = \mathsf{Z}^T \mathbf{b} \tag{20}$$

in terms of an auxiliary variable $\mathbf{y} \in \mathbb{R}^{N-M}$. Assuming the constraints in C properly remove the null space of matrix A, it follows that, $\mathsf{Z}^T\mathsf{A}\mathsf{Z}$ is symmetric positive definite and can be pre-factored with Cholesky or solved iteratively using the conjugate gradient method. The solution $\mathbf{x}$ can then simply be recovered by $\mathbf{x} = \mathsf{Z}\mathbf{y}$.

The different types of null-space methods depend specifically on the specific properties of Z and the way it is computed. For example, a null space basis can be computed using QR-factorization. The resulting matrix Z might, however, be quite densely populated. For an overview we refer to the work by Benzi, Golub and Liesen [46]. In our setting it is important that Z is sparse in order to preserve the sparsity of the original unconstrained problem. In the following we propose a strategy that leads to maximally sparse null-space bases. The approach is similar to those proposed in [47–49], in which non-uniform degree univariate B-splines and Tchebycheffian splines were constructed. An alternative approach that has features in common with ours is presented in [34].

4.1. Sparse basis for the null space of a single vector

First consider the case of a single constraint, $\mathbf{c}^T\mathbf{x} = 0$, with $\mathbf{c} \in \mathbb{R}^n$. Geometrically, we can consider the solution $\mathbf{x} \in \mathbb{R}^n$ to be restricted to an $n-1$-dimensional subspace of $\mathbb{R}^n$. Therefore, it is possible to transform the original constrained problem to an unconstrained one with $n-1$ degrees of freedom. Let $\mathsf{P} : \mathbb{R}^n \mapsto \mathbb{R}^n$ denote the permutation matrix such that $\tilde{\mathbf{c}} = \mathsf{P}\mathbf{c}$ has the following sparsity structure

$$\tilde{\mathbf{c}}^T = \begin{pmatrix} 0 & \cdots & 0 & \tilde{c}_K & \tilde{c}_{K+1} & \cdots & \tilde{c}_n \end{pmatrix}, \qquad |\tilde{c}_k| \leq |\tilde{c}_{k+1}|.$$

We seek a matrix $\tilde{\mathsf{Z}} \in \mathbb{R}^{n\times n-1}$ such that range$(\tilde{\mathsf{Z}})$ = ker$(\tilde{\mathbf{c}}^T)$. Furthermore, we require $\tilde{\mathsf{Z}}$ to be maximally sparse and have columns with unit 1-norm, which improves conditioning. In general, this is an l^1 optimization problem. In this simple case, however, we can determine $\tilde{\mathsf{Z}}$ explicitly. Firstly, assume $\tilde{\mathsf{Z}}$ has the following sparsity structure

$$\tilde{\mathsf{Z}} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & \tilde{Z}_{K,K} & & \\ & & & \tilde{Z}_{K+1,K} & \ddots & \\ & & & & \ddots & \tilde{Z}_{n-1,n-1} \\ & & & & & \tilde{Z}_{n,n-1} \end{pmatrix} \tag{21}$$

The non-zero values are uniquely determined by requiring that $\tilde{c}_k \tilde{Z}_{k,k} + \tilde{c}_{k+1}\tilde{Z}_{k+1,k} = 0$ and $|\tilde{Z}_{k,k}| + |\tilde{Z}_{k+1,k}| = 1$ for $k = K, \ldots, n-1$. Algorithm 1 encodes the discussed logic. It can be verified that $\tilde{\mathsf{Z}}$ is of full rank, is maximally sparse and is a basis for the null space of $\tilde{\mathbf{c}}$. Finally, note that we require the inverse permutation of $\tilde{\mathsf{Z}}$, that is $\mathsf{Z} = \mathsf{P}^{-1}\tilde{\mathsf{Z}}$.

4.2. Sparse basis for the null space of a matrix

Using induction, the described approach can be extended to the setting with more than one constraint. This logic is encoded in Algorithm 2. The numerical stability of the described approach remains to be studied. We do note that we have not witnessed any unstable behavior in practical computations.

Algorithm 1 Computation of a basis for the null space of a single vector

```
function NULLSPACE($\mathbf{c} \in \mathbb{R}^n$)
    $\mathbf{c} \leftarrow \mathsf{P}\mathbf{c}$        ▷ Order the values of $\mathbf{c}$ according to their absolute magnitude
    if abs($c_n$) $< \epsilon$ then        ▷ If $\mathbf{c}$ is trivial return identity matrix
        $\mathsf{Z} \leftarrow$ identity matrix (size: $n \times n$)
    else
        $\mathsf{Z} \leftarrow$ zero matrix (size: $n \times (n-1)$)
        $k \leftarrow 1$
        while $k < n$ & abs($c_k$) $< \epsilon$ do        ▷ Treat zero values
            $Z_{k,k} \leftarrow 1$
            $k \leftarrow k + 1$
        end while
        while $k < n$ do        ▷ Treat non-zero values
            $v \leftarrow c_k / c_{k+1}$
            $Z_{k,k} \leftarrow 1/(1 + \text{abs}(v))$
            $Z_{k+1,k} \leftarrow -Z_{k,k} * v$        ▷ $\text{abs}(Z_{k,k}) + \text{abs}(Z_{k+1,k}) = 1$
            $k \leftarrow k + 1$
        end while
        $\mathsf{Z} \leftarrow \mathsf{P}^{-1}\mathsf{Z}$        ▷ Apply inverse permutation
    end if
    return Z
end function
```

Algorithm 2 Computation of a null space basis

```
function NULLSPACE_BASIS($\mathsf{C} \in \mathbb{R}^{M \times N}$)
    $\mathsf{Z} \leftarrow$ identity matrix (size: $N \times N$)        ▷ Initialize null-space basis
    for $k = 1 : M$ do        ▷ Loop over constraints
        $\tilde{\mathsf{Z}} \leftarrow$ NULLSPACE($\mathsf{C}_{k:}$)        ▷ Compute nullspace of $k$-th row of C
        $\mathsf{Z} \leftarrow \mathsf{Z} * \tilde{\mathsf{Z}}$        ▷ update Z
        $\mathsf{C} \leftarrow \mathsf{C} * \tilde{\mathsf{Z}}$        ▷ update C
    end for
    return Z
end function
```

