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

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

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

Michelangelo Marsala[a,*], Angelos Mantzaflaris[a], Bernard Mourrain[a], Sam Whyman[b], Mark Gammon[b]

[a] *Inria Centre at Université Côte d'Azur, Sophia Antipolis, France*
[b] *International TechneGroup Ltd, Cambridge, UK*

Abstract

In this work 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. Then we 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, except at the sharp edges. 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 modelling 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 reparation, meshing, numerical simulations, and optimisation. Currently, they require specific engineering efforts, are time consuming, and prone to errors and approximations [5], [11][Chap 1]. This explains why alternative approaches are under investigation.

Boundary representation (B-rep) is used widely in solid modelling applications due to its flexibility and precision when representing manufacturable geometry in mechanical CAD (MCAD) processes. Whilst B-rep prescribes definitive topological relationships between geometric entities, there is no guarantee of geometric fidelity. For example, the embedding NURBS surfaces of topologically neighbouring faces may be C^0 *discontinuous* to an arbitrary extent at their interface. This is referred to as *geometric sloppiness*, 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. Fig. 1 shows an example of a typical MCAD B-rep model, with separate NURBS surfaces as the geometry for each face.

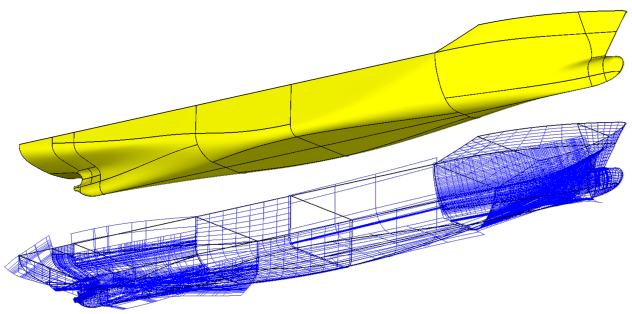

Figure 1: Top: Shaded MCAD B-rep faces of the KCS ship hull model, with edges shown in black wireframe. Bottom: Embedding NURBS patches for each face.

The combination of geometric sloppiness and complex topology can present serious challenges for CAE applications and are often a major bottleneck to an effective design pipeline. 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 modelling*, and allows the benefits of multiple different representa-

*Corresponding author

tions to be exploited. In an effective hybrid modelling 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 [39, 33] 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 optimisation processes, wherein subdivisions surfaces may act as a proxy to the MCAD geometry for convenient manipulation. Within a hybrid modelling 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.

One of the limitations of the widely used Catmull-Clark scheme [10] is that the limit surface has zero cross-curvature at the boundary. In the fields of computer graphics and animation, this is rarely an obstacle to achieving results. Many implementations, such as the `OpenSubdiv` library from Pixar, allow the application of *sharp* or *semi-sharp* creases to either 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. For engineering purposes however, this does not permit accurate modelling of real geometries which exhibit cross-curvature at the boundaries. 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 [36].

To analyse the mechanical behaviour 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 [15]. In recent years an alternative approach called isogeometric analysis (IGA) has been proposed to circumvent these difficulties [11]. Rather than involving an expensive meshing approximation step, it directly exploits the piecewise B-spline parametrisation 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 parametrisation. Furthermore, most applications of these techniques deal with spline functions associated with planar domain partitions.

1.1. Contributions

We present a new scheme for handling MCAD B-rep models, and for analysing their mechanical and physical behaviour. 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 partition 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.

We emphasise that our process *automatically* generates a G^1 smooth Bézier surface that represents a target MCAD B-rep model, while respecting the curvature of the faces.

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 [2] 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.

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 [16]. The research centred around meshing geometric models is still very active, for instance with the generation of quad-meshes suitable for accurate numerical simulations [6, 38, 12]. 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 optimisation 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 [39] 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, [35, 8, 9] or trimmed B-spline representation to subdivision surfaces [36]. 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 [26].

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 [25]. The resulting surface is not necessarily smooth. Several works have addressed the construction of smooth surfaces from (quad) meshes. See e.g., [31, 14, 32, 18, 7, 23, 27].

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., [1, 19, 34, 24, 30] and references therein). The analysis of G^1 spline spaces is much less investigated [29, 20, 21, 22, 3, 4, 28]. 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 (Fig. 2). These can equally be thought of as tangent vectors stored at the ends of the spline.

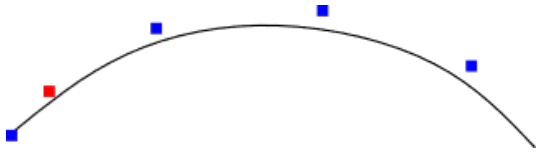

Figure 2: Schematic of a cubic B-spline with knot vector [0,0,0,0,1,2,3,4,...]. The slope control point is shown in red, and all other control points in blue. The location of the red point influences the shape of the spline only within the first two knot spans.

The limit surface is then defined to be the tensor product surface of the boundary B-splines. Since the slope control points effect only the first two knot spans, it follows that the first two layers of patches depart from the usual behaviour of the regular regions of a Catmull-Clark subdivision surface (i.e., away from the EVs). This modification constitues the addition of Bézier edge conditions to the standard Catmull-Clark scheme [36], and is illustrated by Fig. 3. The 3D location of the limit surface at the boundary is influenced *only* by the control points on the boundary. Therefore, neighbouring 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.

