



HAL
open science

From CAD to Representations Suitable for Isogeometric Analysis: a Complete Pipeline

Michelangelo Marsala, Angelos Mantzaflaris, Bernard Mourrain, Sam Whyman, Mark Gammon

► To cite this version:

Michelangelo Marsala, Angelos Mantzaflaris, Bernard Mourrain, Sam Whyman, Mark Gammon. From CAD to Representations Suitable for Isogeometric Analysis: a Complete Pipeline. *Engineering with Computers*, 2024, 10.1007/s00366-024-02065-0 . hal-04185850v2

HAL Id: hal-04185850

<https://hal.science/hal-04185850v2>

Submitted on 19 Sep 2024

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

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

From CAD to Representations Suitable for Isogeometric Analysis: a Complete Pipeline

Michelangelo Marsala^{a,c,*}, Angelos Mantzaflaris^a, Bernard Mourrain^a, Sam Whyman^b, Mark Gammon^b

^aInria Centre at Université Côte d'Azur, Sophia Antipolis, France

^bInternational TechneGroup Ltd, Cambridge, UK

^cDepartment of Mathematics and Informatics "U. Dini", University of Florence, Florence, Italy

Abstract

Next generation CAD/CAE systems shall integrate design and analysis procedures as a means to simplify workflows, boost performance, and speed up time to market. In the present paper we demonstrate a proof of concept of such a unification using subdivision surfaces and the enabling technology of isogeometric analysis. In particular, we present a complete pipeline to convert CAD models into smooth G^1 spline representations, which are suitable for isogeometric analysis. Starting from a CAD boundary representation of a mechanical object, we perform an *automatic* control cage extraction by means of quadrangular faces, such that its limit Catmull-Clark subdivision surface approximates accurately the input model. We then compute a basis of the G^1 spline space over the quad mesh in order to carry out least squares fitting over a point cloud, acquired by sampling the original CAD geometry. The resulting surface is a collection of Bézier patches with G^1 regularity. Finally, we use the basis functions to perform isogeometric analysis simulations of realistic PDEs on the reconstructed G^1 model. The quality of the construction is demonstrated via several numerical examples performed on a collection of CAD objects presenting various challenging realistic shapes.

Keywords: CAD models, point cloud fitting, geometrically smooth surfaces, spline basis, isogeometric analysis

1. Introduction

Shape modeling and analysis are crucial operations which directly impact engineering and industrial processes in many sectors of our society. The last several decades have witnessed the development of many powerful tools for computer-aided design (CAD), computer-aided engineering (CAE), and computer-aided manufacturing (CAM). These tools assist in handling the complex computations required to convert from the digital model of a shape to its actual production. Such computations can include digital shape description, model repair, meshing, numerical simulations, and optimization. Currently, they require specific engineering efforts, are time consuming, and prone to errors and approximations [1], [2, Chapter 1]. This explains why alternative approaches are under investigation.

The boundary representation (B-rep) is used widely in solid modeling applications due to its flexibility and precision when representing manufacturable geometry in mechanical CAD (MCAD) processes. B-rep models consist of a set of *topological* entities with the following hierarchical relationship: vertices; edges bound by vertices; and faces bound by edges. The valence (i.e., number of incident edges) of the vertices can vary, and may not be considered to be regular (i.e., valence of 4). The topological entities are associated with corresponding geometric information. For instance, each topological edge has a spline curve associated with it. Similarly, each topological face has a parametric surface associated with it (herein referred to as the *embedding surface*). We will assume these to be NURBS surfaces, though analytic representations are also common. Figure 1 shows in (a) an example of a typical MCAD B-rep model and in (b) its separate NURBS surfaces as the embedding geometry for each face.

Whilst a B-rep representation prescribes definitive topological relationships between geometric entities, there is no guarantee of geometric fidelity. The embedding NURBS surfaces of topologically neighboring faces may be *discontinuous* to an arbitrary extent at their interface. Similarly, the spline curves associated with the edges might not necessarily conform to the NURBS patches incident on the edge. The combination of these factors is referred to

*Corresponding author

as *geometric sloppiness* (see schematic in Figure 2), the severity of which depends on the design process employed within the CAD system. Furthermore, the topology of the B-rep can be arbitrarily complex, often for reasons no other than being an artefact of the design process. The side effect of such complexity is often the emergence of poor quality geometry, in the form of sliver faces.

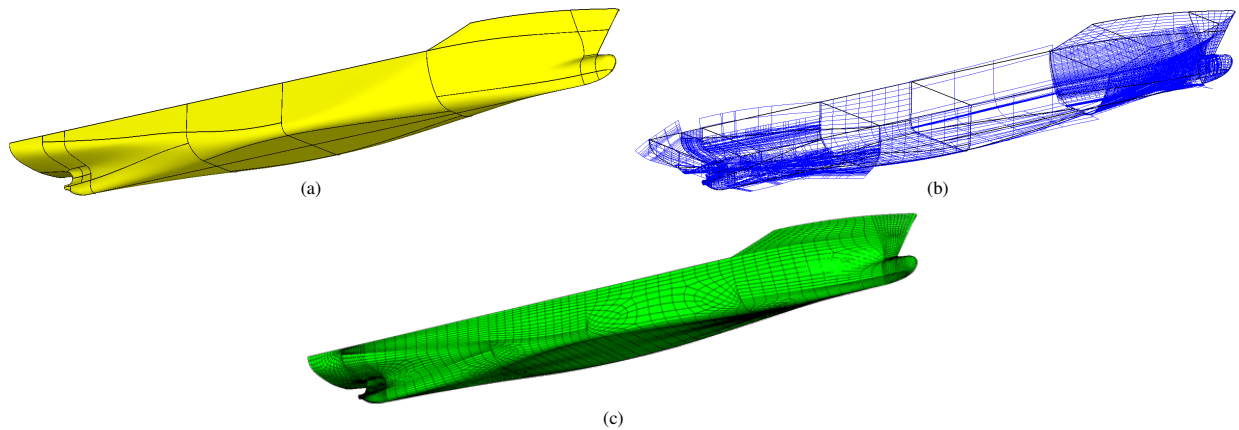


Figure 1: (a): shaded MCAD B-rep faces of the KCS ship hull model, with edges shown in black wireframe. (b): embedding NURBS patches for each face. (c): subdivision surface approximation of the KCS ship hull.

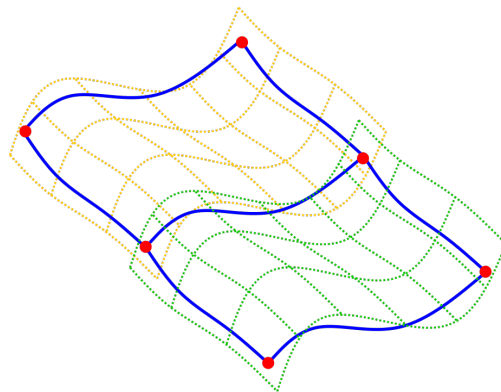


Figure 2: Demonstration of geometric sloppiness inherently present in MCAD representations. The MCAD vertices are represented in red, the edges in blue, and the embedding NURBS surfaces in green and orange. In this example, the edges do not conform to the surfaces.

The combination of geometric sloppiness and complex (or excessive) topology can present serious challenges for CAE applications and are often a major bottleneck to an effective design pipeline [3].

Therefore, there is significant appeal in simplifying both the topology and geometry of the model (or even select parts of the model) into an alternative representation which is more suitable for the specific application. This is a concept known as *hybrid modeling*, and allows the benefits of multiple different representations to be exploited. In an effective hybrid modeling engine, a direct link between the alternate representation and the original MCAD B-rep should always be maintained. This is especially important during the design of manufacturable parts, where MCAD B-rep is considered the standard format.

Subdivision surfaces [4, 5] are one example of an alternative geometry type which has benefits over traditional NURBS patches due to their arbitrary topology via the introduction of extraordinary vertices (EVs). This means that a *single* subdivision surface can be used to represent an entire B-rep body comprised of many contiguous faces, with arbitrary topological complexity. This contrasts with the strictly rectangular nature of NURBS which dictates the need for multiple adjacent patches. Such alternative representation may be of particular interest for design optimization processes [6], wherein subdivisions surfaces may act as a proxy to the MCAD geometry for convenient manipulation. Within a hybrid modeling framework, subdivision surfaces may be treated just as any other embedding geometry for

a B-rep face. This is crucial for the hybrid concept as it allows full compatibility with pre-existing operations offered by geometry engine software.

The goal of hybrid modeling is not to replace B-rep, but to complement it. Subdivision surfaces offer an intermediate representation which allows for convenient local modification to the geometry via adjustment of relatively few control points. Other representations may be more suitable when, for example, topological adjustments are desired, such as implicit representations, but an effective hybrid modeling engine should use a combination of different representations, exploiting the strengths of each.

One of the limitations of the widely used Catmull-Clark scheme [7] is in how the limit surface behaves at boundaries. Many implementations, such as the `OpenSubdiv` library from Pixar, allow the application of *sharp* or *semi-sharp* creases to the boundary (or the interior) of the subdivision surface. These tools allow the creation of features such as fillets and chamfers without the need for introducing additional control cage vertices. However, these schemes do not allow for any limit surface curvature in the direction orthogonal to the boundary (referred to as *cross-curvature*) [8]. In the fields of computer graphics and animation, this is rarely an obstacle to achieving results. For engineering purposes however, this does not permit accurate modeling of real geometries which exhibit genuine cross-curvature at the boundary. Furthermore, we generally require precise control over the external boundaries of the limit surface so that they behave more similarly to MCAD edges and can be manipulated as such. These requirements for *engineering-grade* subdivision surfaces can be achieved by the addition of Bézier edge conditions [9].

To analyze the mechanical behavior of an MCAD model, a classical approach consists of meshing the shape from its B-rep description. Then, standard finite element methods (FEM) can be employed to run numerical simulations. Computing meshes from MCAD B-rep, which are suitable for numerical simulation, is not trivial due to the sloppiness of the geometric description, and the trimmed NURBS patchwork nature of the shape. Moreover, the meshing process produces piecewise linear approximations of the shape, which may require expensive mesh refinement to achieve sufficient accuracy in regions of high curvature. This is an obstacle for the development of high order numerical methods [10]. In recent years the alternative approach of isogeometric analysis (IGA) has been proposed to circumvent these difficulties [2]. Rather than involving an expensive meshing approximation step, it directly exploits the piecewise B-spline parametrization of the shape and associated B-spline basis functions in order to apply B-spline based FEM. This approach allows for high order numerical methods, requiring smaller finite element spaces, at the cost of a more expensive step in assembling the mass or stiffness matrices. However, this requires the computation of spline basis functions associated with the given geometry parametrization.

1.1. Contributions

We present a new scheme for handling MCAD B-rep models, and for analyzing their mechanical and physical behavior. It can be decomposed into the following steps:

- Computation of a single control cage, guided by a subdivision surface.
- Sampling of points on the MCAD B-rep model, adapted to the control cage partitioning of the domain of arbitrary topology.
- Reconstruction of a geometrically smooth surface from the data points, using the G^1 basis functions associated with the control cage mesh.
- Isogeometric analysis of the mechanical behavior, using the G^1 basis associated with the control cage mesh.

Figure 3 provides a diagram illustrating such a pipeline. We emphasize that our process generates a G^1 smooth Bézier surface that represents a target MCAD B-rep model, while respecting the curvature of the faces, whose control cage generation is completely *automatic*, including satisfying the topological requirements for the edge conditions (cf. Section 2). Moreover, the subsequent steps of sampling and G^1 basis generation on the mesh do not require manual work, since they are based on the topology of the control cage. Therefore, our procedure demonstrates that high order isogeometric analysis is an enabling technology for next generation CAD/CAE systems.

Our approach to generating control cages guided by subdivision surfaces from an initial MCAD B-rep model employs a combination of partitioning and meshing technology [11] to produce a quad mesh control cage of the domain. The full construction process is explained in detail in Section 2.

Moreover, we also compute a basis of the G^1 functions associated with the control cage, and use it for fitting accurately the MCAD model and running high order numerical simulation via the IGA methodology.