5. Quadrilateral layout and NURBS fitting

The template constraints at singularities lead to well-posed behavior around singularities. The computed parameterization is guaranteed to be inversion-free within the one-ring elements that are constrained. Consequently, all ν *separatrices* – iso-lines of the parameterization emanating from parameterization singularities – corresponding to a valence ν singularity are uniquely defined and can be successfully traced away from the singularity, across the triangulation M. If the parameterization is inversion-free then its separatrices, assumed of finite length, uniquely divide the surface up into a set of quadrilateral patches [40]. This so called quadrilateral layout, depicted in Fig. 12, provides the main structure that is used to build a conforming, watertight NURBS parameterization. To every

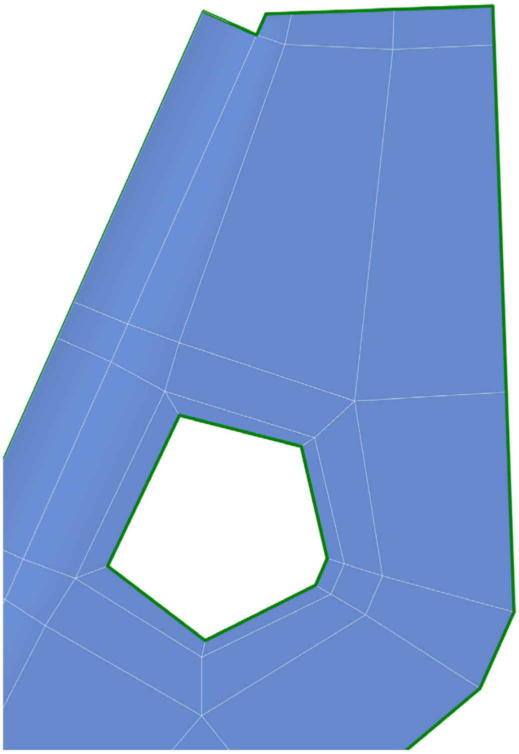


Fig. 12. The coarse quadrilateral layout forms the main topological structure used to reconstruct a multi-patch watertight NURBS parameterization.

face in the layout we associate a quadrilateral NURBS patch. Using NURBS, the continuity between patches will typically be limited to C^0. Unstructured spline technologies, such as T-splines, can additionally provide higher order continuity between patches.

5.1. Fitting problem

The fitting stage can be separated into an in-plane (intrinsic) and out-of plane (extrinsic) problem. The in-plane problem can be solved within the parameter space of the initial B-rep and only involves approximation near the trimming curves. The out of plane difference between the initial B-rep and the new conforming spline surface is where most of the approximation enters. Refine-ability of spline basis functions imply that the error due to approximation can be made arbitrarily small. Several projection methods may be useful here such as Bézier projection [50] and Coon's patches [5,51]. Other references of interest are [10,13].

Although spline-fitting is a well studied subject, there is remarkably little prior work done in the context of the considered application. We have identified several important criteria

1. Flat faces should remain flat.
2. Convex shapes should remain convex.
3. Conic shapes should be represented exactly.
4. Boundary curves and other feature curves, such as creases or those bordering fillets, should be represented, if possible, exactly. This does not include Boolean intersection curves, because these cannot in general be represented with low to medium order polynomials [52].
5. Free-form surfaces should be fitted up to a user-defined tolerance.

Note that some of these requirements may be at odds with one-another, e.g. convexity and optimal approximation do not usually go hand-in-hand. Consequently, the fitting stage represents a non-trivial step within the re-parameterization pipe-line.

5.2. Implementation

In this paper we use cubic rational Bézier patches to rebuild a watertight representation. First, each edge in the quadrilateral layout is approximated by a cubic rational Bézier curve. For the in-plane problem, we use a Coon's patch to reconstruct a rational surface from the boundary data, see [5,51]. Out-of-plane fitting can be performed by solving a Hermite interpolation problem. Consequently, the resulting watertight and conforming B-rep is G^1 continuous between patch interfaces. A more advanced approach, using e.g. T-splines, is left for future work.

6. Results

Traditionally, meshing has been considered a separate, intermediate process in between design and analysis. Instead, we apply re-parameterization techniques directly within CAD as part of the design process. This results in computer aided design models that are directly suitable for downstream applications such as finite element analysis. In this section we discuss implementational aspects of the methodology and apply it to several computer aided design models of engineering interest. The framework yields computer aided design models that are, by construction, analysis suitable. In this chapter this is illustrated by shell modal analyses applied to several parts of the unclassified generic hull vehicle model, provided by the U.S. Army Combat Capability Development Center (CCDC) Ground Vehicle Systems Center (USACCDCGVSC) (see Fig. 13).

6.1. Implementation in Rhino/Grasshopper

A proof of concept of the proposed methodology is implemented in Rhino 6 for Windows using the RhinoCommon C# SDK.[6] This includes support for sparse and dense linear algebra, triangulation of B-rep CAD geometry, computation of cross- and frame-fields, topological operations such as cut-graphs, computation of a quadrilateral parameterization, extraction of quad-meshes, and finally, native implementations of watertight and conforming NURBS geometry. Rhino ships with Grasshopper[7] which provides a visual programming interface for Rhino and has access to Rhino geometry objects. We use it to combine the different stages in the pipeline: (1) B-rep triangulation; (2) frame-field computation; (3) computation of the fundamental domain and extraction of individual frame-field components using matching; (4) global parameterization; (5) layout extraction. Grasshopper also provides an interface that allows selection of path-constraints in the global parameterization step. Application of the pipeline to the unclassified generic hull vehicle model is illustrated in Fig. 14.

6.2. Isogeometric shell analysis using LS-dyna

Figs. 15 and 16 show two re-parameterized B-rep models from the generic hull vehicle. This includes the model that has been used throughout this paper to illustrate several aspects of the methodology and a more complicated model. The figures illustrate several of the details discussed, such as alignment of the parameterization with main features and global coarse patch structure. We recall that the workflow discussed automatically places singularities of a certain type in desirable locations. The global patch layout is subsequently optimized by the user by prescribing a small set of path constraints connecting singularities. Finally, the B-rep is rebuilt according to the obtained patch layout using cubic rational Bézier patches. Both models exactly describe the circular holes and fit exactly the prescribed features of the initial B-rep. Note that symmetry across the main axis has been used to create the watertight NURBS models.