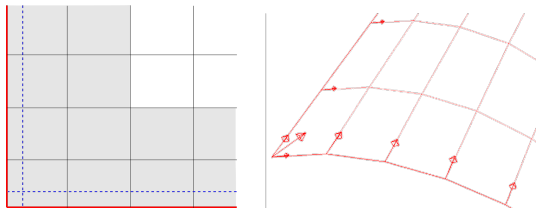

Figure 3: Left: Schematic showing the application of Bézier edge conditions to a subdivision surface. The red lines represent the rows of control points which approximate the MCAD edges as cubic B-splines. The blue dashed lines represent the rows of slope control points. Shaded patches are defined using the tensor product of the boundary B-splines, and the unshaded patches are the usual regular bicubic B-spline patches. Right: 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.

The use of Bézier edge conditions imposes strict topological requirements upon the control cage. Namely, no EV may be placed on the boundary, 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

Fig. 3). The appearance of any EV 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 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 Fig. 4.

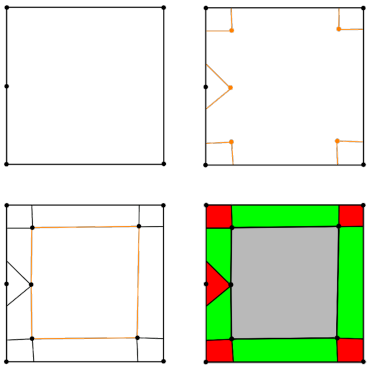

Figure 4: Schematic outlining the SubD layering process. Top left: MCAD B-rep face with 5 vertices. Top right: corner blocks are constructed around the MCAD vertices. Bottom left: corners blocks are connected to form edge blocks. Bottom right: Domain is partitioned into corner (red) and edge (green) blocks which can receive structured mesh, and one interior (grey) region which receives unstructured mesh.

The size and positions of the corner blocks are determined using the two-dimensional medial axis [2] as a guide to ensure that the blocks do not intersect. An example of this is given in Fig. 5.

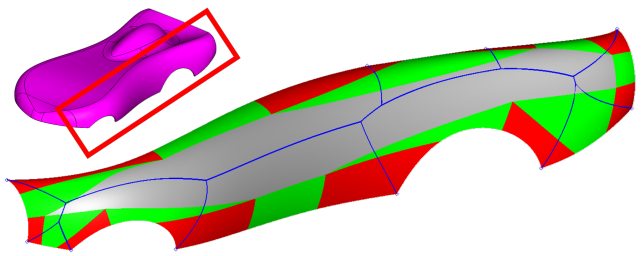

Figure 5: SubD layering for the wheel arch of the Car model, showing corner (red), edge (green) and unstructured regions (grey). The 2D medial axis (blue) determines the shape of the blocks so that they do not intersect.

Once the domain partitioning has been completed, we generate a coarse mesh of the layer faces. The boundary layer is meshed using a transfinite interpolation technique [17] and is specified to be one element thick. The interior region is meshed using quad-dominant meshing technology [2]. The coarse mesh is then subdivided twice, and 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. This refinement of a coarse mesh is illustrated in Fig. 6.

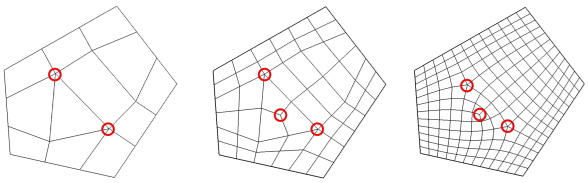

Figure 6: Left: Initial coarse mesh for a 5-sided face. Centre: One level of refinement. Right: Two levels of refinement (with mesh smoothing applied). The EVs are highlighted in red. After two levels of refinement the location of the EVs meet the topological requirements for the Bézier edge conditions.

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 NURBS patches. An example of a watertight limit surface is given in Fig. 7.

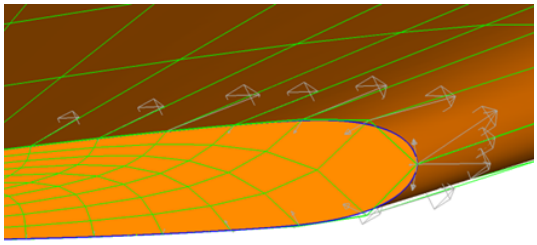

Figure 7: A single subdivision surface representing the wing tip of the SNC Dream Chaser model, which is comprised of multiple MCAD B-rep faces. The control cage is shown in green, and the limit surface is shaded in orange. Bézier edge conditions (with slope control vectors shown in grey) are applied 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).

An example of the control cage computation is presented in Fig. 8. This may be contrasted with the multiple NURBS patch representation shown in Fig. 1.

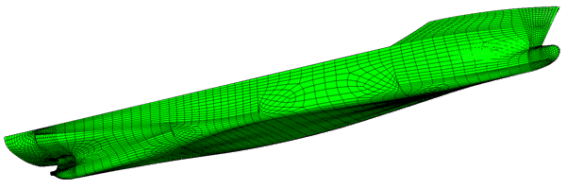

Figure 8: Subdivision surface approximation of the KCS ship hull. The control cage is shown in wireframe, and the limit surface shaded.

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 behaviour 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. For the interior, we found that 1000 rounds of iterative control cage adjustment give a sufficiently good approximation for each of the four models. This is demonstrated in Fig. 9, which shows the front section of the NASA CRM.