We demonstrate the viability of our approach on real models from the automotive, shipbuilding, and aerospace industries, running the full process from the B-rep shape representation to numerical simulation.

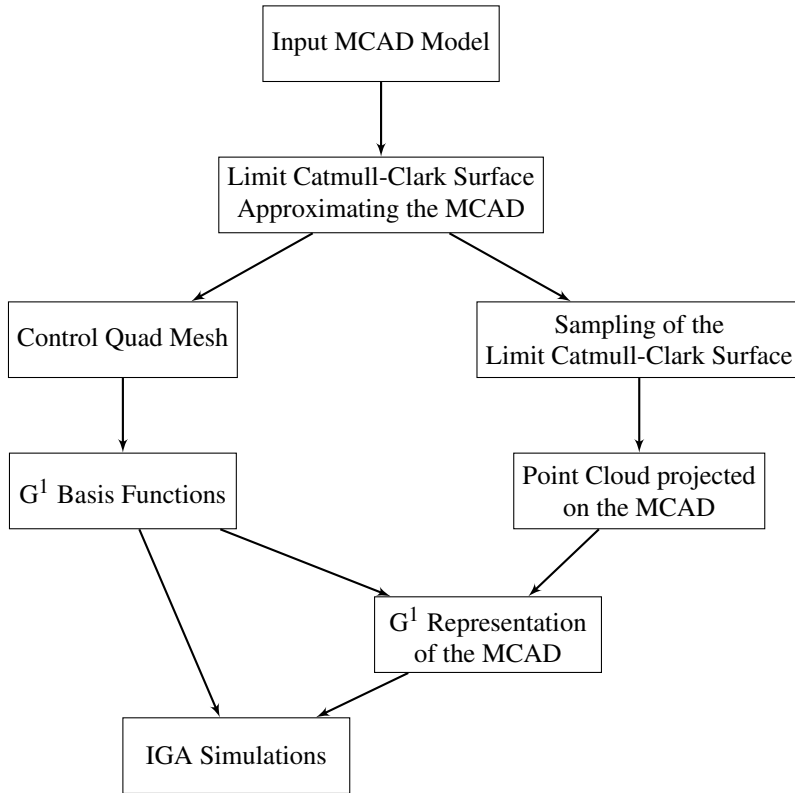


Figure 3: Schematic representation of the different steps composing the proposed pipeline.

1.2. Positioning

The need to run numerical simulation on CAD models is ubiquitous in engineering and industry. A classical approach, which has prevailed in the last few decades, is to mesh the CAD model and run standard FEM on the generated mesh. An entire suite of meshing technologies has been developed over the years, which today is commonly integrated into commercial products, or accessible in open source software, e.g., [12].

Converting a quadrilateral mesh into a spline model is a task that has been studied from multiple points of view over the last decade, especially due to the advent of isogeometric analysis. T-splines were used for this task in [13], due to their support of unstructured meshes, notably implemented in a commercial package [14]. In [15], the authors enhance T-spline construction to be C^1 at EVs and apply them to the analysis of Kirchhoff-Love shell problems. In [16], a semi-automatic frame-field guided parametrization approach is developed, which converts trimmed B-rep geometry to conforming, watertight NURBS. In [17, 18], the authors present a method to rebuild (trimmed) CAD models based on surface Ricci flow with metric optimization. In [19], the authors use the history of employing Boolean operations in the construction of CAD models to ensure that trimmed CAD model are watertight. In [20], patch surface meshes serve as a guide to generating volume meshes suitable for use in fluid-structure interaction simulations. Overall, research centered around meshing geometric models is still very active [21, 22, 23, 24]. However, the generated mesh remains an approximation of the geometric model and may require tuned refinement operations in regions with high error. This leads to complex and costly optimization computations to obtain accurate solutions in numerical simulations.

Another trend has been to approximate CAD models with higher order and more accurate representations of geometry which are simple to manipulate. Subdivision surfaces [4] and even volumes [25, 26] appeared as possible candidates due to their capacities to reproduce B-spline functions in regular regions, and to represent shapes with complex topologies. Some works have been developed to convert B-spline representations into subdivision schemes [27, 28, 29], or trimmed B-spline representation [30] to subdivision surfaces [9]. In general, the limit surface has good global approximation properties but poor geometric quality around the extraordinary vertices. Moreover, FEM

based on subdivision schemes are not straightforward to control and require advanced techniques (such as dedicated quadrature rules) to obtain the expected precision [31].

An alternative approach to achieve faithful geometric representation, and perform accurate numerical simulation, is to use high order elements both for the geometry and the FEM. An approximate conversion of a Catmull-Clark subdivision surface into a collection of bicubic B-spline patches is described in [32]. The resulting surface is not necessarily smooth. Several works have addressed the construction of smooth surfaces from (quad) meshes. See e.g., [33, 34, 35, 36, 37, 38, 39].

To complete our pipeline, we also need to compute bases of spline functional spaces over the computed geometry. The analysis of spline spaces over planar domains is well-developed, though many open problems still remain (see e.g., [40, 41, 42, 43, 44] and references therein). The analysis of G^1 spline spaces is much less investigated [45, 46, 47, 48, 49, 50, 51]. We use these recent basis constructions both to construct an accurate representation of the geometry and to obtain functional elements of high order in IGA simulations.

2. Control cage generation from MCAD geometry

In this section, we detail the construction of the control cage from the boundary representation of a model. The foundation to our approach is to represent each MCAD edge with a cubic B-spline with a multiplicity-4 knot at each end. This allows the end curvature to be controlled by so-called *slope control points* which are not themselves part of the control cage topology (Figure 4). These can equally be thought of as tangent vectors stored at the ends of the spline.

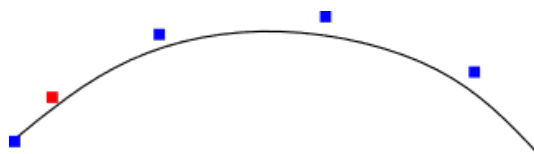


Figure 4: Schematic of a cubic B-spline with knot vector $[0,0,0,0,1,2,3,4,\dots]$. The slope control point is shown in red, and all other control points in blue. The location of the red point influences the shape of the spline only within the first two knot spans.

The limit surface is then defined to be the tensor product surface of the boundary B-splines. Since the slope control points affect only the first two knot spans, it follows that the first two layers of patches depart from the usual behavior of the regular regions of a Catmull-Clark subdivision surface (i.e., away from the EVs). This modification constitutes the addition of Bézier edge conditions to the standard Catmull-Clark scheme [9], and is illustrated by Figure 5. The 3D location of the limit surface corresponding to the boundary of the B-rep faces is influenced *only* by the control points associated with the MCAD edges. Therefore, neighboring control cages sharing the same boundary control points will maintain at least C^0 -continuity between their limit surfaces, regardless of the positions of the interior control points, or the slope control points.

The use of Bézier edge conditions imposes strict topological requirements upon the control cage. Namely, no *new* EV may be placed on a B-rep face boundary (where one was not already present in the B-rep topology, such as in Figure 11), nor within the first two layers of control cage faces, due to the tensor product nature of the limit surface within these regions.

These topological constraints, to which engineering-grade subdivision surfaces must adhere, pose challenges when generating a suitable control cage. Each MCAD vertex must be treated as a corner, i.e., it must be associated with a 2-valent control cage vertex (such as in Figure 5). The introduction of any new EV, where one was not already present in the B-rep topology, is restricted solely to the interior of the control cage. Meeting these requirements through traditional quad-dominant meshing techniques is challenging, and therefore we present a novel *automatic* approach referred to as *SubD layering*, which is tailored to satisfying the prescribed topological constraints.

The process involves partitioning an MCAD face into regions of structured and unstructured mesh. The structured regions form a *boundary layer* from which new EVs are fully excluded. These are formed by constructing 4-sided blocks around each MCAD vertex (referred to as corner blocks), and then connecting these to form 4-sided edge blocks associated with each MCAD edge. The remaining interior region constitutes a block with topology identical to the original MCAD face, and this is filled with unstructured mesh. This process is illustrated in Figure 6.

The size and positions of the corner blocks are determined using the two-dimensional medial axis [11] as a guide to ensure that the blocks do not intersect. The construction of the medial axis follows the work in [52]: the algorithm starts with a Delaunay mesh of the domain and discovers triangles that are representative of medial vertices and edges,

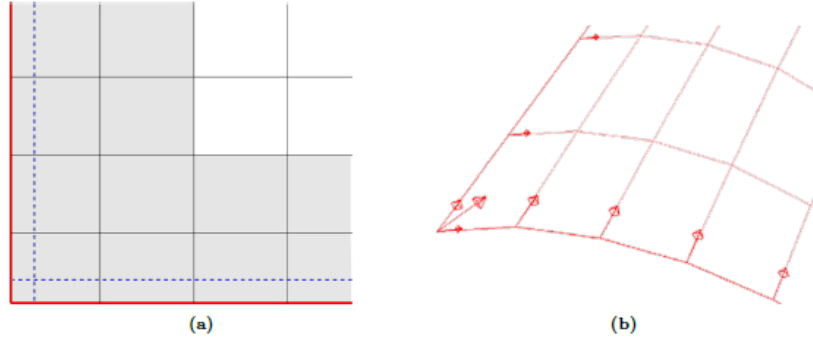


Figure 5: (a): schematic showing the application of Bézier edge conditions to a subdivision surface. The red lines represent the rows of control points which approximate the MCAD edges as cubic B-splines. The blue dashed lines represent the rows of slope control points. Shaded patches are defined using the tensor product of the boundary B-splines, and the unshaded patches are the usual regular bicubic B-spline patches. (b): 3D control cage with vectors pointing to the implied positions of the slope control points. Each boundary control point stores one vector, whereas the corners store three.

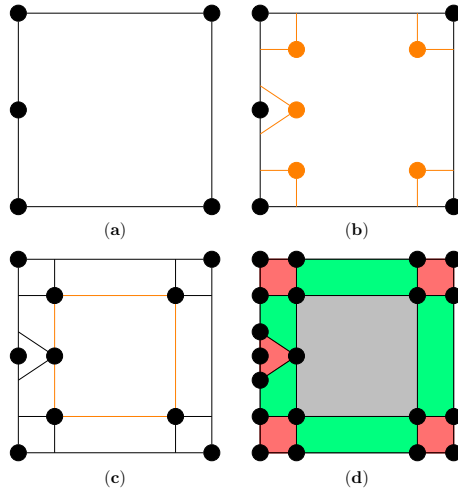


Figure 6: Schematic outlining the SubD layering process. (a): MCAD B-rep face with 5 vertices. (b): corner blocks are constructed around the MCAD vertices. (c): corners blocks are connected to form edge blocks. (d): Domain is partitioned into corner (red) and edge (green) blocks which can receive structured mesh, and one interior (gray) region which receives unstructured mesh.

while it refines triangles which are not isomorphic to the medial circle. The resulting representation is reduced to a topology of minimum size. A snapshot of a computed SubD layering and the medial axis for a face from the Car model is given in Figure 7.

Once the domain partitioning has been completed, we generate a coarse mesh of the layer faces. The boundary layer (consisting of the corner and edge blocks) is meshed using a transfinite interpolation technique [53] and is specified to be one element thick between the boundary and the interior (see Figure 6). This does not preclude increasing the mesh density *along* the length of the edge blocks, thus limiting the aspect ratio of the resulting elements. The interior region is meshed using quad-dominant meshing technology [11]. Owing to the strict placement rules of the edge and corner blocks, the resulting control cage faces can exhibit notable skewness. Furthermore, it should be noted from Figure 7 that some of the corner blocks have colinear quad edges (i.e., associated with MCAD vertices on flat edges), which may lead to degenerate Jacobians. We acknowledge that this is a current limitation of our approach, and propose the need of future research to address the problem. The entire coarse mesh (including both the boundary and interior regions) is then refined twice using the Catmull-Clark subdivision scheme. This ensures that there are no EVs within the first two layers of control cage faces from the boundary, thus satisfying the topological requirements for the Bézier edge conditions. A simple example of the refinement of a coarse mesh, serving to demonstrate the EV placement criteria, is illustrated in Figure 8. The refinement for a face from the Car model is shown in Figure 9.