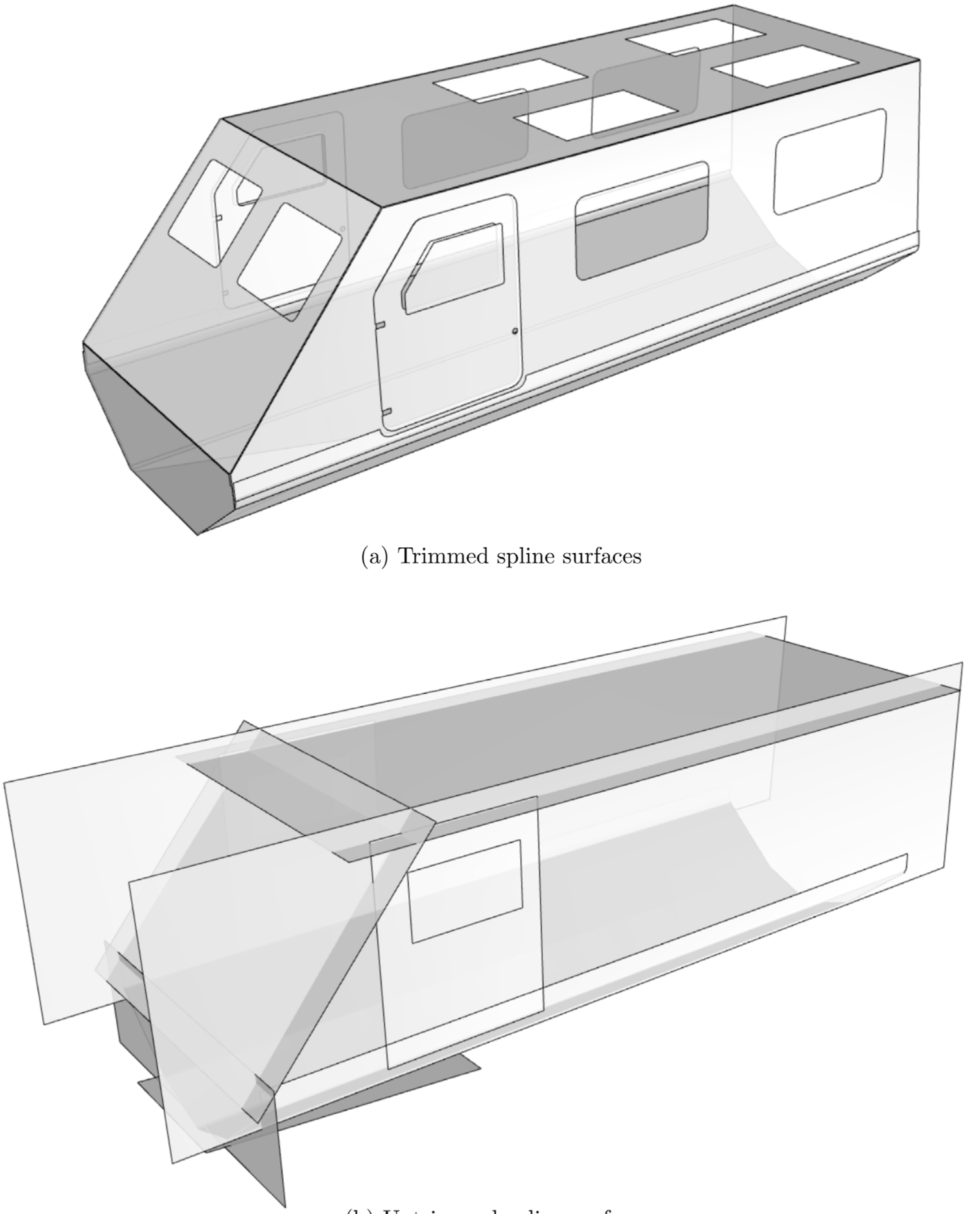
We perform isogeometric shell modal analysis on the two models using LS-DYNA of Livermore Software Technology Corporation,[8] an industrial, general purpose finite element code. LS-DYNA features a shell NURBS element and has several aspects of isogeometric analysis implemented, including explicit/implicit dynamics, eigenvalue problems and large deformation analyses involving contact [53,54], combined with a large class of constitutive models. The analysis models, depicted in Figs. 15d and 16d are obtained by uniform h-refinement using global knot insertion [55].

We have used the following setup

[6] https://www.rhino3d.com/download/Rhino-SDK/6.0/release.

[7] https://www.rhino3d.com/6/new/grasshopper.

[8] https://www.lstc.com/.



(a) Trimmed spline surfaces

(b) Untrimmed spline surfaces

Fig. 13. Trimmed representation of the outer hull plating of the DEVCOM Generic Hull vehicle model.

- Reissner–Mindlin shell formulation.
- Linear isotropic material with density $\rho = 7800$ kg/m^3, Young's modulus $E = 200$ GPa and Poisson's ratio $\nu = 0.3$ (aluminum).
- Plate thickness $h = 12$ mm.
- Full Gaussian quadrature.
- Consistent mass.

The Reissner–Mindlin shell simulates shear deformable shells and features rotational degrees of freedom. Consequently, we require only C^0-continuity in between elements and patches. We refer to Benson et al. [54] for further details.

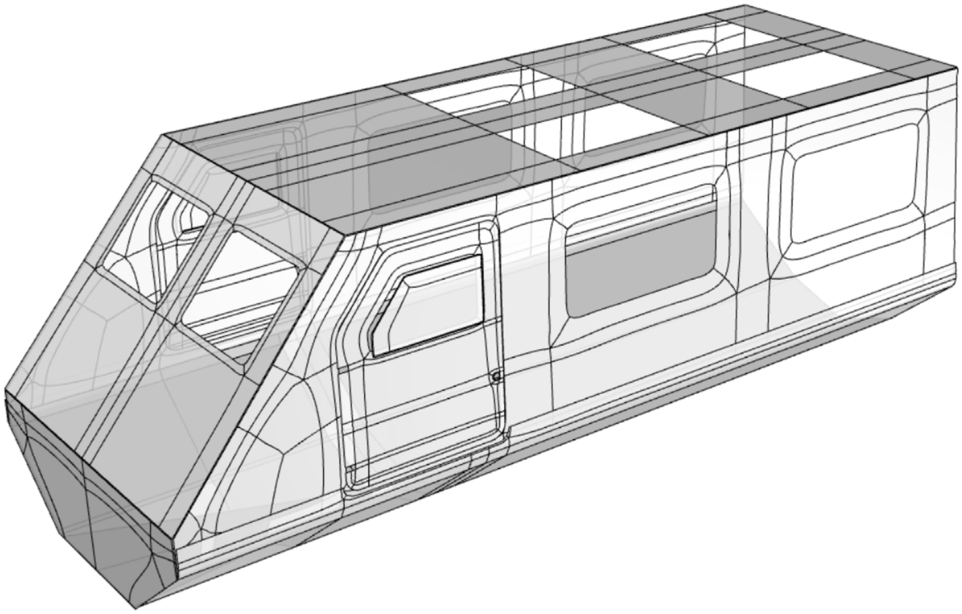


Fig. 14. Conforming parameterization of mid-surfaces of the outer hull plating of the DEVCOM Generic Hull vehicle model.