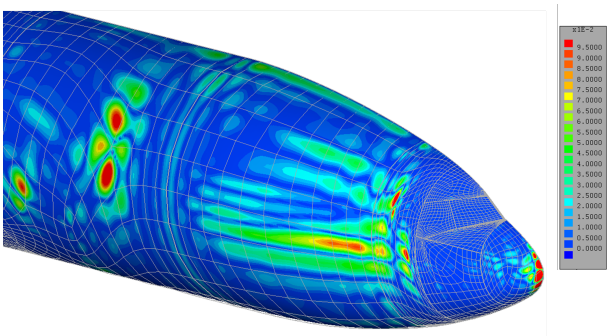

Figure 9: Heat map of the distance from the limit surface to the target MCAD geometry, for the forward section of the NASA CRM. A maximum fitting error of approximately 0.005% of the aircraft length is achieved after 1000 rounds of iterative control cage adjustment.

The fitting procedure 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 [37]. The corresponding 3D positions on the limit surface (which *approximates* the geometry) are then moved *exactly* 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 measures aimed at preventing creases and folds appearing in highly curved regions. The result is a uniform non-folding structured grid of sample points lying on the target geometry, for each patch of the limit surface. Fig. 10 shows an example of the typical distribution of the point cloud sampling.

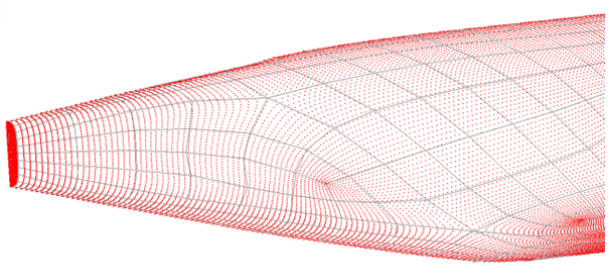

Figure 10: An example of the distribution of points in the sampling of the MCAD geometry, for the tail of the NASA CRM. Each patch of the subdivision surface receives a uniform grid of samples, which are projected onto the embedding surfaces of the MCAD B-rep faces.

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

Let $f = (f_\sigma)_{\sigma\in\mathcal{M}}$ be a collection of functions defined over the faces σ of a quad mesh $\mathcal{M}$. We name e the

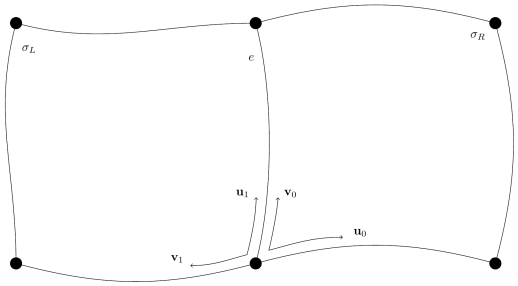

Figure 11: Local coordinate systems between two adjacent patches.

common edge shared by two adjacent patches $f^L = f\big|_{\sigma_L}$ and $f^R = f\big|_{\sigma_R}$ of a collection f. With reference to the local systems' orientation in Fig. 11, two functions are

said to be G^1 (or tangent plane continuous) if they join G^0 (or C^0), i.e.,

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

and there exists three function $a(u)$, $b(u)$, $c(u)$ called *gluing functions* such that

$$f^L_{v_1}(u_1,0)c(u_1) + f^R_{u_1}(0,u_1)b(u_1) + f^L_{v_0}(0,u_1)a(u_1) = 0\,, \tag{2}$$
$$c(u_1)b(u_1) < 0\,, \tag{3}$$
$$\langle f^L_{v_1}(u_1,0), f^R_{v_0}(0,u_1)\rangle \neq 0\,, \tag{4}$$

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. A particular spline space defined over quad meshes and verifying eqs. (1) to (4) is the G^1ACC derived in [27]; it is obtained from a collection of biquintic Bézier functions by imposing G^1 continuity across extraordinary regions making use of the gluing functions

$$\begin{aligned} a(u) &= a_0 B^0_2(u) - a_2 B^2_2(u)\,,\\ b(u) &= -1\,,\\ c(u) &= 1\,, \end{aligned}$$

with $B^k_d(u) = \binom{d}{k}u^k(1-u)^{d-k}$ the univariate Bernstein polynomial 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). The relations for the tangent plane continuity constraints between the control points $\mathbf{b}_{i,j}, i,j = 0,\dots,5$ of two neighbouring Bézier patches f^L, f^R defining the G^1ACC space are the following:

$$\begin{aligned}
\mathrm{b}^L_{0,1} + \mathrm{b}^R_{1,0} &= \bar{a}_0\mathrm{b}^L_{0,0} + a_0\mathrm{b}^L_{1,0}\,,\\
5(\mathrm{b}^L_{1,1} + \mathrm{b}^R_{1,1}) &= a_0\mathrm{b}^L_{0,0} + 5\bar{a}_0\mathrm{b}^L_{1,0} + 4a_0\mathrm{b}^L_{2,0}\,,\\
10(\mathrm{b}^L_{2,1} + \mathrm{b}^R_{1,2}) &= -a_0\mathrm{b}^L_{0,0} + 5a_0\mathrm{b}^L_{1,0} + 10\bar{a}_0\mathrm{b}^L_{2,0} + 6a_0\mathrm{b}^L_{3,0}\,,\\
10(\mathrm{b}^L_{3,1} + \mathrm{b}^R_{1,3}) &= a_0\mathrm{b}^L_{0,0} - 5a_0\mathrm{b}^L_{1,0} + 10a_0\mathrm{b}^L_{2,0} + 10\bar{a}_0\mathrm{b}^L_{3,0}\\
&\quad + 4a_0\mathrm{b}^L_{4,0}\,,\\
\mathrm{b}^L_{4,1} + \mathrm{b}^R_{1,4} &= 2\mathrm{b}^L_{4,0}\,,\\
\mathrm{b}^L_{5,1} + \mathrm{b}^R_{1,5} &= 2\mathrm{b}^L_{5,0}\,,\\
10(\mathrm{b}^L_{3,0} - \mathrm{b}^L_{2,0}) &= \mathrm{b}^L_{0,0} - 5\mathrm{b}^L_{1,0} + 5\mathrm{b}^L_{4,0} - \mathrm{b}^L_{5,0}\,,
\end{aligned}$$

with $\bar{a}_0 = 2 - a_0$.