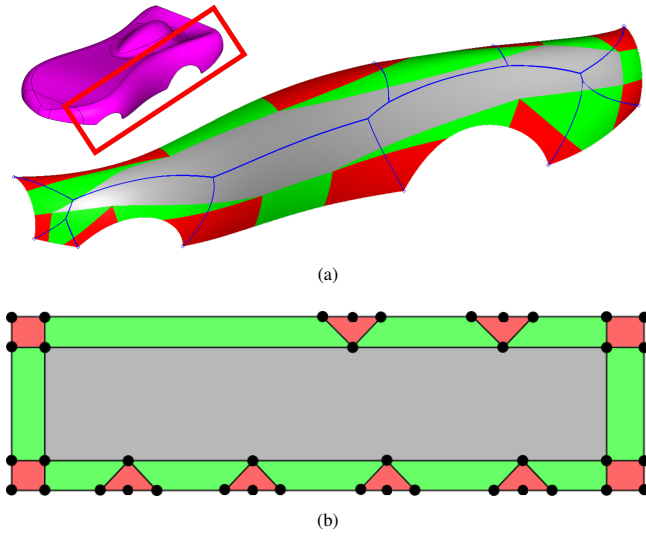


Figure 7: (a): SubD layering for the wheel arch of the Car model showing corner (red), edge (green), and unstructured (gray) regions. The 2D medial axis (blue) determines the shape of the blocks so that they do not intersect. (b): Schematic of the SubD layering within the parametric domain, which may be compared with Figure 6. Each region is subsequently meshed to generate the control cage.

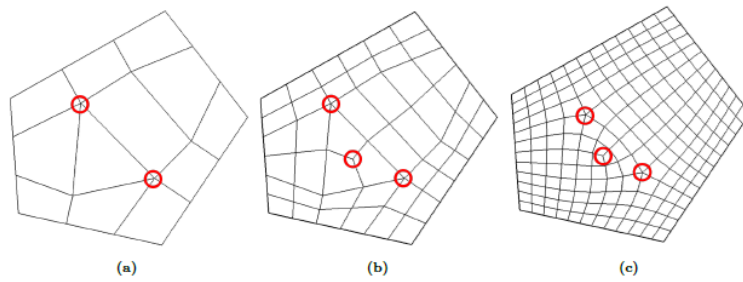


Figure 8: (a): initial coarse mesh for a 5-sided face. (b): one level of refinement. (c): two levels of refinement (with Laplacian mesh smoothing applied). The EVs are highlighted in red. After two levels of refinement the location of the EVs meet the topological requirements for the Bézier edge conditions.

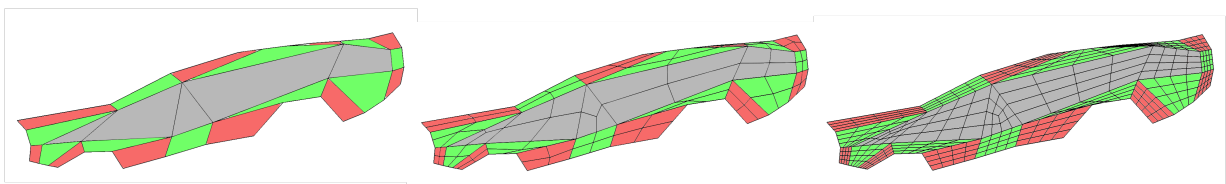


Figure 9: Mesh refinement for the wheel arch of the Car model. From left to right: initial coarse mesh; one level of refinement; two levels of refinement. The control cage faces corresponding to the corner and edge blocks are shown in red and green, respectively. The faces corresponding to the interior region is shown in gray. This may be compared with the parametric domain partitioning presented in Figure 7.

The SubD layering process is applied to each MCAD face in the model in turn, such that the resulting control cage meets the topological requirements dictated by all MCAD edges, both external and internal (i.e., connectivity 1 and 2, respectively). By stipulating that the common vertices along the edges between contiguous faces are shared, a *single* limit surface is able to represent the entire model while remaining fully watertight (i.e., C^0 continuous). This guarantee is *not* maintained for MCAD B-rep geometry as a network of (trimmed) NURBS patches. An example of a watertight limit surface is given in Figure 10. An example of the control cage computation is presented in Figure 1(c).

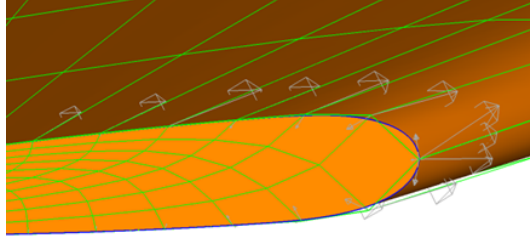


Figure 10: A single subdivision surface representing the wing tip of the SNC Dream Chaser model, which is comprised of multiple MCAD B-rep faces. The control cage is shown in green, and the limit surface is shaded in orange. Bézier edge conditions (with slope control vectors shown in gray) are applied such that contiguous regions of the limit surface meet with the same C^1 discontinuity as the original B-rep. However, the limit surface is guaranteed to be exactly C^0 continuous along the join reflecting the MCAD edges (blue).

Despite the restricted placement of EVs with respect to the B-rep face boundaries, the SubD layering construction is fully compatible with B-rep face networks of *any* topology. This includes the presence of MCAD vertices with a face valence not equal to four. This is because the limit surface corresponding to each separate face is governed solely by the edges which bound that face, and only that face. This is demonstrated in Figure 11.

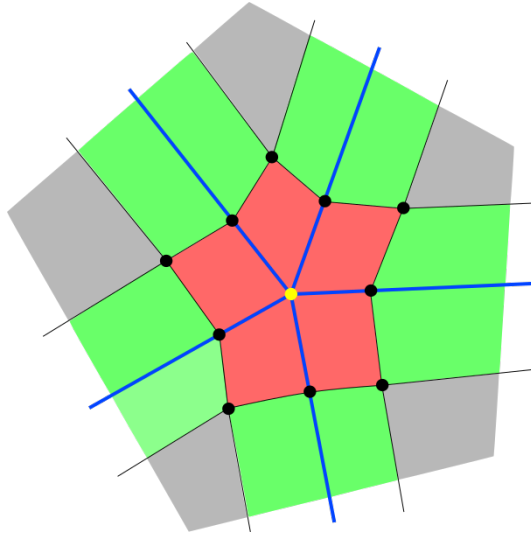


Figure 11: A schematic of the SubD layering construction centered around an MCAD vertex (yellow) with a face valence equal to five. The MCAD edges are shown in blue. The control cage vertices (black) introduced by the layering process are shared between the topologically connected faces (i.e., those coincident with the blue lines). Each red region corresponds to a section of limit surface which is governed solely by the control points belonging to that face (see Figure 5), whilst maintaining C^0 -continuity with the neighboring red regions. This is a departure from the standard Catmull-Clark scheme, wherein *each* of the control points surrounding the EV has an influence over the limit surface.

3. Control cage adjustment and point cloud sampling

The SubD layering process outlined in Section 2 is primarily focused on achieving the correct *topology* for the control cage. The second step of the process is to adjust the control cage so that its limit surface coincides with the target MCAD geometry. To achieve this, we use the ability of our engineering-grade subdivision surfaces to accurately represent geometry; the behavior of the edges is governed by the B-spline edge conditions, which can be made to respect the MCAD edges via a least squares fitting process.

Once the positions of the edge control points have been determined, they are fixed. Then, to adjust the *interior* control points, we minimize the error between a grid of sample points on the limit surface and the corresponding point grid on the MCAD geometry. This is done by computing gradients of the error and by iteratively adjusting the control

cage until a sufficiently good approximation is attained. Given a set of all control points \mathcal{X} , consider the set of sample points $\hat{\mathcal{P}}$ over the limit surface, where each sample point $\mathbf{p}_i \in \hat{\mathcal{P}}$ can be expressed as the *limit stencil*

$$\mathbf{p}_i = \sum_{\mathbf{x}_j \in C} w_j^i \mathbf{x}_j, \quad (1)$$

where $C \subset \mathcal{X}$, is the set of control points supporting \mathbf{p}_i . The weights w_j^i are functions of the local patch parameter $[s, t]$ corresponding to the limit surface evaluation point \mathbf{p}_i , such that

$$w_j^i(s, t) = [\mathbf{M}_t \mathbf{t}]_{\text{row}(j)} [\mathbf{M}_s \mathbf{s}]_{\text{col}(j)}, \quad (2)$$

where \mathbf{s} is the column vector $[1, s, s^2, s^3]$, \mathbf{t} is the column vector $[1, t, t^2, t^3]$, and \mathbf{M}_s and \mathbf{M}_t are matrices representing the cubic spline evaluation. The functions $\text{row}(j)$ and $\text{col}(j)$ map the control point index $j \in C$ to the corresponding row and column of the tensor product grid, respectively. The matrices are determined solely by the control cage topology. For example, for regular patches under the Catmull-Clark scheme the matrices \mathbf{M}_s and \mathbf{M}_t take the form,

$$\mathbf{M}_s = \mathbf{M}_t = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

For patches influenced by the Bézier edge conditions (see Section 2), where $\mathbf{M}_s \neq \mathbf{M}_t$, the matrices are determined using the De Boor algorithm [54]. For patches containing EVs, we employ the evaluation method of [55]. Once the appropriate matrices have been determined, the weights may subsequently be evaluated conveniently for any given parameter value $[s, t]$ according to (2).

The set $\hat{\mathcal{P}}$ is constructed by iterating over each control cage face and appending a uniform grid of local patch parameters. Each sample point \mathbf{p}_i on the limit surface is assigned a corresponding target location \mathbf{q}_i on the MCAD geometry. This is achieved via bilinear interpolation over the parametric sub-domain of the MCAD face which is bounded by the control cage face. The MCAD face is evaluated at the interpolated coordinate to find \mathbf{q}_i .

A gradient vector δ_i for each control point in \mathcal{X} is defined to be

$$\delta_i = \frac{1}{|\mathcal{A}_i|} \sum_{\mathbf{p}_j \in \mathcal{A}_i} (\mathbf{q}_j - \mathbf{p}_j) w_j^i, \quad (4)$$

where \mathcal{A}_i denotes the set of sample points which are influenced by control point \mathbf{x}_i . One round of control point adjustment equates to applying the operation $\mathbf{x}_i = \mathbf{x}_i + \delta_i, \forall \mathbf{x}_i \in \mathcal{X}$. This procedure avoids the need to solve a large linear system and instead relies on local updates of the control points at hand. We stress that this fitting is an intermediate result obtained prior to the subsequent least-squares fitting of the G^1 spline surfaces. As such, we only require an adequate approximation of the MCAD geometry. We found that applying 1000 rounds of control cage adjustment was sufficient in the majority of cases; nonetheless, we acknowledge that this is an area for future improvement by means of introducing robust convergence criteria. The fitting results are demonstrated in Figure 12, which shows the front section of the NASA CRM.

The procedure concerning the fitting for our multipatch G^1 spline representation combines the generation of points on the geometry with a standard regression between the sample points and the parametric points on the subdivision surface. We require a set of samples points which lie exactly on the geometry, including the edges. Each point must be mapped to a corresponding patch of the subdivision surface, together with a local patch parameter coordinate. This is achieved by sampling each patch of the limit surface at a predetermined set of parameter values, using explicit evaluation [55]. The corresponding 3D positions on the limit surface (which *approximates* the geometry) are then *projected back* onto the MCAD geometry. Given that the limit surface samples already lie very close to the target geometry, they may be projected easily onto the embedding NURBS surfaces of the MCAD B-rep faces. We also apply additional heuristics such as mean value interpolation when the displacements are large, which aims at preventing creases and folds appearing in highly curved regions. These measures also serve to reduce the impact of geometric sloppiness and poor quality B-rep, ensuring that the sample grids for neighboring faces meet without gaps or folds. The result is a uniform non-folding structured grid of sample points lying on the target geometry, for each patch of the limit surface. Figure 13 shows an example of the typical distribution of the point cloud sampling.