Figs. 17 and 18 show a selection of the low modes obtained from the discussed linear modal analysis. The color plots show the maximum displacement of the mode under consideration.

7. Conclusions and recommendations

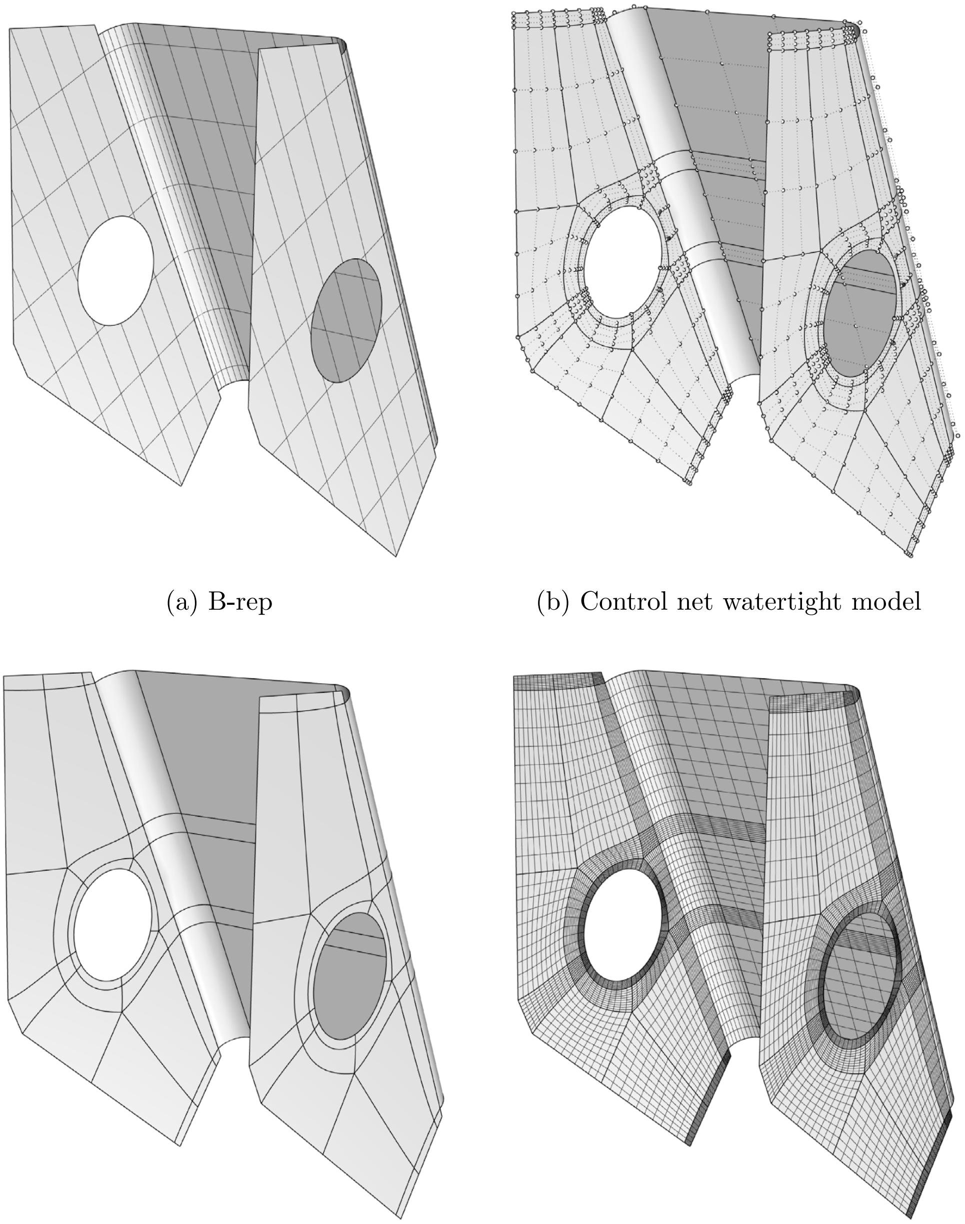
Although computer aided design has in many ways been very successful it has fallen short of its original intent to serve as an interactive environment for efficient product development. Many of the problems trace back to a particular geometry description developed in the early 70s, called the *B-rep* or *boundary representation*. The B-rep combines aspects of explicit parametric geometry with an implicit representation that arises during Boolean operations using a process called trimming. The description is non-conforming, often fraught with inconsistencies such as gaps and overlaps, and therefore incompatible with downstream applications such as finite element analysis, optimization and manufacturing [5]. This paper has focused on a combination of techniques that, ultimately, should alleviate some of the complications arising from trim. The methodology is based on frame-field guided global parameterization and watertight spline representations.

7.1. Contributions

Frame-field guided parameterization methods offer the kind of high quality meshes necessary within the context of computer aided design. These methods, however, suffer from several theoretical and practical problems leading to lack of robustness. Typically, these methods involve integer constraints, which are difficult and expensive to solve. A more pressing issue is that the computed parameterization may have folds and other undesirable artifacts due to an input frame-field that is far from being locally integrable. Undesirable folds are especially common near parameterization singularities.

We proposed specialized constraints that led to well defined behavior near parameterization singularities by locally incorporating properties of an analytical solution. The computed parameterization is guaranteed to be inversion-free within the constrained neighborhood of the singularity. Consequently, all separatrices are uniquely defined and can be successfully traced away from the singularity. This significantly simplifies the quad-layout extraction process, alleviating the need for complicated post-processing as proposed in [40].