4.2. Construction

The aim of this work 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 discretise 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 [28] 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^1ACC 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 will get 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^1ACC 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 [28] 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 points $P_i \in \mathbb{R}^3$, $i = 1,\dots,n_P$ with associated parameters $\boldsymbol{\xi}^i = (\xi^i_1, \xi^i_2) \in \mathbb{R}^2$ on the face σ_i, and a set of basis functions $B_j \in \mathcal{B}$. The idea of the least squares fitting technique is to find the coefficients $c_j \in \mathbb{R}^3$ defining, for instance, a spline surface $S(\boldsymbol{\xi};\sigma) = \sum_j c_j B_j(\boldsymbol{\xi};\sigma)$ for $\sigma \in \mathcal{M}$ such that the quantity

$$\sum_{i=1}^{n_P} \|S(\xi^i_1, \xi^i_2; \sigma_i) - P_i\|^2_2 \tag{5}$$

is minimal. In this work we consider as $\mathcal{B}$ the set of G^1 bases 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 point cloud $\mathcal{P}$ is the MCAD sampling outlined in section 3, whose cardinality will be indicated with n_P.

The next step of our pipeline is to compute the fitted surface S by solving the minimisation problem given in (5). 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:

$$L^\infty := \max_{i=1,\dots,nP} \|S(\boldsymbol{\xi}^i) - P_i\|_2\,, \tag{6}$$

$$\text{RMSE} := \sqrt{\frac{1}{n_P}\sum_{i=1}^{n_P} \|S(\boldsymbol{\xi}^i) - P_i\|_2^2}\,, \tag{7}$$

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 fitting 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 [11] (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 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

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

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 J, i.e.,

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

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

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

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

$$\widehat{g} : \widehat{\Omega} \longrightarrow \mathbb{R}\,, \quad \boldsymbol{\xi} \longmapsto \widehat{g}(\boldsymbol{\xi})\,, \quad \widehat{g}(\boldsymbol{\xi}) := \sqrt{\det\left(\widehat{G}(\boldsymbol{\xi})\right)}\,.$$

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

$$\nabla_\Omega \varphi(\mathbf{x}) := \left[\widehat{J}(\boldsymbol{\xi})\widehat{G}^{-1}(\boldsymbol{\xi})\widehat{\nabla}\widehat{\varphi}(\boldsymbol{\xi})\right] \circ \mathbf{x}^{-1}(\boldsymbol{\xi})\,, \tag{8}$$

with $\widehat{\nabla}$ representing the gradient operator in the parametric space.

For a more precise explanation of operators for isogeometric solutions to PDEs on manifolds, we refer the reader to [13].

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

$$\begin{cases} \dfrac{\partial u}{\partial t}(\mathbf{x},t) = c^2 \Delta_\Omega u(\mathbf{x},t)\,, & (\mathbf{x},t) \in \Omega \times (0,T]\,, \\ u(\mathbf{x},0) = u_0(\mathbf{x})\,, & \mathbf{x} \in \Omega\,, \\ u(\mathbf{x},t) = u_D(\mathbf{x},t)\,, & (\mathbf{x},t) \in \partial\Omega \times (0,T]\,, \end{cases} \tag{9}$$

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 (9) 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:

$$\widehat{a}(\widehat{v},\widehat{w}) := \int_{\widehat{\Omega}} \widehat{\nabla}\widehat{v}\cdot\left(\widehat{G}^{-1}\widehat{\nabla}\widehat{w}\right)\widehat{g}\,d\widehat{\Omega}\,, \quad \widehat{b}(\widehat{v},\widehat{w}) := \int_{\widehat{\Omega}} \widehat{v}\,\widehat{w}\,\widehat{g}\,d\widehat{\Omega}$$

Thus, the weak form of the heat equations can be formulated as: find $u \in H^1(\widehat{\Omega})$ s.t., for almost all $t \in (0,T)$,

$$\begin{cases} \widehat{a}(\widehat{u}(\mathbf{x},t),\widehat{w}) = \widehat{b}(\partial_t\widehat{u}(\mathbf{x},t),\widehat{w})\,, & (\mathbf{x},t) \in \Omega \times (0,T]\,, \\ u(\mathbf{x},0) = u_0(\mathbf{x})\,, & \mathbf{x} \in \Omega\,, \\ u(\mathbf{x},t) = u_D(\mathbf{x},t)\,, & (\mathbf{x},t) \in \partial\Omega \times (0,T]\,, \end{cases} \tag{10}$$

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

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 equation using geometrically smooth basis functions in the IGA environment.

The CAD models used for the numerical investigation are standard target examples for this type of problem. These are: Car model, Dream Chaser shuttle model, KCS hull model and the NASA CRM. 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 = 1 m. All of the numerical experiments have been performed on two different machines: the first has been devoted to the control cage generation and the MCAD sampling, while the second ran the bases computation and consequent spline fitting and IGA simulation. Their specifications are: *Windows 10, Intel(R) Xeon(R) CPU E3-1240 v6 @ 3.70GHz, 16.0*

GB RAM, 4 cores, and *Windows 10, Intel(R) Core(TM) CPU i7-9850 @ 2.60GHz, 16.0 GB RAM, 6 cores*, respectively.