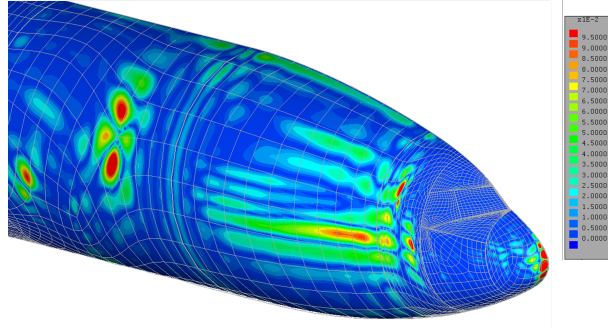


Figure 12: Heat map of the distance from the limit surface to the target MCAD geometry, for the forward section of the NASA CRM. A maximum fitting error of approximately 0.005% of the aircraft length is achieved after 1000 rounds of iterative control cage adjustment.

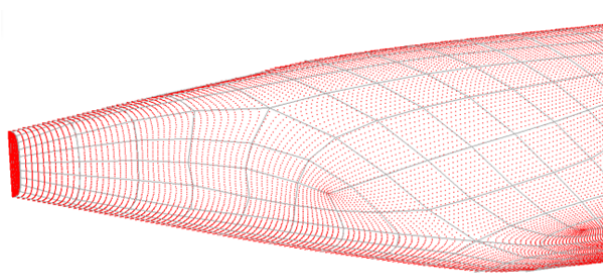


Figure 13: An example of the distribution of points in the sampling of the MCAD geometry, for the tail of the NASA CRM. Each patch of the subdivision surface receives a uniform grid of samples, which are projected onto the embedding surfaces of the MCAD B-rep faces.

4. G^1 functions on quad meshes

To apply least squares fitting and IGA methods, we need to define a function space of regular functions such as a spline space. In this section, we briefly introduce some basic definitions and the tools required to build spaces of geometrically smooth functions over a quad mesh \mathcal{M} .

4.1. Definition of G^1 functions

Let $f = (f_\sigma)_{\sigma \in \mathcal{M}}$ be a collection of functions defined over the faces σ of a quad mesh \mathcal{M} . All faces $\sigma \in \mathcal{M}$ have parametric domain $[0, 1]^2$. We name e the common edge shared by two adjacent patches $f^L = f|_{\sigma_L}$ and $f^R = f|_{\sigma_R}$ of a

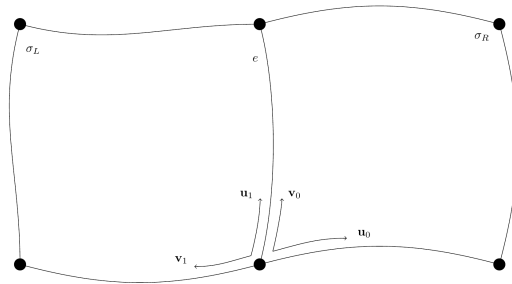


Figure 14: Local coordinate systems between two adjacent patches.

collection f . With reference to the local systems' orientation in Figure 14, two functions are said to be G^1 (or tangent plane continuous) if they join G^0 (or C^0), i.e.,

$$f^L(u_1, 0) = f^R(0, u_1), \quad u_1 \in [0, 1], \quad (5)$$

and there exists three function $a, b, c : e \rightarrow \mathbb{R}$ called *gluing functions* such that

$$f_{v_1}^L(u_1, 0)c(u_1) + f_{u_1}^R(0, u_1)b(u_1) + f_{v_0}^L(0, u_1)a(u_1) = 0, \quad (6)$$

$$c(u_1)b(u_1) < 0, \quad (7)$$

$$\langle f_{v_1}^L(u_1, 0), f_{v_0}^R(0, u_1) \rangle \neq 0, \quad (8)$$

where $u_1 \in [0, 1]$, $f_u = \frac{\partial f}{\partial u}$ and $\langle \cdot | \cdot \rangle$ refers to the standard Euclidean scalar product. In detail, (6) means that the tangent plane across the common edge e of f^L and f^R is the same, (8) signifies that the tangent plane has no vanishing normal, and (7) controls the orientation of the patches in order to avoid cusps. We refer the reader to [56] for further details.

A particular spline space defined over quad meshes and verifying eqs. (5) to (8) is the G^1 ACC derived in [39]; it is obtained from a collection of biquintic Bézier functions by imposing G^1 continuity across extraordinary regions making use of the *quadratic* gluing functions

$$a(u) = a_0 B_2^0(u) - a_2 B_2^2(u),$$

$$b(u) = -1,$$

$$c(u) = 1,$$

with $B_d^k(u) = \binom{d}{k} u^k (1-u)^{d-k}$ the univariate Bernstein polynomials and $a_i = 2 \cos(2\pi/N_i)$, $i = 0, 2$ where N_i represent the valence of the two vertices belonging to the common edge (i.e., the number of edges attached to them).

More generally, C^1 continuity of surfaces can be obtained from G^1 (geometric) continuity after some regular reparametrization of adjacent pieces (i.e., patches sharing the edge e), so that $f^L = f^R \circ \phi_{\sigma_R, \sigma_L}$. In our case, the reparametrization is defined as

$$\phi_{\sigma_R, \sigma_L} : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad (u_1, v_1) \mapsto (u_0, v_0) = \begin{pmatrix} v_1 b(u_1) \\ u_1 + v_1 a(u_1) \end{pmatrix}, \quad \text{where} \quad b(u) = \frac{b(u)}{c(u)}, \quad a(u) = \frac{a(u)}{c(u)}.$$

The relations for the tangent plane continuity constraints between the control points $\mathbf{b}_{i,j}$, $i, j = 0, \dots, 5$ of two neighboring Bézier patches f^L, f^R defining the G^1 ACC space are the following:

$$\begin{aligned} b_{0,1}^L + b_{1,0}^R &= \bar{a}_0 b_{0,0}^L + a_0 b_{1,0}^L, \\ 5(b_{1,1}^L + b_{1,1}^R) &= a_0 b_{0,0}^L + 5\bar{a}_0 b_{1,0}^L + 4a_0 b_{2,0}^L, \\ 10(b_{2,1}^L + b_{1,2}^R) &= -a_0 b_{0,0}^L + 5a_0 b_{1,0}^L + 10\bar{a}_0 b_{2,0}^L + 6a_0 b_{3,0}^L, \\ 10(b_{3,1}^L + b_{1,3}^R) &= a_0 b_{0,0}^L - 5a_0 b_{1,0}^L + 10a_0 b_{2,0}^L + 10\bar{a}_0 b_{3,0}^L + 4a_0 b_{4,0}^L, \\ b_{4,1}^L + b_{1,4}^R &= 2b_{4,0}^L, \\ b_{5,1}^L + b_{1,5}^R &= 2b_{5,0}^L, \\ 10(b_{3,0}^L - b_{2,0}^L) &= b_{0,0}^L - 5b_{1,0}^L + 5b_{4,0}^L - b_{5,0}^L, \end{aligned}$$

with $\bar{a}_0 = 2 - a_0$. The above system of G^1 relations can be solved by following the approach in [39].

4.2. Construction

The aim of this step of the pipeline is to obtain a representation of an MCAD model in terms of smooth G^1 functions. In Section 3 it has been shown how to discretize an MCAD model as a dense point cloud preserving its features, while Section 2 presents a method for the automatic generation of a control cage (i.e., a quad mesh) approximating the MCAD. Therefore, the idea is to use the construction in [51] to obtain a multipatch G^1 spline representation via point cloud fitting using basis functions defined over the control cage supporting the data points. These bases are obtained by performing an extraction procedure which returns the control points defining the basis function in the Bézier form. More precisely, to get the Bézier coefficient of the bases, we fix the value of a control point appearing in the equations defining the G^1 ACC in Section 4.1 to be, for example, 1, and all the remaining free coefficients in the system to be 0. Hence, with these initial values we solve all the equations defining the G^1

constraints and as a result of this operation, we obtain the control points for our basis functions. Repeating for all the Bézier points involved in the system results in the entire set of bases generating the G^1 ACC spline space. The resulting bases set can be decomposed as a direct sum of three subsets formed by particular functions attached to the different features of a quad mesh; these sets are the so-called vertex bases set, edge bases set and face bases set. As suggested by the names, the vertex bases set contains basis functions whose support lies on the patches attached to a vertex. In presence of an EV, i.e., a vertex with valence $N \neq 4$, the resulting set of bases is composed of $N + 3$ elements, while for a regular vertex the corresponding space consists of 4 functions. Belonging to the edge bases set are all those functions whose support is contained in the two patches attached to a specific edge of the mesh. We will have 2 basis functions for each extraordinary edge (that is, an edge sharing an EV) and 4 for each regular and boundary edge. Lastly, we find in the face vertex set the basis functions whose support is entirely contained within a single patch. Part of this set are the face and corner basis functions, which appear in groups of 4 each. We refer the reader to [51] for a detailed analysis of the basis functions and their construction.

4.3. G^1 spline fitting

Consider a point cloud \mathcal{P} , which is a collection of n_p points $\mathbf{P}_i \in \mathbb{R}^3$, $i = 1, \dots, n_p$ with associated parameters $\xi^i = (\xi_1^i, \xi_2^i) \in \mathbb{R}^2$ on the face σ_ℓ , and the set of G^1 basis functions $B_j \in \mathcal{B}$ containing n_b elements (presented in Section 4.2) defined over the control cage \mathcal{M} formed by n_f faces generated with the strategy described in Section 2. The idea of the least squares fitting technique is to find the coefficients $\mathbf{c} = \{c_j\}_{j=1}^{n_b}$, $c_j \in \mathbb{R}^3$, defining a spline surface

$$S(\xi; \sigma) = \sum_{j=1}^{n_b} c_j B_j(\xi; \sigma), \quad \sigma \in \mathcal{M},$$

such that the squared distance from the set of points is minimal, that is, obtained by computing

$$\min_{\mathbf{c}} \sum_{i=1}^{n_p} \left\| \sum_{j=1}^{n_b} c_j B_j(\xi^i; \sigma_\ell) - \mathbf{P}_i \right\|_2^2 + \lambda E_{thin}(\mathbf{c}), \quad \lambda \geq 0. \quad (9)$$

We also take into account in our minimization problem (9) an energy term given by the standard thin-plate energy

$$E_{thin}(\mathbf{c}) = \sum_{\ell=1}^{n_f} \iint_{[0,1]^2} \left\| S_{\xi_1 \xi_1}(\xi; \sigma_\ell) \right\|_2^2 + 2 \left\| S_{\xi_1 \xi_2}(\xi; \sigma_\ell) \right\|_2^2 + \left\| S_{\xi_2 \xi_2}(\xi; \sigma_\ell) \right\|_2^2 d\xi,$$

where $S_{\xi_r \xi_s} = \frac{\partial^2}{\partial \xi_r \partial \xi_s} S(\xi; \sigma)$. Having such a term is useful to control and avoid possible unpleasant oscillations that might arise from the fitting. Note that the mesh is not modified during the fitting procedure, and the point cloud \mathcal{P} is the MCAD sampling given in Section 3.

We can now compute the fitted surface S by solving the minimization problem given in (9). This is a least squares problem, whose solution is obtained by solving a linear system. In order to investigate the quality of the fitting we compute the following error indicators:

$$L^\infty := \max_{i=1, \dots, n_p} \|S(\xi^i) - \mathbf{P}_i\|_2, \quad \text{RMSE} := \sqrt{\frac{1}{n_p} \sum_{i=1}^{n_p} \|S(\xi^i) - \mathbf{P}_i\|_2^2}, \quad (10)$$

which represent, respectively, the maximum ℓ_2 distance and the so-called root mean squared error (RMSE). Section 6 presents several numerical experiments showing the quality of the surfaces obtained with the use of our multipatch G^1 basis functions.