These and other linear constraints were build into a reduced basis that spans a configuration space of feasible parameterization functions. The enabling technique used is a new null-space method that, besides satisfying the



(a) B-rep

(b) Control net watertight model

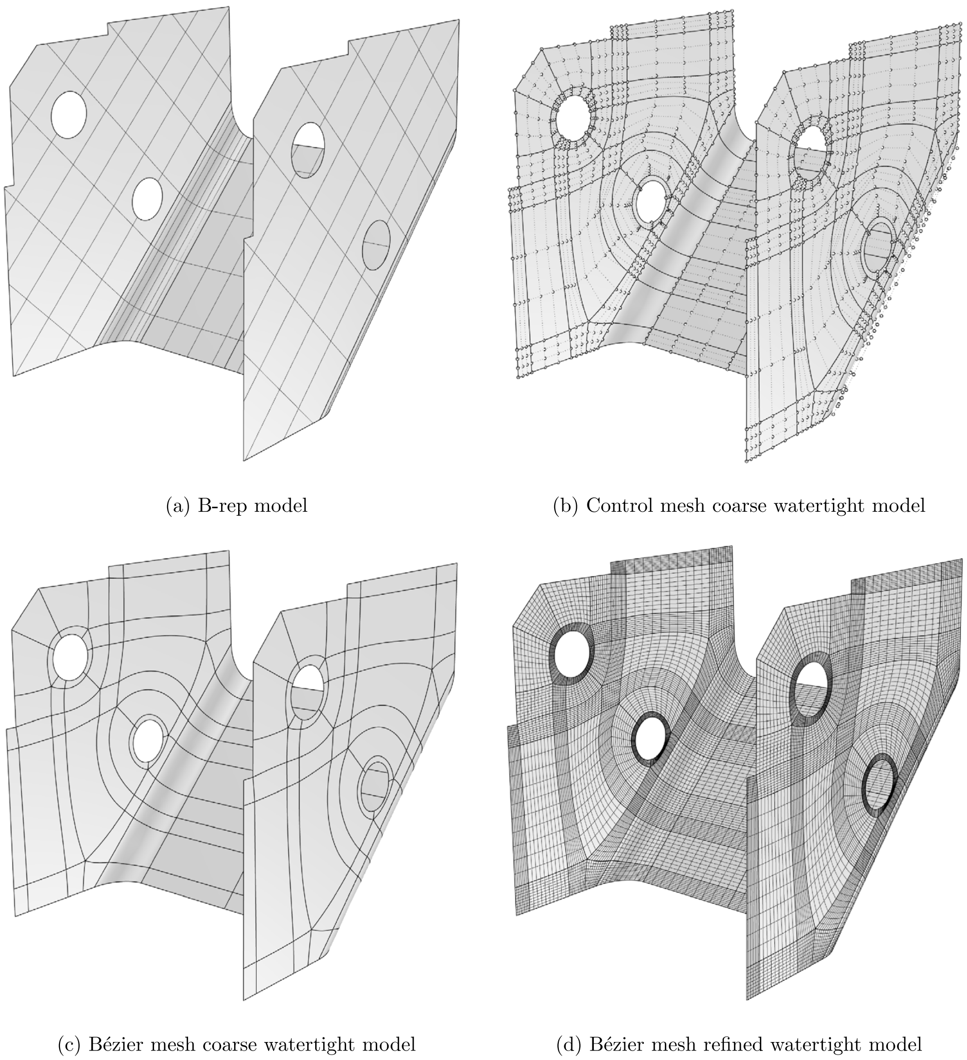
(c) Coarse watertight model

(d) Refined watertight model

Fig. 15. A B-rep is re-parameterized into a conforming, watertight model consisting of 78 cubic rational Bézier patches. Figure (b) shows the NURBS control net and Figure (c) and (d) the Bézier mesh corresponding to the coarse and fine discretization. The refined model, which is used in modal analysis, consists of 10 × 10 uniformly refined cubic NURBS patches that are C^2 within the patch and C^0 in between patches.

specified constraints exactly, is optimally sparse. Consequently, the sparsity of the original problem is preserved, thereby enabling fast solution techniques such as Cholesky and the Conjugate Gradient method. The proposed algorithm is simpler to implement and leads to sparser null-spaces bases than the approach developed in [34].

Several criteria were discussed to reconstruct spline parameterizations on coarse quadrilateral patch layouts obtained from the global parameterization method. We used rational cubic Bézier patches combined with Coon's


(a) B-rep model

(b) Control mesh coarse watertight model

(c) Bézier mesh coarse watertight model

(d) Bézier mesh refined watertight model

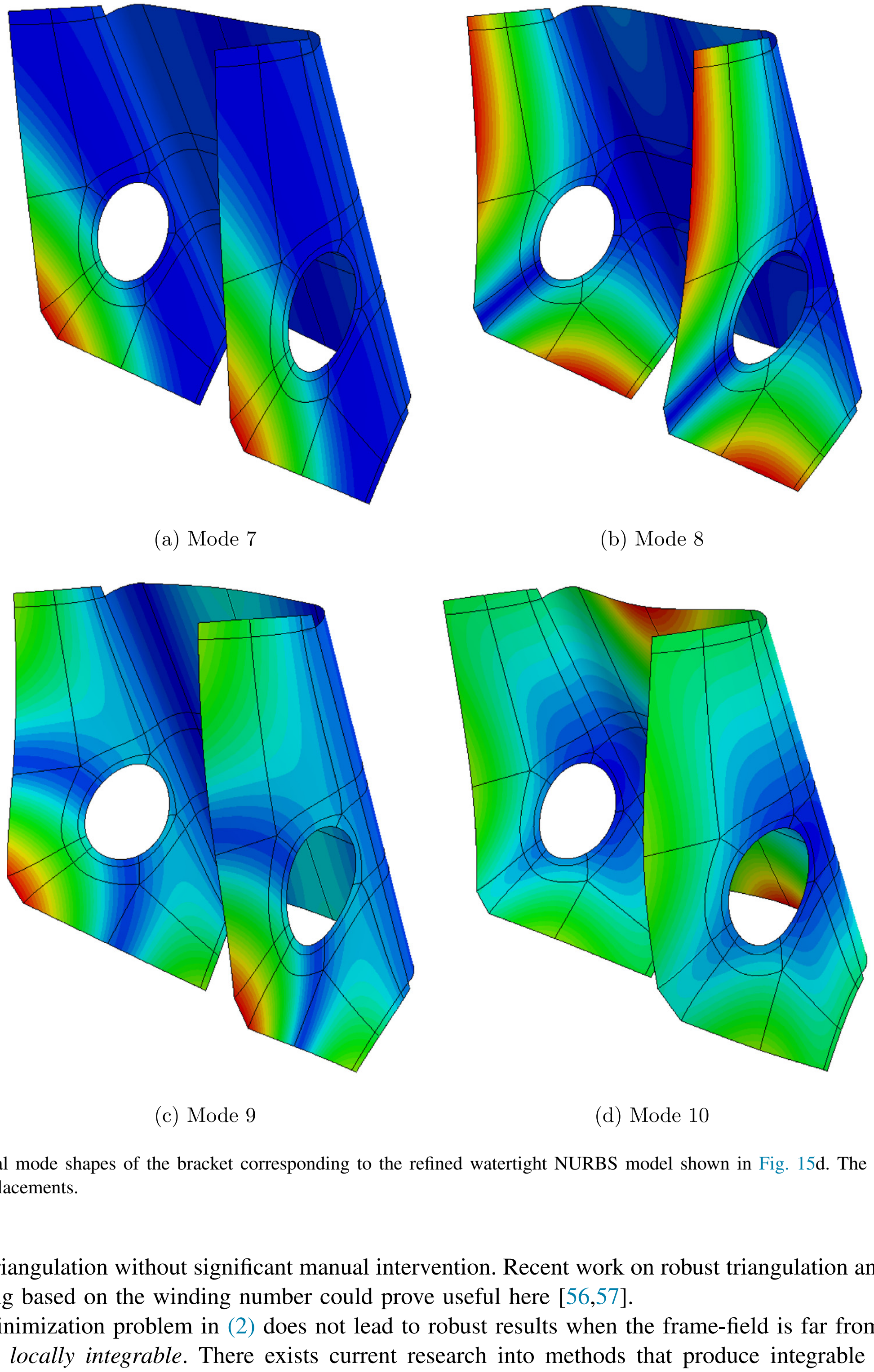
Fig. 16. A B-rep is re-parameterized into a conforming, watertight model consisting of 146 cubic rational Bézier patches. Figure (b) shows the NURBS control net and Figure (c) and (d) the Bézier mesh corresponding to the coarse and fine discretization. The refined model, which is used in modal analysis, consists of 10 × 10 uniformly refined cubic NURBS patches that are C^2 within the patch and C^0 in between patches.