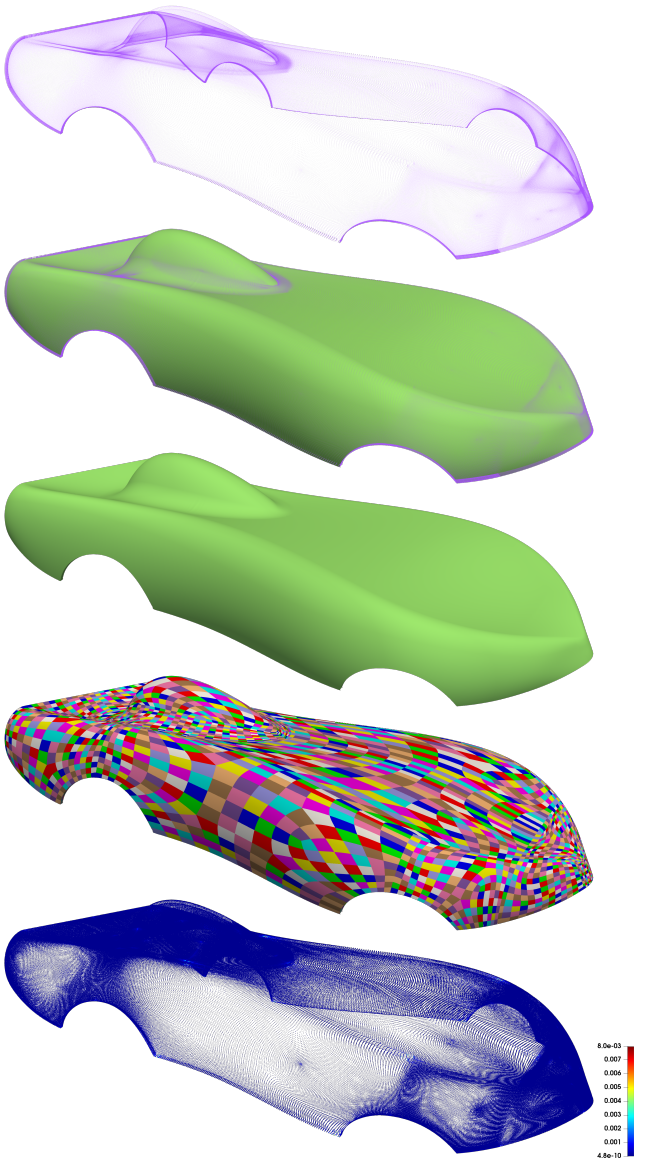

Figure 12: Car model. From top to bottom: point cloud sampling of the original MCAD model. Comparison between point cloud and resulting fitted surface. Surface in solid colour. Surface in multipatch colour. Error colour plot representing the ℓ_2 distance between the point cloud and the resulting surface.

Figures 12 to 16 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 colouring; the fitted surface in multipatch colouring and a colour plot representing the approximation error in Eucledian norm. Moreover, Table 1 summarises the dimensions of the starting point cloud, control cage and spline space together with the approximation errors evaluated with the formulae in (6)-(7). From the L^∞ errors represented in Table 1, as well as in the colour map of the models' colour plot, it can be noticed that the highest errors are located, as expected, around the EVs and near the sharp regions of the CAD 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 (see Fig. 15). 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 CAD model which are to be preserved. Defining on those edges only C^0 basis functions, our output surface will manifest these sharp features. Fig. 15 shows a detail of the KCS hull model where the top edge is first computed with G^1 smoothness, and then recomputed as a C^0 sharp feature.

Regarding IGA simulations, some experiments with the heat equation have been performed. With reference to eq. (10), we run the simulations for each of the four CAD 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. Fig. 17 to 20 present the results of the IGA simulations. These are obtained by setting, as initial conditions, an heat source at the likely location of the engines within the various vehicles represented. This demonstrates a realistic analysis of the thermal behaviour of such models. Moreover, Table 2 presents the running times required to compute each fragment of the pipeline.

	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
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 CAD model features for the experiments in Figures 12-13-14-16.

	Car	Dream Chaser	KCS	NASA CRM
Cage generation	1 min 45 sec	1 min 4 sec	55.25 sec	57.88 sec
MCAD sampling	22 min 28 sec	26 min 24 sec	22 min 23 sec	22 min 53 sec
Bases computation	11.39 sec	10.98 sec	10.04 sec	6.03 sec
Fitting	2 min 43 sec	3 min 11 sec	2 min 51 sec	1 min 42 sec
IGA simulation	15.65 sec/t_{step}	9.39 sec/t_{step}	7.39 sec/t_{step}	1 min 17 sec/t_{step}

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

7. Conclusion

In this work we presented a complete and efficient pipeline to convert CAD models into G^1 smooth objects which are suitable for isogeometric analysis simulations. Starting from a CAD object, we first produce a quad mesh whose Catmull-Clark limit surface adequately approximates the input CAD. Guided by this limit surface we compute a point cloud sampling of the CAD model, which is fitted with the use of basis functions defined over