5. Isogeometric Analysis

Isogeometric analysis (IGA) is a highly efficient technique for solving PDEs numerically. Its basic idea, presented in [2] (which unifies the FEM approach and Computer Aided Geometric Design) is to use the *same* basis functions

for both reproducing exactly the computational domain, *and* for the numerical approximation of the PDE. Here we focus on solving the heat equation and the biharmonic flow equation on a 2-manifold $\Omega \subset \mathbb{R}^3$, which is defined by our MCAD model. Hence, we need to extend the standard Laplace operator to the Laplace-Beltrami operator, which is necessary when dealing with manifolds. To do that, we define the geometry map

$$\mathbf{x} : \widehat{\Omega} \rightarrow \mathbb{R}^3, \quad \boldsymbol{\xi} = (\xi_1, \xi_2) \mapsto \mathbf{x}(\boldsymbol{\xi}),$$

with $\widehat{\Omega} = [0, 1]^2$, which defines our manifold by means of a mapping from the parametric domain $\widehat{\Omega}$ into the physical space \mathbb{R}^3 . The Jacobian of the mapping \mathbf{J} , i.e.,

$$\widehat{\mathbf{J}} : \widehat{\Omega} \rightarrow \mathbb{R}^{3 \times 2}, \quad \boldsymbol{\xi} \mapsto \widehat{\mathbf{J}}(\boldsymbol{\xi}), \quad \widehat{J}_{i,j}(\boldsymbol{\xi}) := \frac{\partial x_i}{\partial \xi_j}(\boldsymbol{\xi}),$$

is used to define the *first fundamental form* of the mapping $\widehat{\mathbf{G}}$ defined as

$$\widehat{\mathbf{G}} : \widehat{\Omega} \rightarrow \mathbb{R}^{2 \times 2}, \quad \boldsymbol{\xi} \mapsto \widehat{\mathbf{G}}(\boldsymbol{\xi}), \quad \widehat{\mathbf{G}}(\boldsymbol{\xi}) := \widehat{\mathbf{J}}(\boldsymbol{\xi})^T \widehat{\mathbf{J}}(\boldsymbol{\xi}),$$

together with its determinant \widehat{g} , which is

$$\widehat{g} : \widehat{\Omega} \rightarrow \mathbb{R}, \quad \boldsymbol{\xi} \mapsto \widehat{g}(\boldsymbol{\xi}), \quad \widehat{g}(\boldsymbol{\xi}) := \sqrt{\det(\widehat{\mathbf{G}}(\boldsymbol{\xi}))}.$$

Finally, we have all the tools to define the gradient operator on the manifold Ω in parametric coordinates,

$$\nabla_{\Omega} \varphi(\mathbf{x}) = \left[\widehat{\mathbf{J}}(\boldsymbol{\xi}) \widehat{\mathbf{G}}^{-1}(\boldsymbol{\xi}) \widehat{\nabla} \widehat{\varphi}(\boldsymbol{\xi}) \right] \circ \mathbf{x}^{-1}(\boldsymbol{\xi}), \quad (11)$$

as well as the Laplace-Beltrami operator of a sufficiently smooth function φ associated to the manifold Ω ,

$$\Delta_{\Omega} \varphi(\mathbf{x}) = \left[\frac{1}{\widehat{g}(\boldsymbol{\xi})} \widehat{\nabla} \cdot \left(\widehat{g}(\boldsymbol{\xi}) \widehat{\mathbf{G}}^{-1}(\boldsymbol{\xi}) \widehat{\nabla} \widehat{\varphi}(\boldsymbol{\xi}) \right) \right] \circ \mathbf{x}^{-1}(\boldsymbol{\xi}),$$

where $\widehat{\varphi}(\boldsymbol{\xi}) = \varphi(\mathbf{x}(\boldsymbol{\xi}))$ and $\widehat{\nabla} \cdot, \widehat{\nabla}$ identify, respectively, the gradient and the divergence operators in the parametric space. For a more precise explanation of operators for isogeometric solutions to PDEs on manifolds, we refer the reader to [57].

We can therefore formulate the (strong) Cauchy problem for the heat equation. Let Ω be a manifold; find $u \in C^2(\Omega) \times C^1(\mathbb{R}_+)$ such that

$$\begin{cases} \frac{\partial u}{\partial t}(\mathbf{x}, t) = c^2 \Delta_{\Omega} u(\mathbf{x}, t), & (\mathbf{x}, t) \in \Omega \times (0, T], \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}, t) = u_D(\mathbf{x}, t), & (\mathbf{x}, t) \in \partial\Omega \times (0, T], \end{cases} \quad (12)$$

with Δ_{Ω} the Laplace-Beltrami operator, $c, T > 0$ and $u_0(\mathbf{x}), u_D(\mathbf{x}, t)$ given initial data. The weak formulation of the problem (12) with reference to the parametric space, which will be the target of our IGA simulation, can be obtained with the use of the following form and operator:

$$\widehat{a}_1(\widehat{v}, \widehat{w}) = c^2 \int_{\widehat{\Omega}} \widehat{\nabla} \widehat{v} \cdot \widehat{\mathbf{G}}^{-1} \cdot \widehat{\nabla} \widehat{w} \widehat{g} \, d\boldsymbol{\xi}, \quad \widehat{b}_1(\widehat{v}, \widehat{w}) = \int_{\widehat{\Omega}} \widehat{v} \widehat{w} \widehat{g} \, d\boldsymbol{\xi}.$$

Thus, the weak form of the heat equations can be formulated as: find $u \in H^1(\Omega)$, where $u = (\widehat{u} \circ \mathbf{x}^{-1})(\boldsymbol{\xi})$, such that for almost all $t \in (0, T)$,

$$\begin{cases} \widehat{a}_1(\widehat{u}(\boldsymbol{\xi}, t), \widehat{w}) = \widehat{b}_1(\partial_t \widehat{u}(\boldsymbol{\xi}, t), \widehat{w}), & (\boldsymbol{\xi}, t) \in \widehat{\Omega} \times (0, T], \\ \widehat{u}(\boldsymbol{\xi}, 0) = u_0(\mathbf{x}), & \boldsymbol{\xi} \in \widehat{\Omega}, \widehat{\mathbf{x}}(\boldsymbol{\xi}) \in \Omega, \\ \widehat{u}(\boldsymbol{\xi}, t) = u_D(\mathbf{x}, t), & (\boldsymbol{\xi}, t) \in \partial\widehat{\Omega} \times (0, T], \end{cases} \quad (13)$$

for every $\widehat{w} \in H^1(\widehat{\Omega})$.

Similarly to what we did for the heat equation (12), we can formulate the Cauchy problem, and consequently its weak formulation, for the so-called biharmonic flow equation defined as

$$\begin{cases} \frac{\partial u}{\partial t}(\mathbf{x}, t) = \Delta_{\Omega}^2 u(\mathbf{x}, t), & (\mathbf{x}, t) \in \Omega \times (0, T], \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}, t) = u_D(\mathbf{x}, t), & (\mathbf{x}, t) \in \partial\Omega \times (0, T], \\ \nabla u(\mathbf{x}, t) \cdot \mathbf{n} = u_N(\mathbf{x}, t), & (\mathbf{x}, t) \in \partial\Omega \times (0, T]. \end{cases} \quad (14)$$

Thus, the weak form is: find $u \in H^2(\Omega)$, $u = (\hat{u} \circ \mathbf{x}^{-1})(\xi)$ such that for almost all $t \in (0, T)$,

$$\begin{cases} \hat{a}_2(\hat{u}(\xi, t), \hat{w}) = \hat{b}_2(\partial_t \hat{u}(\xi, t), \hat{w}), & (\xi, t) \in \hat{\Omega} \times (0, T], \\ \hat{u}(\xi, 0) = u_0(\mathbf{x}), & \xi \in \hat{\Omega}, \hat{\mathbf{x}}(\xi) \in \Omega, \\ \hat{u}(\xi, t) = u_D(\mathbf{x}, t), & (\mathbf{x}, t) \in \partial\Omega \times (0, T], \\ \nabla_{\Omega} u(\xi, t) \cdot \mathbf{n} = u_N(\mathbf{x}, t), & (\mathbf{x}, t) \in \partial\hat{\Omega} \times (0, T], \end{cases} \quad (15)$$

where \mathbf{n} is the normal vector at $\mathbf{x}(\xi)$ and

$$a_2(\hat{v}, \hat{w}) = \int_{\hat{\Omega}} \frac{1}{\hat{g}} \hat{\nabla} \cdot (\hat{g} \hat{G}^{-1}(\xi) \hat{\nabla} \hat{v}) \hat{\nabla} \cdot (\hat{g} \hat{G}^{-1} \hat{\nabla} \hat{w}) d\xi, \quad b_2(\hat{v}, \hat{w}) = \int_{\hat{\Omega}} \hat{v} \hat{w} \hat{g} d\xi$$

and for every $\hat{w} \in H^2(\hat{\Omega})$.

In the experiments that follow, Dirichlet boundary conditions are enforced strongly on the G^1 -spline discretization space, whereas Neumann conditions are applied in the system matrix. Note that our variational forms are semi-discrete, and solutions in time are obtained by means of the Crank-Nicolson method.

In the weak formulation of the biharmonic operator, second derivatives appear (see [58, 59] for a detailed derivation of the weak form and its solution space). Therefore, having C^1 continuity of the basis functions involved in the numerical solution of (14) is fundamental for obtaining coherent results.

6. Experimentation

Here we report the numerical experiments to demonstrate the quality of our construction. First we present the least squares fitting procedure to reconstruct a G^1 surface from an MCAD point cloud, then we will use the previous result as the geometric domain over which to solve the heat and biharmonic equations using geometrically smooth basis functions in the IGA environment. The proposed functions will lead to optimal convergence rates in isogeometric analysis (equal to 5 for an elliptic problem in the energy norm) when inserting knots uniformly inside each patch. We emphasize that the basis functions are biquintic polynomials on every patch of the geometry, but they have a bicubic parametrization on regular patches. However, if each finer model is re-generated from the MCAD (see Section 2), the rates of convergence will be bounded by the geometry approximation error made in the process. We refer the reader to [60, Chapter 4] for detailed study and numerical experiments demonstrating this behavior.

The CAD models used for the numerical investigation are standard target examples for this type of problem. These are: Car model¹, Dream Chaser shuttle model², KCS hull model³ and the NASA CRM⁴. These are standard open domain models that originate from industrial cases.

All of them present special features and sharp edges which are a notable characteristic to be recovered in the fitted surface. The samplings of the original models are obtained following the procedure explained in Section 3. In order to obtain a precise result, the target point clouds contain large amounts of data. For the same reason, the control cage obtained from the MCAD (Section 2) presents a significant quantity of faces. The models are represented in their original scale, i.e., 1 unit = 1m. All of the numerical experiments have been performed on three different machines:

¹<https://drexel.edu/>

²<https://www.sierraspace.com/dream-chaser-spaceplane/>

³<http://www.simman2008.dk/kcs/container.html>

⁴<https://commonresearchmodel.larc.nasa.gov/>

the first has been devoted to the control cage generation and the MCAD sampling using the software CADfix [61] (C); the second ran the bases computation using Julia language (J) and consequent spline fitting using the G+Smo library [62] (G); the third has been used to compute the numerical solution of the heat and biharmonic equations on G+Smo. Respectively, their specifications are: Intel Xeon E3-1240 @ 3.70GHz, 16.0 GB RAM, 4 cores; Intel Core 7-9850 @ 2.60GHz, 16.0 GB RAM, 6 cores; and Intel Xeon Gold 6230R CPU @ 2.10GHz, 504.0 GB RAM, 104 cores.