patch interpolation to rebuild watertight and conforming NURBS parameterizations of the trimmed B-reps. Finally, we used LS-Dyna to perform isogeometric shell modal analysis applied to several of the re-parameterized NURBS models in order to illustrate their analysis suitability.

7.2. Directions for future work

The following list addresses limitations of the proposed technology and provides recommendations for future work:

(i) The proposed methodology assumes that a valid triangulation can be obtained directly from the B-rep geometry. We found that current CAD systems do not have all the necessary tools to generate valid triangulations of B-rep geometry for complex models. Specifically, in the setting where B-rep models have defects that exceed the tolerance of the CAD system it is generally not possible to obtain a topologically



(a) Mode 7

(b) Mode 8

(c) Mode 9

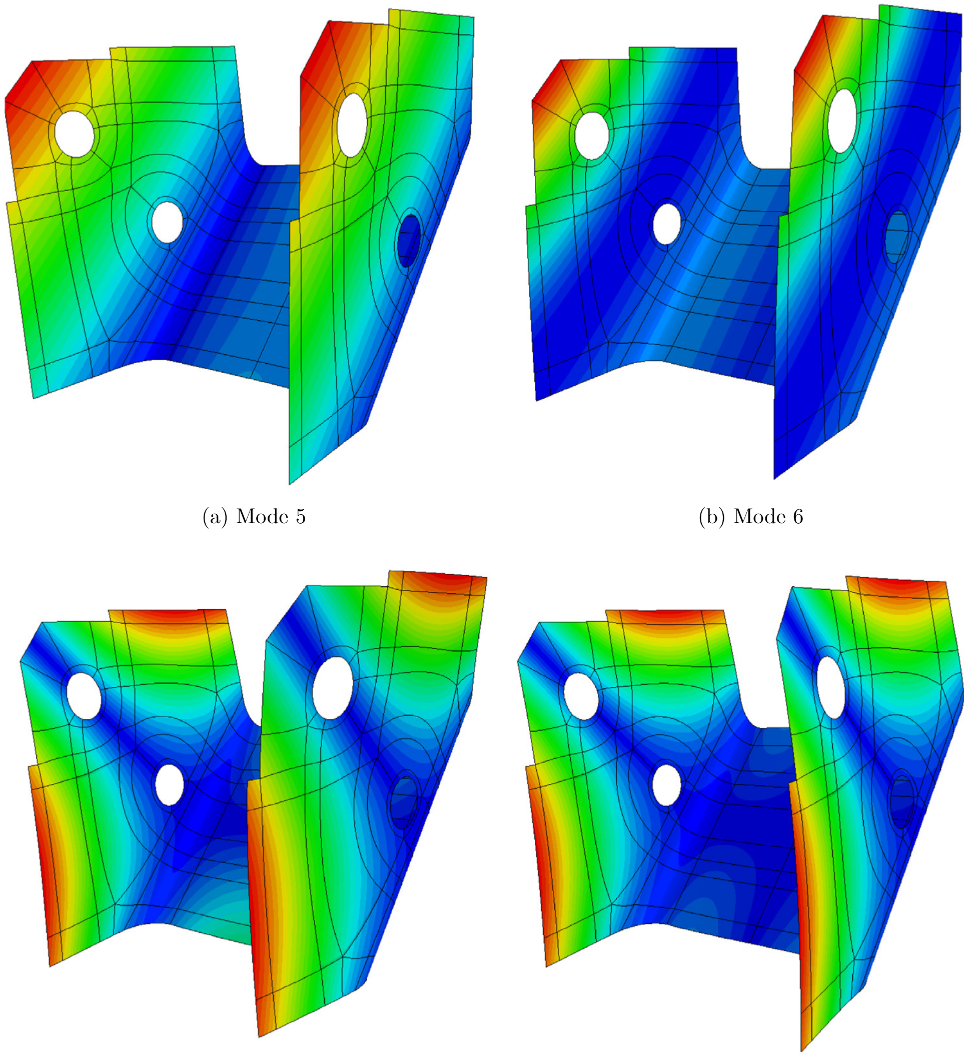
(d) Mode 10

Fig. 17. Several mode shapes of the bracket corresponding to the refined watertight NURBS model shown in Fig. 15d. The color highlights maximum displacements.

valid triangulation without significant manual intervention. Recent work on robust triangulation and tetrahedral meshing based on the winding number could prove useful here [56,57].