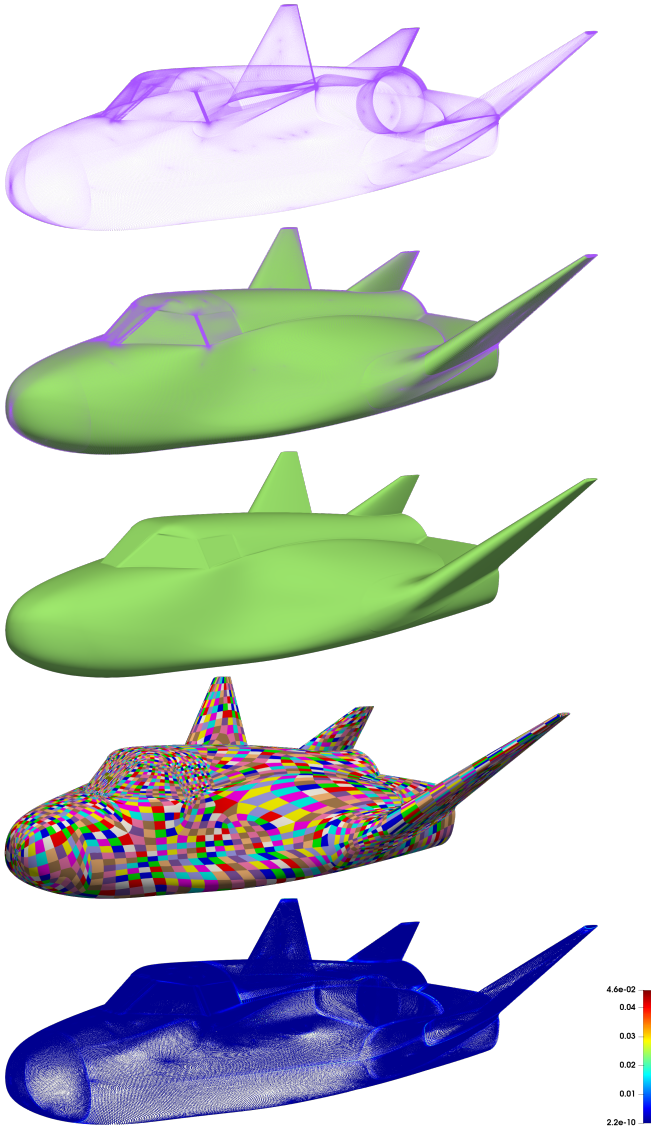

Figure 13: Dream Chaser shuttle model. From top to bottom: point cloud sampling of the original MCAD model. Comparison between point cloud and resulting fitted surface. Surface in solid colour. Surface in multipatch colour. Error colour plot representing the ℓ_2 distance between the point cloud and the resulting surface.

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 CAD models are provided with their error measurements. We illustrate the IGA simulations for the heat equation in real-life situations, highlighting the suitability of our approach for analysis.

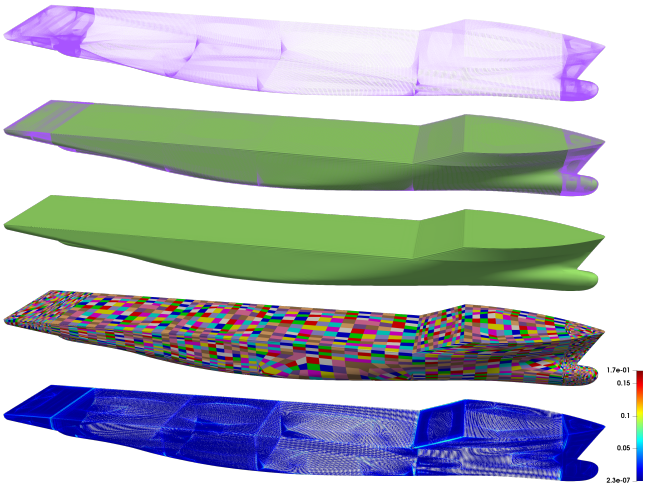

Figure 14: KCS hull model. From top to bottom: point cloud sampling of the original MCAD model. Comparison between point cloud and resulting fitted surface. Surface in solid colour. Surface in multipatch colour. Error colour plot representing the ℓ_2 distance between the point cloud and the resulting surface.

Figure 15: Zoom-in of the top edge of the KCS hull model. Left: smoothed edge with consequent oscillations due to the G^1 continuity. Right: sharp edge achieved by imposing C^0 regularity along the sharp edge.

References

[1] P. Alfeld and L. Schumaker. The dimension of bivariate spline spaces of smoothness r for degree $d \geq 4r + 1$. *Constr. Approx.*, 3(2):189–197, 1987.

[2] Z. Ali, J. Tyacke, P.G. Tucker, and S. Shahpar. Block topology generation for structured multi-block meshing with hierarchical geometry handling. *Procedia Engineering*, 26:212–224, 2016.

[3] A. Blidia, B. Mourrain, and N. Villamizar. G^1-smooth splines on quad meshes with 4-split macro-patch elements. *Computer Aided Geometric Design*, 52-53:106–125, 2017.

[4] A. Blidia, B. Mourrain, and G. Xu. Geometrically smooth spline bases for data fitting and simulation. *Comput. Aided Geom. Design*, 78:101814, 15, 2020.

[5] P. Boggs, A. Althsuler, A. Larzelere, E. Walsh, R. Clay, and M. Hardwick. DART system analysis. Technical Report SAND2005-4647 (OSTI), 2005.

[6] D. Bommes, T. Lempfer, and L. Kobbelt. Global structure optimization of quadrilateral meshes. *Computer Graphics Forum*, 30(2):375–384, April 2011.