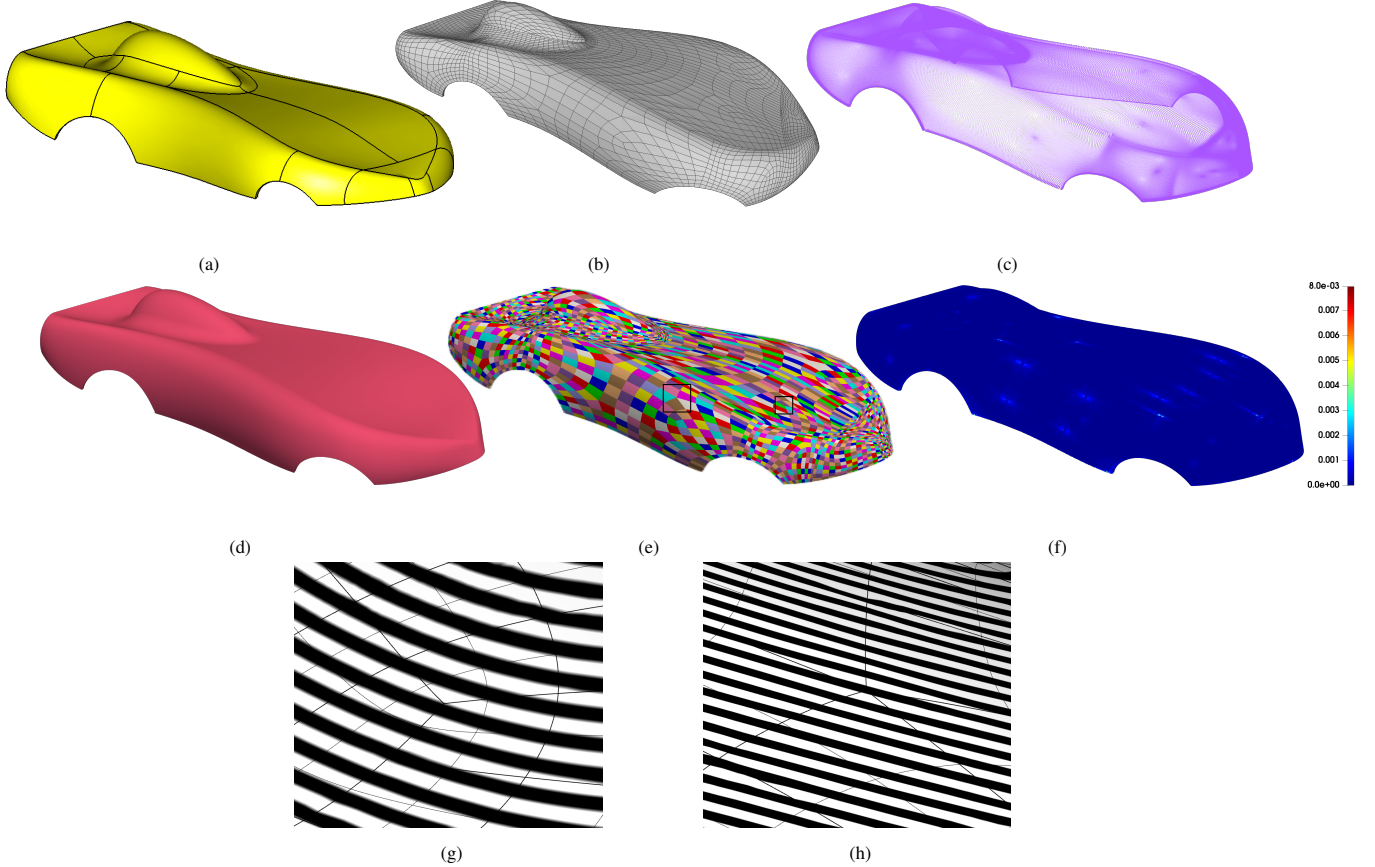


Figure 15: Car model. (a): original MCAD model. (b): quad mesh extrapolated from the MCAD geometry. (c): point cloud sampling of the original MCAD model. (d): surface in solid color. (e): surface in multipatch color. (f): error color plot representing the ℓ_2 distance between the point cloud and the resulting surface. (g)-(h): reflection lines around two EVs of different valences pointed out in (e).

Figures 15 to 18 show for each of the four models, in order: the point cloud acquired from the initial geometries; a comparison between the input data and the fitted surface; the fitted surface in solid coloring; the fitted surface in multipatch coloring; and a color plot representing the approximation error in Euclidean norm. Moreover, Table 1 summarizes the dimensions of the starting point cloud, control cage and spline space together with the approximation errors evaluated with the formulae in (10). Such errors, for the Car, KCS and NASA CRM models, have been obtained without forcing any smoothing (i.e., $\lambda = 0$). For the Dream Chaser model a parameter $\lambda = 0.1$ has been used. From the L^∞ errors represented in Table 1, as well as in the color map of the models' color plot, it can be noticed that the highest errors are located, as expected, around the EVs and near the sharp regions of the MCAD models. This is because we are fitting sharp edges with high smoothness bases which cannot properly recreate the actual shape of the model in these regions. In order to increase the quality of the fitting (i.e., decrease the error) and faithfully reproduce the characteristics of the input model, our construction allows us to identify the sharp edges of the MCAD model which are to be preserved. This is automated, owing to the fact that the point cloud sample grid (see Section 3) derives directly from the subdivision surface. Since the control cage maintains an exact topological relationship with the target B-rep, the location of the MCAD edges is known explicitly. Defining on those edges (via, for example, an input list) C^0 basis functions *only*, our output surface will manifest these sharp features. It is important to note that

despite the difference in the sizes of the four models, the relative error is always of the same order of magnitude.

Regarding IGA simulations, some experiments with the heat and biharmonic equations have been performed. With reference to (13) and (15), we run the simulations for each of the four MCAD models, considering as the final time the instances $T = 0.1$, $T = 0.2$, $T = 0.3$ and $T = 0.4$ minutes. These time dependent integrations have been carried out using 20 time steps t_{step} under the Crank-Nicolson method.

	Car	Dream Chaser	KCS	NASA CRM
n_b	166198	168187	166881	116607
n_F	10432	10576	10456	7336
n_p	1262272	1279696	1265179	887656
L^∞ error	7.965e-03	4.610e-02	4.644e-01	2.946e-03
Relative error	2.7e-03	5.1e-03	2.0e-03	1.7e-03
RMSE error	1.292e-04	1.660e-03	5.890e-03	8.264e-05
Maximal length	3	9	230	1.70

Table 1: Fitting errors, spline space and MCAD model features for the experiments in Figures 15-16-17-18.

	Car	Dream Chaser	KCS	NASA CRM
Cage generation (C)	1 min 45 sec	1 min 4 sec	55.25 sec	57.88 sec
MCAD sampling (C)	22 min 28 sec	26 min 24 sec	22 min 23 sec	22 min 53 sec
Bases computation (J)	11.39 sec	10.98 sec	10.04 sec	6.03 sec
Fitting (G)	2 min 43 sec	3 min 11 sec	2 min 51 sec	1 min 42 sec
Heat simulation (G)	-	1.43 sec/ t_{step}	-	0.9 sec/ t_{step}
Biharmonic simulation (G)	1.34 sec/ t_{step}	-	1.47 sec/ t_{step}	-

Table 2: Comprehensive elapsed time for each step of the procedure.

Figure 19 to Figure 22 present the results of the IGA simulations for both the heat (12) and biharmonic (14) equations. These are obtained by setting, as initial conditions, a heat source at the likely location of the engines within the various vehicles represented. This demonstrates a realistic analysis of the thermal behavior of such models. Moreover, Table 2 presents the running times required to compute each fragment of the pipeline.

7. Conclusion

This work serves as a proof of concept for unifying isogeometric analysis and subdivision surfaces as enabling technologies for simulation. We presented a complete and efficient pipeline to convert MCAD models into G^1 smooth objects which are suitable for isogeometric analysis simulations. Starting from an MCAD object, we first produce a quad mesh whose Catmull-Clark limit surface adequately approximates the input geometry. Guided by this limit surface we compute a point cloud sampling of the MCAD model, which is fitted with the use of basis functions defined over the previously extracted quad mesh, in a least squares approach. The obtained spline surface and the G^1 basis are used to run IGA simulations. To demonstrate the quality of the fitting, various numerical experiments derived from real MCAD models are provided with their error measurements. We illustrate the IGA simulations for the heat and biharmonic equations in real-life situations, highlighting the suitability of our approach for analysis.

Our process features the translation of traditional MCAD B-rep to a format more suitable for CAE applications. The technique aims to reproduce the originating B-rep, maintaining all prior topological relationships. This exposes the sensitivity of our method to both the complexity and the quality of the input geometry (in particular, the presence of sliver faces). As an area of future study, we highlight the potential of employing *virtual topology* [63] for mitigating the impacts of these artefacts by suppressing unnecessary topological features.

The exploitation of alternative geometric representations underpins the notion of a *hybrid modeling* approach. We anticipate that these alternative representations will become increasingly integrated into the design, analysis, and optimization stages of product development. Consequently, this will simplify the workflow, boost performance and productivity, and reduce time to market. However, a realistic expectation within industry would be to produce an MCAD B-rep representation as the end result, as typically mandated by many manufacturing processes. Therefore, the original MCAD model must be updated to reflect the results of the design optimization process. Our G^1 quad mesh spline representation maintains an *exact* topological associativity with the original B-rep, therefore lending itself to

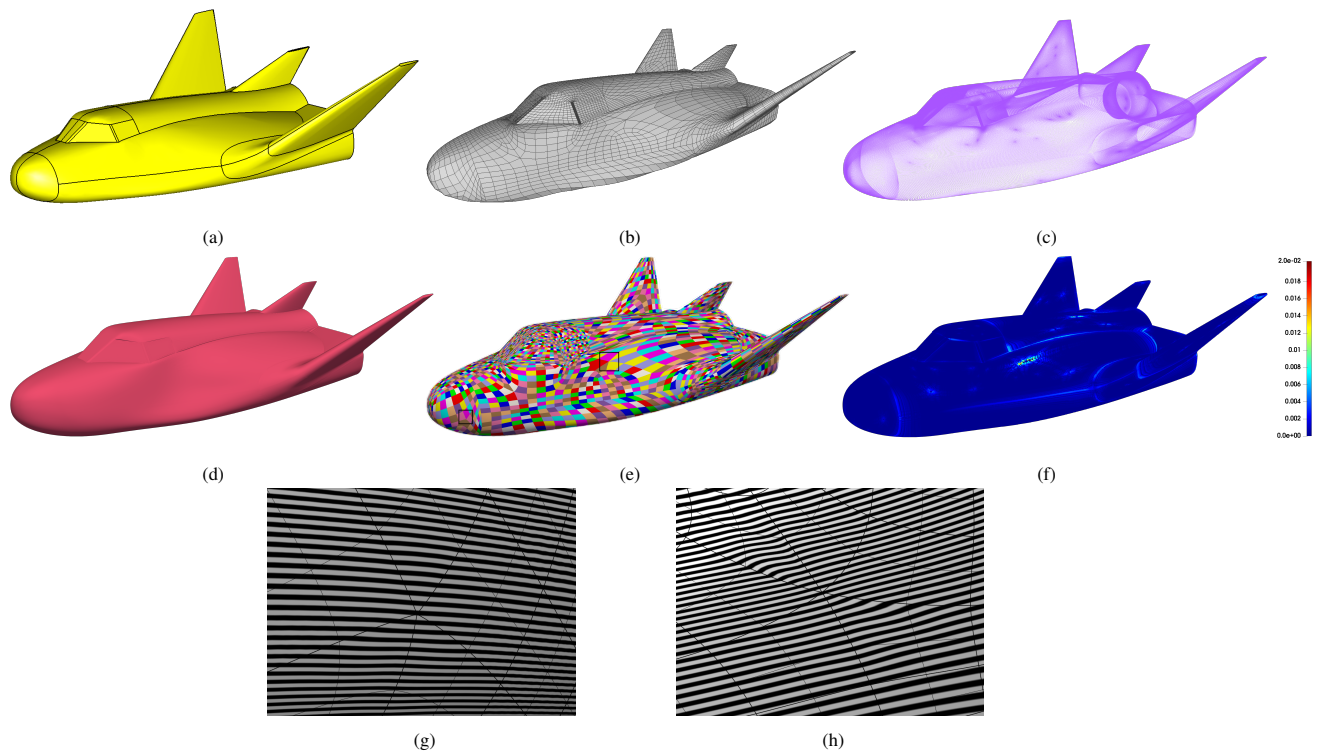


Figure 16: Dream Chaser shuttle model. (a): input MCAD model. (b): quad mesh extrapolated from the MCAD geometry. (c): point cloud sampling of the original MCAD model. (d): surface in solid color. (e): surface in multipatch color. (f): error color plot representing the ℓ_2 distance between the point cloud and the resulting surface. (g)-(h): reflection lines around two EVs of different valences pointed out in (e).