(ii) The minimization problem in (2) does not lead to robust results when the frame-field is far from being *curl-free* or *locally integrable*. There exists current research into methods that produce integrable frame-fields, see [35,38,58]. Another direction would be to use only the directional information of the frame-field, not its magnitude, see e.g. [39].

(iii) In this work we used manual constraints to line up singularities. While this avoided the use of integer constraints, it would be good to have a means of automatically selecting such constraints. The dual strip



(a) Mode 5

(b) Mode 6

(c) Mode 7

(d) Mode 8

Fig. 18. Several mode shapes of the bracket corresponding to the refined watertight NURBS model shown in Fig. 16d. The color highlights maximum displacements.

weaving method proposed in [19,29] could provide a means of achieving this without the need for integer variables.

(iv) Models with different levels of detail require areas with local refinement. Hence, instead of a quadrilateral mesh or layout the objective could be a T-mesh. A few works provide a first approach in this direction [39,59].

(v) Our approach to spline fitting used C^0 NURBS patches to rebuild a B-rep. It is cumbersome to maintain geometric continuity between patches unless continuity is build into the basis. Technologies such as T-splines remedy these issues. Future work includes a combination of item (iv) above and technologies such as T-splines, together with more advanced fitting techniques.

(vi) The current approach constitutes a global reparameterization of the input B-rep model. Typically, the generated watertight parameterization is quite different from the original one. However, this global approach admits boundary conditions that can be used to apply the theory locally. Thus there is potential for a user to apply the reparameterization computation to some subdomain of interest. Such a reparameterization would likely require some level of user interaction, and would generally introduce T-junctions. Future work includes exploring this framework.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Dr. David Bommes, Dr. Leif Kobbelt and Dr. Daniele Panozzo for useful discussions related to framefield guided global parameterization. T.J.R. Hughes and R.R. Hiemstra were partially supported by the National Science Foundation Industry/University Cooperative Research Center (IUCRC) for Efficient Vehicles and Sustainable Transportation Systems (EV-STS), and the United States Army CCDC Ground Vehicle Systems Center (TARDEC/NSF Project # 1650483 AMD 2). Kendrick Shepherd was partially supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1610403. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] E. Cohen, T. Lyche, R.F. Riesenfeld, MCAD: Key historical developments, Comput. Methods Appl. Mech. Engrg. 199 (5–8) (2010) 224–228.

[2] R.F. Riesenfeld, R. Haimes, E. Cohen, Initiating a CAD renaissance: Multidisciplinary analysis driven design: Framework for a new generation of advanced computational design, engineering and manufacturing environments, Comput. Methods Appl. Mech. Engrg. 284 (2015) 1054–1072.

[3] P.T. Boggs, A.E.E. Althsuler, A.R.E.E. Larzelere, E.J. Walsh, R.L. Clay, M.F.S.N.L. Hardwick, DART System Analysis, Tech. rep., Sandia National Laboratories, 2005.

[4] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39–41) (2005) 4135–4195.

[5] B. Marussig, T.J. Hughes, A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects, Arch. Comput. Methods Eng. 25 (4) (2018) 1059–1127.

[6] D.J. Kasik, W. Buxton, D.R. Ferguson, Ten CAD challenges, IEEE Comput. Graph. Appl. 25 (2005) 81–92.

[7] L.A. Piegl, Ten challenges in computer-aided design, Comput.-Aided Des. 37 (4) (2005) 461–470.

[8] D.E. LaCourse, Handbook of Solid Modeling, McGraw-Hill, Inc., 1995, pp. 6.17–6.18, 6.24.

[9] I. Stroud, Boundary Representation Modelling Techniques, Springer Science & Business Media, 2006.

[10] B. Urick, B. Marussig, E. Cohen, R.H. Crawford, T.J. Hughes, R.F. Riesenfeld, Watertight boolean operations: A framework for creating CAD-compatible gap-free editable solid models, Comput. Aided Des. 115 (2019) 147–160.

[11] A. Vaxman, M. Campen, O. Diamanti, D. Panozzo, D. Bommes, K. Hildebrandt, M. Ben-Chen, Directional field synthesis, design, and processing, in: Computer Graphics Forum, Vol. 35, Wiley Online Library, 2016, pp. 545–572.

[12] D. Bommes, M. Campen, H.-C. Ebke, P. Alliez, L. Kobbelt, Integer-grid maps for reliable quad meshing, ACM Trans. Graph. 32 (4) (2013) 98.

[13] T.W. Sederberg, G.T. Finnigan, X. Li, H. Lin, H. Ipson, Watertight trimmed NURBS, ACM Trans. Graph. 27 (3) (2008) 79:1–79:8.

[14] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Trans. Graph. 22 (3) (2003) 477–484.

[15] D. Toshniwal, H. Speleers, T.J. Hughes, Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations, Comput. Methods Appl. Mech. Engrg. 327 (2017) 411–458.

[16] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, D. Zorin, Quad-mesh generation and processing: A survey, in: Computer Graphics Forum, Vol. 32, Wiley Online Library, 2013, pp. 51–76.

[17] T.D. Blacker, M.B. Stephenson, Paving: A new approach to automated quadrilateral mesh generation, Int. J. Numer. Methods Eng. 32 (4) (1991) 811–847.

[18] S.J. Owen, M.L. Staten, S.A. Canann, S. Saigal, Q-Morph: an indirect approach to advancing front quad meshing, Int. J. Numer. Methods Eng. 44 (9) (1999) 1317–1340.

[19] M. Campen, D. Bommes, L. Kobbelt, Dual loops meshing: quality quad layouts on manifolds, ACM Trans. Graph. 31 (4) (2012) 110.

[20] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, J.C. Hart, Spectral surface quadrangulation, in: ACM SIGGRAPH 2006 Papers, in: SIGGRAPH '06, Association for Computing Machinery, New York, NY, USA, 2006, pp. 1057–1066.

[21] X. Gu, S.-T. Yau, Global conformal surface parameterization, in: L. Kobbelt, P. Schroeder, H. Hoppe (Eds.), Eurographics Symposium on Geometry Processing, The Eurographics Association, 2003.

[22] F. Kälberer, M. Nieser, K. Polthier, Quadcover-surface parameterization using branched coverings, in: Computer Graphics Forum, Vol. 26, Wiley Online Library, 2007, pp. 375–384.

[23] N. Lei, X. Zheng, J. Jiang, Y.-Y. Lin, D.X. Gu, Quadrilateral and hexahedral mesh generation based on surface foliation theory, Comput. Methods Appl. Mech. Engrg. 316 (2017) 758–781.