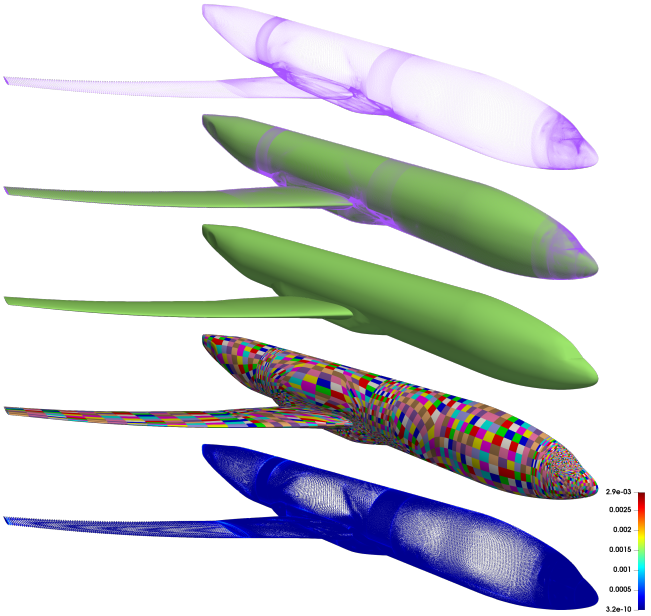

Figure 16: NASA CRM. From top to bottom: point cloud sampling of the original MCAD model. Comparison between point cloud and resulting fitted surface. Surface in solid colour. Surface in multipatch colour. Error colour plot representing the ℓ_2 distance between the point cloud and the resulting surface.

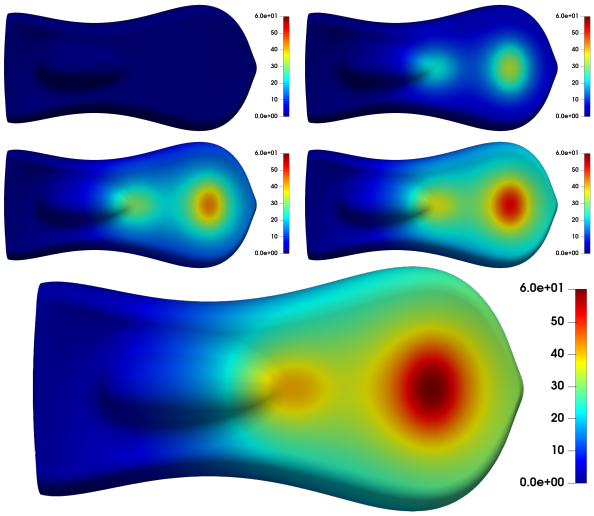

Figure 17: Solution for the heat equation on the Car model at the instants from $T = 0$ to $T = 0.4$ min.

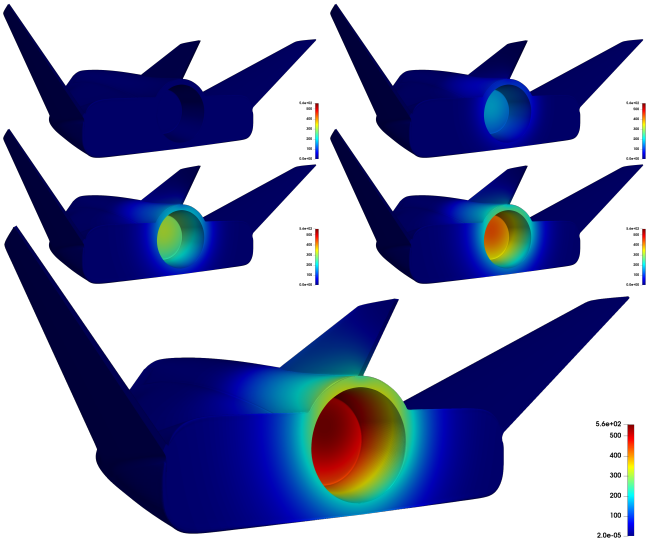

Figure 18: Solution for the heat equation on the Dream Chaser model at the instants from $T = 0$ to $T = 0.4$ min.

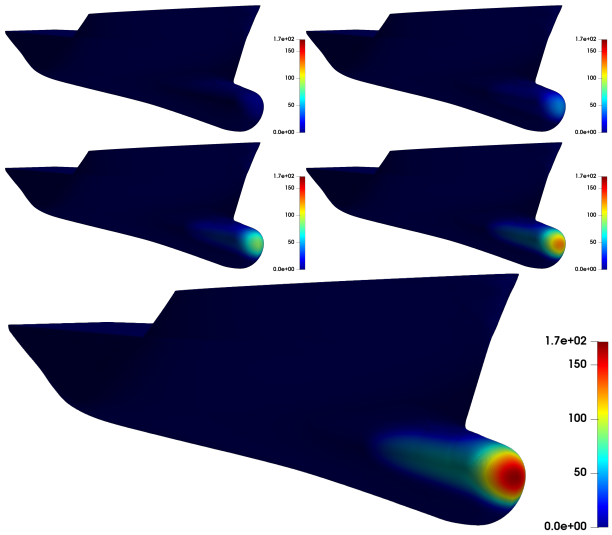

Figure 19: Solution for the heat equation on the KCS ship model at the instants from $T = 0$ to $T = 0.4$ min.