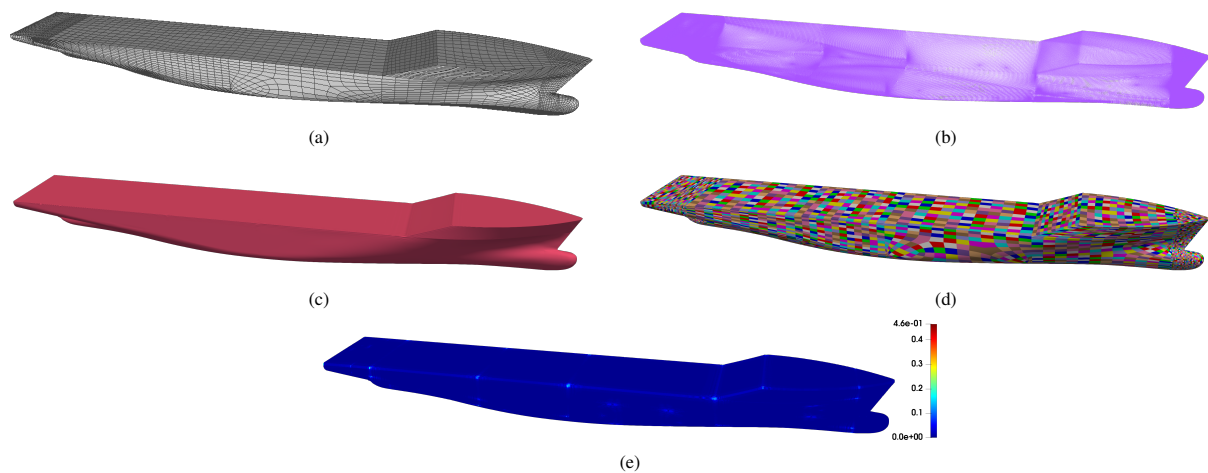


Figure 17: KCS hull model. (a): quad mesh extrapolated from the MCAD geometry. (b): point cloud sampling of the original MCAD model. (c): surface in solid color. (d): surface in multipatch color. (e): error color plot representing the ℓ_2 distance between the point cloud and the resulting surface.

a convenient route back to a traditional MCAD B-rep representation *without altering the topology*. We propose this as an important topic for further study, as it would complete the tool set for employing a hybrid modeling approach within an industrial setting.

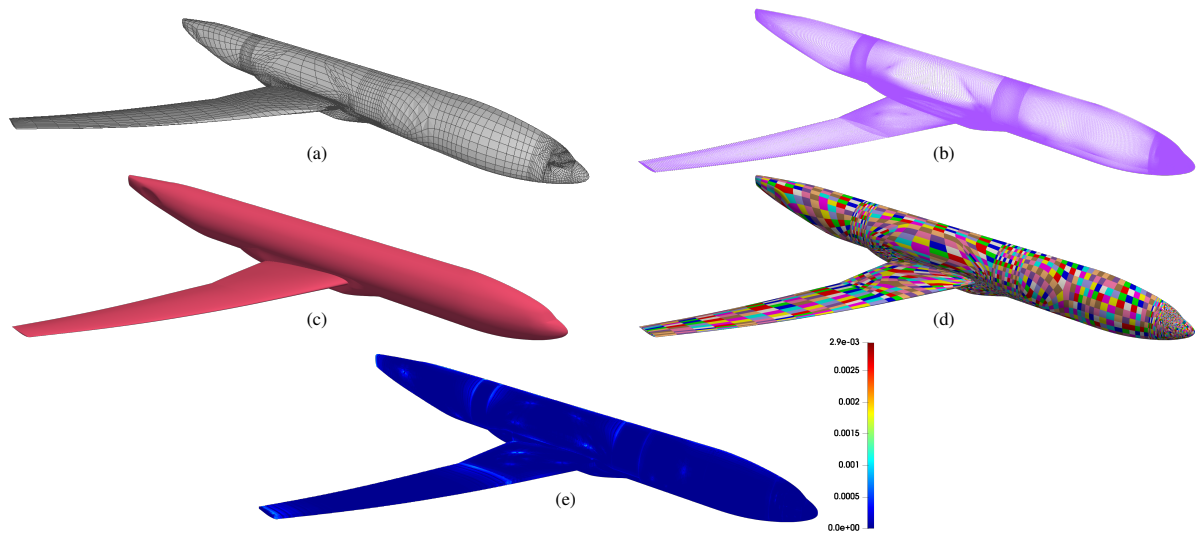


Figure 18: NASA CRM. (a): quad mesh extrapolated from the MCAD geometry. (b): point cloud sampling of the original MCAD model. (c): surface in solid color. (d): surface in multipatch color. (e): error color plot representing the ℓ_2 distance between the point cloud and the resulting surface.

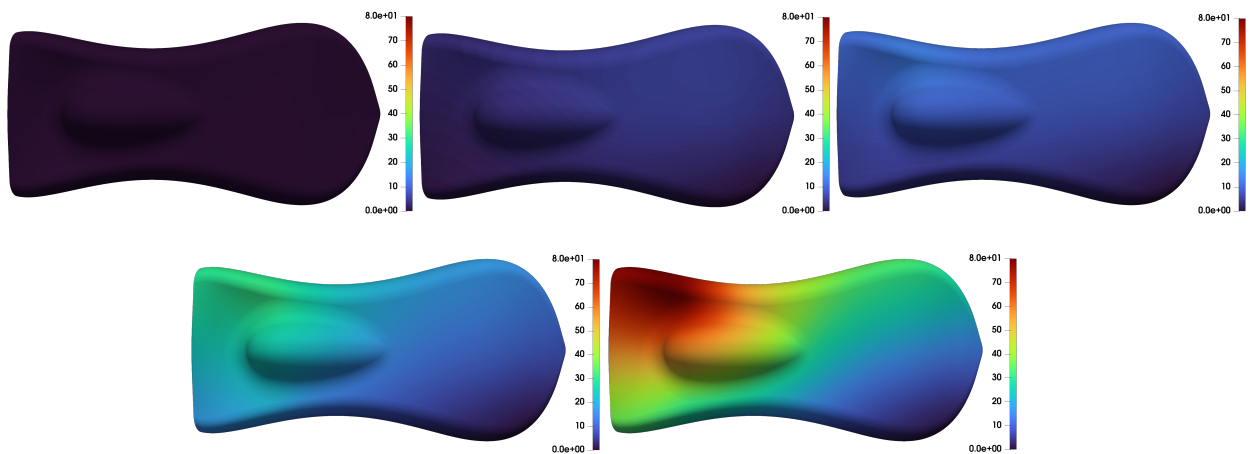


Figure 19: Solution for the biharmonic equation on the Car model at the instants from $T = 0, 0.1, 0.2, 0.3$ and $T = 0.4$ minutes.

8. Acknowledgments

This work has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 860843, and partially from the ATI/IUK project COLIBRI grant agreement No. 46349.

References

- [1] P. Boggs, A. Althsuler, A. Larzelere, E. Walsh, R. Clay, M. Hardwick, DART system analysis., Tech. Rep. SAND2005-4647 (OSTI) (2005).
- [2] J. Cottrell, T. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & sons, Ltd., 2009.
- [3] F. Bianconi, P. Conti, L. Di Angelo, Interoperability among CAD/CAM/CAE systems: A review of current research trends, in: Geometric Modeling and Imaging–New Trends (GMAI’06), 2006, pp. 82–89.
- [4] D. Zorin, P. Schröder, A. Derose, L. Kobbelt, A. Levin, W. Sweldens, Subdivision for modeling and animation, 2000, course Notes of SIGGRAPH.
- [5] J. Peters, U. Reif, Subdivision surfaces, Springer, 2008.

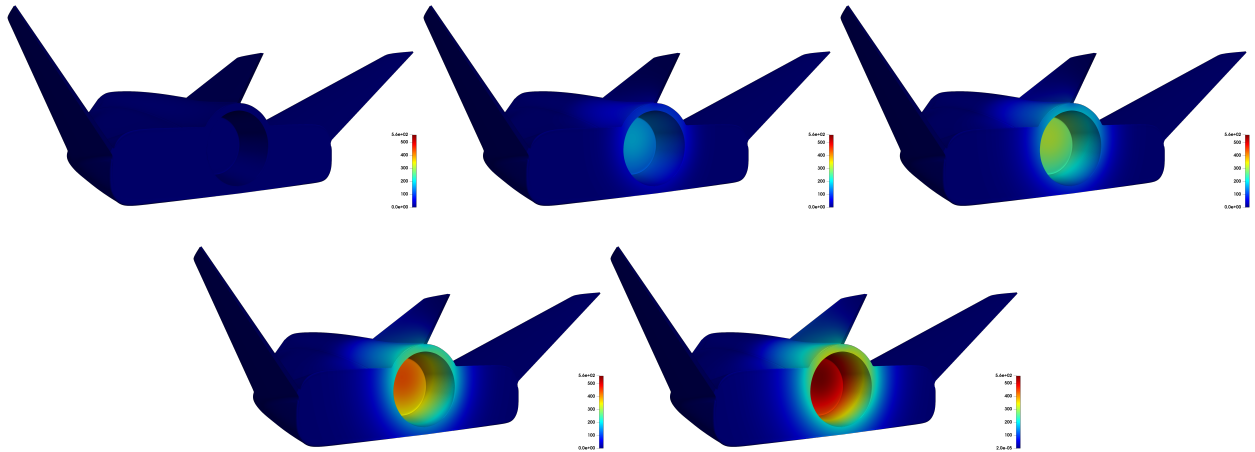


Figure 20: Solution for the heat equation on the Dream Chaser model at the instants from $T = 0$ to $T = 0.4$ min.

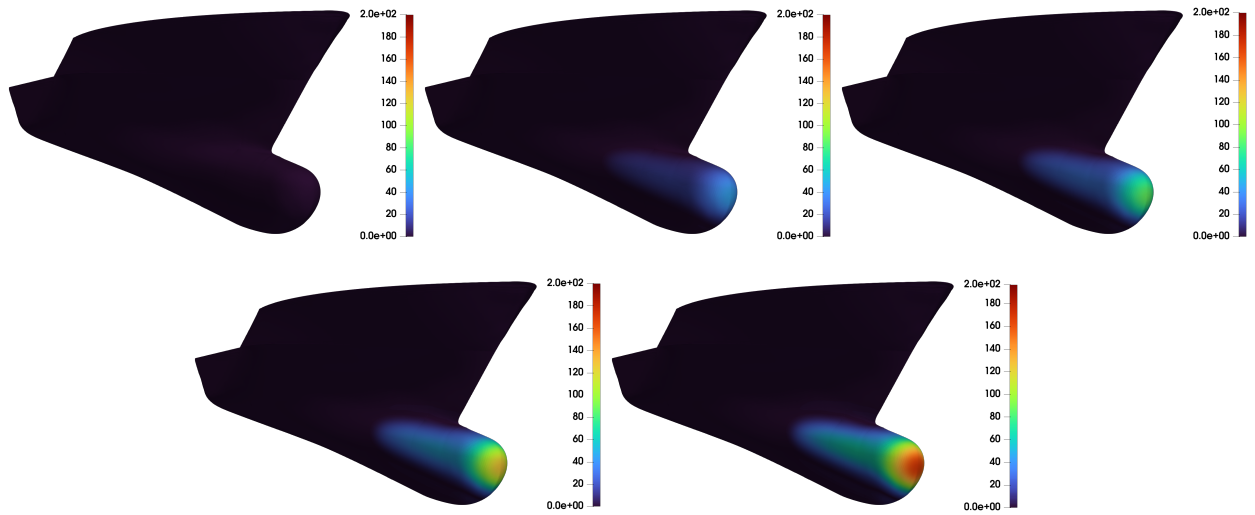


Figure 21: Solution for the biharmonic equation on the KCS ship model at the instants from $T = 0$ to $T = 0.4$ min.