[24] N. Lei, X. Zheng, Z. Luo, D.X. Gu, Quadrilateral and hexahedral mesh generation based on surface foliation theory II, Comput. Methods Appl. Mech. Engrg. 321 (2017) 406–426.

[25] X. Fang, H. Bao, Y. Tong, M. Desbrun, J. Huang, Quadrangulation through morse-parameterization hybridization, ACM Trans. Graph. 37 (4) (2018).

[26] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, H. Bao, Spectral quadrangulation with orientation and alignment control, in: ACM SIGGRAPH Asia 2008 Papers, in: SIGGRAPH Asia '08, Association for Computing Machinery, New York, NY, USA, 2008.

[27] A. Myles, N. Pietroni, D. Kovacs, D. Zorin, Feature-aligned t-meshes, ACM Trans. Graph. 29 (4) (2010).

[28] K. Crane, M. Desbrun, P. Schröder, Trivial connections on discrete surfaces, in: Computer Graphics Forum, Vol. 29, Wiley Online Library, 2010, pp. 1525–1533.

[29] M. Campen, L. Kobbelt, Dual strip weaving: interactive design of quad layouts using elastica strips, ACM Trans. Graph. 33 (6) (2014) 183.

[30] Y. Tong, P. Alliez, D. Cohen-Steiner, M. Desbrun, Designing quadrangulations with discrete harmonic forms, in: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, in: SGP '06, Eurographics Association, 2006, pp. 201–210.

[31] N. Ray, B. Vallet, W.C. Li, B. Lévy, N-symmetry direction field design, ACM Trans. Graph. 27 (2) (2008) 10.

[32] M. Jin, X. Gu, Y. He, Y. Wang, Conformal Geometry: Computational Algorithms and Engineering Applications, Springer, 2018.

[33] D. Bommes, H. Zimmer, L. Kobbelt, Mixed-integer quadrangulation, ACM Trans. Graph. (TOG) 28 (3) (2009) 1–10.

[34] D. Bommes, H. Zimmer, L. Kobbelt, Practical mixed-integer optimization for geometry processing, in: International Conference on Curves and Surfaces, Springer, 2010, pp. 193–206.

[35] O. Diamanti, A. Vaxman, D. Panozzo, O. Sorkine-Hornung, Integrable polyvector fields, ACM Trans. Graph. 34 (4) (2015) 38.

[36] D. Panozzo, E. Puppo, M. Tarini, O. Sorkine-Hornung, Frame fields: Anisotropic and non-orthogonal cross fields, ACM Trans. Graph. 33 (4) (2014) 134.

[37] M. Campen, H. Shen, J. Zhou, D. Zorin, Seamless parametrization with arbitrary cones for arbitrary genus, ACM Trans. Graph. 39 (1) (2019).

[38] M. Zhang, J. Huang, X. Liu, H. Bao, A wave-based anisotropic quadrangulation method, ACM Trans. Graph. 29 (4) (2010) 118:1–118:8.

[39] A. Myles, N. Pietroni, D. Zorin, Robust field-aligned global parametrization, ACM Trans. Graph. 33 (4) (2014) 135.

[40] H.-C. Ebke, D. Bommes, M. Campen, L. Kobbelt, QEx: robust quad mesh extraction, ACM Trans. Graph. 32 (6) (2013) 168.

[41] F. Knöppel, K. Crane, U. Pinkall, P. Schröder, Globally optimal direction fields, ACM Trans. Graph. (ToG) 32 (4) (2013) 59.

[42] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Lévy, Polygon Mesh Processing, AK Peters/CRC Press, 2010.

[43] M. De Berg, M. Van Kreveld, M. Overmars, O. Schwarzkopf, Computational geometry, in: Computational Geometry, Springer, 1997, pp. 1–17.

[44] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.

[45] Y. Soliman, D. Slepčev, K. Crane, Optimal cone singularities for conformal flattening, ACM Trans. Graph. 37 (4) (2018).

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

[47] R.R. Hiemstra, T. Hughes, C. Manni, H. Speleers, D. Toshniwal, A Tchebycheffian extension of multi-degree B-splines: Algorithmic computation and properties, SIAM Journal on Numerical Analysis, 58 (2) (2020) 1138–1163.

[48] D. Toshniwal, H. Speleers, R.R. Hiemstra, T.J. Hughes, Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 316 (2017) 1005–1061.

[49] D. Toshniwal, H. Speleers, R.R. Hiemstra, C. Manni, T.J. Hughes, Multi-degree B-splines: Algorithmic computation and properties, Comput. Aided Geom. Design 76 (2020) 101792.

[50] D.C. Thomas, M.A. Scott, J.A. Evans, K. Tew, E.J. Evans, Bézier projection: a unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis, Comput. Methods Appl. Mech. Engrg. 284 (2015) 55–105.

[51] L. Piegl, W. Tiller, The NURBS Book, Springer Science &amp; Business Media, 2012.

[52] T.W. Sederberg, D.C. Anderson, R.N. Goldman, Implicit representation of parametric curves and surfaces, Comput. Vis. Graph. Image Process. 28 (1) (1984) 72–84.

[53] D. Benson, Y. Bazilevs, M.-C. Hsu, T. Hughes, A large deformation, rotation-free, isogeometric shell, Comput. Methods Appl. Mech. Engrg. 200 (13–16) (2011) 1367–1378.

[54] D.J. Benson, Y. Bazilevs, M.C. Hsu, T.J.R. Hughes, Isogeometric shell analysis: the Reissner–Mindlin shell, Comput. Methods Appl. Mech. Engrg. 199 (5–8) (2010) 276–289.

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

[56] Y. Hu, T. Schneider, X. Gao, Q. Zhou, A. Jacobson, D. Zorin, D. Panozzo, Triwild: robust triangulation with curve constraints, ACM Trans. Graph. 38 (4) (2019) 52.

[57] Y. Hu, Q. Zhou, X. Gao, A. Jacobson, D. Zorin, D. Panozzo, Tetrahedral meshing in the wild, ACM Trans. Graph. 37 (4) (2018) 60:1–60:14.

[58] N. Ray, W.C. Li, B. Lévy, A. Sheffer, P. Alliez, Periodic global parameterization, ACM Trans. Graph. 25 (4) (2006) 1460–1485.

[59] M. Campen, D. Zorin, Similarity maps and field-guided T-splines: a perfect couple, ACM Trans. Graph. 36 (4) (2017) 1–16.