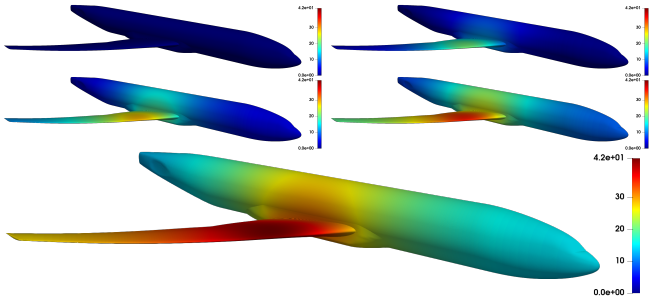

Figure 20: Solution for the heat equation on the NASA CRM at the instants from $T = 0$ to $T = 0.4$ min.

[7] G. Bonneau and S. Hahmann. Flexible G^1 interpolation of quad meshes. *Graphical Models*, 76(6):669–681, 2014.

[8] T.J. Cashman, U.H. Augsdörfer, N.A. Dodgson, and M.A. Sabin. NURBS with extraordinary points. *ACM Transactions on Graphics*, 28(3):1–9, July 2009.

[9] T.J. Cashman, N.A. Dodgson, and M.A. Sabin. A

symmetric, non-uniform, refine and smooth subdivision algorithm for general degree B-splines. *Computer Aided Geometric Design*, 26(1):94–104, January 2009.

[10] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Copmuter-Aided Design*, 10(6), 1978.

[11] J. Cottrell, T. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & sons, Ltd., 2009.

[12] M. Couplet, M. Reberol, and 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, July 2021.

[13] L. Dedè and A. Quarteroni. Isogeometric analysis for second order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 284:807–834, 2015. Isogeometric Analysis Special Issue.

[14] J. Fan and J. Peters. On smooth bicubic surfaces from quad meshes. In *International Symposium on Visual Computing*, pages 87–96. Springer, 2008.

[15] P.J. Frey and P.L. George. *Mesh Generation*. Wiley, January 2008.

[16] C. Geuzaine and 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):1309–1331, May 2009.

[17] W.N. Gordon and C.A. Hall. Construction of curvilinear coordinate systems and application to mesh generation. *International J. Num. Methods in Eng.*, 7:461–477, 1973.

[18] S. Hahmann, G. Bonneau, and B. Caramiaux. Bicubic G^1 interpolation of irregular quad meshes using a 4-split. In *International Conference on Geometric Modeling and Processing*, pages 17–32. Springer, 2008.

[19] D. Hong. Spaces of bivariate spline functions over triangulation. *Approx. Theory Appl.*, 7(1):56–75, 1991.

[20] M. Kapl, G. Sangalli, and T. Takacs. Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations. *Computer Aided Geometric Design*, 52-53:75–89, 2017.

[21] M. Kapl, G. Sangalli, and T. Takacs. Isogeometric analysis with C^1 functions on planar, unstructured quadrilateral meshes. *The SMAI Journal of computational mathematics*, S5:67–86, 2019.

[22] M. Kapl, G. Sangalli, and T. Takacs. An isogeometric C^1 subspace on unstructured multi-patch planar domains. *Computer Aided Geometric Design*, 69:55–75, 2019.

[23] K. Karčiauskas and J. Peters. Improved shape for refinable surfaces with singularly parameterized irregularities. *Computer-Aided Design*, 90:191–198, September 2017.

[24] M. Lai and L. Schumaker. *Spline functions on triangulations*, volume 110 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2007.

[25] C. Loop and S. Schaefer. Approximating Catmull-Clark subdivision surfaces with bicubic patches. *ACM Transactions on Graphics*, 27(1):8:1–8:11, 2008.

[26] Y. Ma and W. Ma. A subdivision scheme for unstructured quadrilateral meshes with improved convergence rate for isogeometric analysis. *Graphical Models*, 106:101043, 2019.

[27] M. Marsala, A. Mantzaflaris, and B. Mourrain. G^1-smooth biquintic approximation of Catmull-Clark subdivision surfaces. *Computer Aided Geometric Design*, 99:102158, 2022.

[28] M. Marsala, A. Mantzaflaris, and B. Mourrain. Point cloud fitting by G^1 smooth spline functions. hal-04002985, February 2023.

[29] B. Mourrain, R. Vidunas, and N. Villamizar. Geometrically continuous splines for surfaces of arbitrary topology. *Computer Aided Geometric Design*, 45:108–133, 2016.

[30] B. Mourrain and N. Villamizar. Homological techniques for the analysis of the dimension of triangular spline spaces. *J. Symbolic Comput.*, 50:564–577, 2013.

[31] J. Peters. Patching Catmull-Clark meshes. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 255–258, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[32] J. Peters and J. Fan. On the complexity of smooth spline surfaces from quad meshes. *Computer Aided Geometric Design*, 27(1):96–105, 2010.

[33] J. Peters and U. Reif. *Subdivision surfaces*. Springer, 2008.

[34] L. Schumaker. Bounds on the dimension of spaces of multivariate piecewise polynomials. *Rocky Mountain J. Math.*, 14(1):251–264, 1984.

[35] T.W. Sederberg, J. Zheng, D. Sewell, and 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.

[36] J. Shen, J. Kosinka, M.A. Sabin, and N.A. Dodgson. Conversion of trimmed NURBS surfaces to Catmull-Clark subdivision surfaces. *Computer Aided Geometric Design*, 31(7-8):486–498, October 2014.

[37] 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.

[38] M. Tarini, E. Puppo, D. Panozzo, N. Pietroni, and P. Cignoni. Simple quad domains for field aligned mesh parametrization. *ACM Transactions on Graphics*, 30(6):1–12, December 2011.

[39] D. Zorin, P. Schröder, A.D. Derose, L. Kobbelt, A. Levin, and W. Sweldens. Subdivision for modeling and animation. 2000. Course Notes of SIGGRAPH.