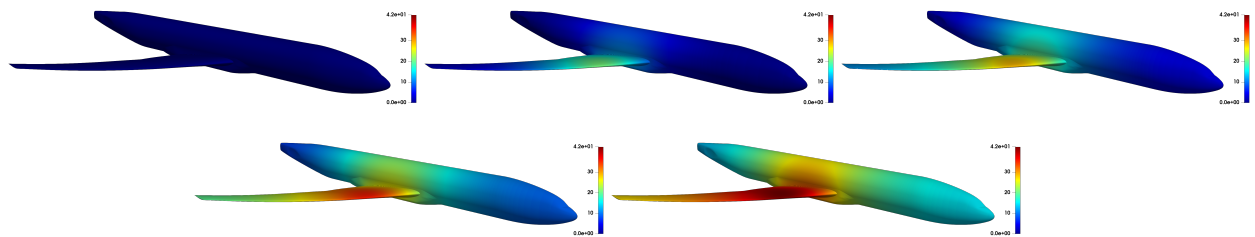


Figure 22: Solution for the heat equation on the NASA CRM at the instants from $T = 0$ to $T = 0.4$ min.

- [6] K. Bandara, T. Rüberg, F. Cirak, Shape optimisation with multiresolution subdivision surfaces and immersed finite elements, *Computer Methods in Applied Mechanics and Engineering* 300 (2016) 510–539.
- [7] E. Catmull, J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design* 10 (6) (1978).
- [8] A. Nasri, M. Sabin, R. A. Zaki, N. Nassiri, R. Santina, Feature curves with cross curvature control on Catmull-Clark subdivision surfaces, in: T. Nishita, Q. Peng, H.-P. Seidel (Eds.), *Advances in Computer Graphics*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 761–768.
- [9] J. Shen, J. Kosinka, M. Sabin, N. Dodgson, Conversion of trimmed NURBS surfaces to Catmull-Clark subdivision surfaces, *Computer Aided Geometric Design* 31 (7-8) (2014) 486–498.
- [10] P. Frey, P. George, *Mesh Generation*, Wiley, 2008.
- [11] Z. Ali, J. Tyacke, P. Tucker, S. Shahpar, Block topology generation for structured multi-block meshing with hierarchical geometry handling, *Procedia Engineering* 26 (2016) 212–224.
- [12] C. Geuzaine, J. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, *International Journal for Numerical Methods in Engineering* 79 (11) (2009) 1309–1331.
- [13] W. Wang, Y. Zhang, M. A. Scott, T. J. R. Hughes, Converting an unstructured quadrilateral mesh to a standard T-spline surface, *Computational Mechanics* 48 (4) (2011) 477–498.
- [14] Y. Lai, L. Liu, Y. J. Zhang, J. Chen, E. Fang, J. Lua, Rhino 3D to Abaqus: A T-Spline Based Isogeometric Analysis Software Framework, Springer International Publishing, 2016, p. 271–281.
- [15] H. Casquero, X. Wei, D. Toshniwal, A. Li, T. J. Hughes, J. Kiendl, Y. J. Zhang, Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines, *Computer Methods in Applied Mechanics and Engineering* 360 (2020) 112765.
- [16] R. R. Hiemstra, K. M. Shepherd, M. J. Johnson, L. Quan, T. J. Hughes, Towards untrimmed nurbs: CAD embedded reparameterization of trimmed b-rep geometry using frame-field guided global parameterization, *Computer Methods in Applied Mechanics and Engineering* 369 (2020) 113227.
- [17] K. M. Shepherd, X. D. Gu, T. J. Hughes, Feature-aware reconstruction of trimmed splines using ricci flow with metric optimization, *Computer Methods in Applied Mechanics and Engineering* 402 (2022) 115555.
- [18] K. M. Shepherd, X. D. Gu, T. J. Hughes, Isogeometric model reconstruction of open shells via ricci flow and quadrilateral layout-inducing energies, *Engineering Structures* 252 (2022) 113602.
- [19] B. Urlick, 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, *Computer-Aided Design* 115 (2019) 147–160.
- [20] E. Wobbes, Y. Bazilevs, T. Kuraishi, Y. Otoguro, K. Takizawa, T. E. Tezduyar, *Advanced IGA Mesh Generation and Application to Structural Vibrations*, Springer International Publishing, 2023, p. 513–531.
- [21] D. Bommers, T. Lempfer, L. Kobbelt, Global structure optimization of quadrilateral meshes, *Computer Graphics Forum* 30 (2) (2011) 375–384.
- [22] M. Tarini, E. Puppo, D. Panozzo, N. Pietroni, P. Cignoni, Simple quad domains for field aligned mesh parametrization, *ACM Transactions on Graphics* 30 (6) (2011) 1–12.
- [23] M. Couplet, M. Reberol, J. Remacle, Generation of high-order coarse quad meshes on CAD models via integer linear programming, in: *AIAA AVIATION 2021 FORUM*, American Institute of Aeronautics and Astronautics, 2021.
- [24] Y. Yu, X. Wei, A. Li, J. G. Liu, J. He, Y. J. Zhang, HexGen and Hex2Spline: Polycube-based hexahedral mesh generation and spline modeling for Isogeometric Analysis applications in LS-DYNA, in: C. Manni, H. Speleers (Eds.), *Geometric Challenges in Isogeometric Analysis*, Springer International Publishing, Cham, 2022, pp. 333–363.
- [25] X. Wei, Y. J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J. A. Evans, T. J. Hughes, Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, *Computer Methods in Applied Mechanics and Engineering* 341 (2018) 609–639.
- [26] J. Xie, J. Xu, Z. Dong, G. Xu, C. Deng, B. Mourrain, Y. J. Zhang, Interpolatory Catmull-Clark volumetric subdivision over unstructured hexahedral meshes for modeling and simulation applications, *Computer Aided Geometric Design* 80 (2020) 101867.
- [27] T. Sederberg, J. Zheng, D. Sewell, M. Sabin, Non-uniform recursive subdivision surfaces, in: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, ACM Press, 1998.
- [28] T. Cashman, U. Augsdörfer, N. Dodgson, M. Sabin, NURBS with extraordinary points, *ACM Transactions on Graphics* 28 (3) (2009) 1–9.
- [29] T. Cashman, N. Dodgson, M. Sabin, A symmetric, non-uniform, refine and smooth subdivision algorithm for general degree B-splines, *Computer Aided Geometric Design* 26 (1) (2009) 94–104.
- [30] B. Marussig, T. J. R. Hughes, A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects, *Archives of Computational Methods in Engineering* 25 (4) (2017) 1059–1127.
- [31] Y. Ma, W. Ma, A subdivision scheme for unstructured quadrilateral meshes with improved convergence rate for isogeometric analysis, *Graphical Models* 106 (2019) 101043.
- [32] C. Loop, S. Schaefer, Approximating Catmull-Clark subdivision surfaces with bicubic patches, *ACM Transactions on Graphics* 27 (1) (2008) 8:1–8:11.
- [33] J. Peters, Patching Catmull-Clark meshes, in: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000, pp. 255–258.
- [34] J. Fan, J. Peters, On smooth bicubic surfaces from quad meshes, in: *International Symposium on Visual Computing*, Springer, 2008, pp. 87–96.
- [35] J. Peters, J. Fan, On the complexity of smooth spline surfaces from quad meshes, *Computer Aided Geometric Design* 27 (1) (2010) 96–105.
- [36] S. Hahmann, G. Bonneau, B. Caramiaux, Bicubic G^1 interpolation of irregular quad meshes using a 4-split, in: *International Conference on Geometric Modeling and Processing*, Springer, 2008, pp. 17–32.
- [37] G. Bonneau, S. Hahmann, Flexible G^1 interpolation of quad meshes, *Graphical Models* 76 (6) (2014) 669–681.
- [38] K. Karčiauskas, J. Peters, Improved shape for refinable surfaces with singularly parameterized irregularities, *Computer-Aided Design* 90 (2017) 191–198.
- [39] M. Marsala, A. Mantzaflaris, B. Mourrain, G^1 – smooth biquintic approximation of Catmull-Clark subdivision surfaces, *Computer Aided*

- Geometric Design 99 (2022) 102158.
- [40] P. Alfeld, L. Schumaker, The dimension of bivariate spline spaces of smoothness r for degree $d \geq 4r + 1$, *Constr. Approx.* 3 (2) (1987) 189–197.
 - [41] D. Hong, Spaces of bivariate spline functions over triangulation, *Approx. Theory Appl.* 7 (1) (1991) 56–75.
 - [42] L. Schumaker, Bounds on the dimension of spaces of multivariate piecewise polynomials, *Rocky Mountain J. Math.* 14 (1) (1984) 251–264.
 - [43] M. Lai, L. Schumaker, Spline functions on triangulations, Vol. 110 of *Encyclopedia of Mathematics and its Applications*, Cambridge University Press, Cambridge, 2007.
 - [44] B. Mourrain, N. Villamizar, Homological techniques for the analysis of the dimension of triangular spline spaces, *J. Symbolic Comput.* 50 (2013) 564–577.
 - [45] B. Mourrain, R. Vidunas, N. Villamizar, Geometrically continuous splines for surfaces of arbitrary topology, *Computer Aided Geometric Design* 45 (2016) 108–133.
 - [46] M. Kapl, G. Sangalli, T. Takacs, Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations, *Computer Aided Geometric Design* 52-53 (2017) 75–89.
 - [47] M. Kapl, G. Sangalli, T. Takacs, Isogeometric analysis with C^1 functions on planar, unstructured quadrilateral meshes, *The SMAI Journal of computational mathematics* S5 (2019) 67–86.
 - [48] M. Kapl, G. Sangalli, T. Takacs, An isogeometric C^1 subspace on unstructured multi-patch planar domains, *Computer Aided Geometric Design* 69 (2019) 55–75.
 - [49] A. Blidia, B. Mourrain, N. Villamizar, G^1 -smooth splines on quad meshes with 4-split macro-patch elements, *Computer Aided Geometric Design* 52-53 (2017) 106–125.
 - [50] A. Blidia, B. Mourrain, G. Xu, Geometrically smooth spline bases for data fitting and simulation, *Comput. Aided Geom. Design* 78 (2020) 101814, 15.
 - [51] M. Marsala, A. Mantzaflaris, B. Mourrain, G^1 spline functions for point cloud fitting, *Applied Mathematics and Computation* 460 (2024) 128279.
 - [52] D. J. Sheehy, C. G. Armstrong, D. J. Robinson, Computing the medial surface of a solid from a domain delaunay triangulation, in: *Proceedings of the Third ACM Symposium on Solid Modeling and Applications, SMA '95*, Association for Computing Machinery, New York, NY, USA, 1995, p. 201–212.
 - [53] W. Gordon, C. Hall, Construction of curvilinear coordinate systems and application to mesh generation, *International J. Num. Methods in Eng.* 7 (1973) 461–477.
 - [54] G. Farin, J. Hoschek, M.-S. Kim, *Handbook to Computed Aided Geometric Design*, North Holland, 2002.
 - [55] J. Stam, Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, in: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, ACM Press, 1998.
 - [56] M. Bercovier, T. Matskewich, *Smooth Bézier Surfaces over Unstructured Quadrilateral Meshes*, Lecture Notes of the Unione Matematica Italiana, Springer International Publishing, 2017.
 - [57] L. Dedè, A. Quarteroni, Isogeometric analysis for second order partial differential equations on surfaces, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 807–834, Isogeometric Analysis Special Issue.
 - [58] P. G. Ciarlet, *The finite element method for elliptic problems*, SIAM, 2002.
 - [59] M. Kapl, V. Vitrih, B. Jüttler, K. Birner, Isogeometric analysis with geometrically continuous functions on two-patch geometries, *Computers & Mathematics with Applications* 70 (7) (2015) 1518–1538.
 - [60] M. Marsala, Modeling, approximation and simulation using smooth splines on unstructured meshes, PhD thesis, Université Côte d'Azur (2023).
 - [61] CADfix, <https://www.iti-global.com/cadfix>.
 - [62] A. Mantzaflaris, An overview of geometry plus simulation modules, in: *Mathematical Aspects of Computer and Information Sciences*, Springer International Publishing, Cham, 2020, pp. 453–456.
 - [63] A. Sheffer, M. Bercovier, T. Blacker, J. Clements, Virtual topology operators for meshing, *International Journal of Computational Geometry & Applications* 10 (03) (2000) 309–331.