axioms

MDPI

Article

From Topological Optimization to Spline Layouts: An Approach for Industrial Real-Wise Parts

Carolina Vittoria Beccari [1,*,†], Alessandro Ceruti [2,†] and Filip Chudy [1,3,†]

[1] Department of Mathematics, University of Bologna, Piazza di Porta San Donato 5, 40126 Bologna, Italy; filip.chudy@cs.uni.wroc.pl
[2] Department of Industrial Engineering, University of Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy; alessandro.ceruti@unibo.it
[3] Institute of Computer Science, University of Wrocław, ul. Joliot-Curie 15, 50-383 Wrocław, Poland
* Correspondence: carolina.beccari2@unibo.it
† These authors contributed equally to this work.

Abstract: Additive manufacturing technologies have allowed the production of complex geometries that are typically obtained by applying topology optimization techniques. The outcome of the optimization process is a tessellated geometry, which has reduced aesthetic quality and unwanted spikes and cusps. Filters can be applied to improve the surface quality, but volume shrinking and geometry modification can be noticed. The design practice suggests manually re-designing the object in Computer-Aided Design (CAD) software, imitating the shape suggested by topology optimization. However, this operation is tedious and a lot of time is wasted. This paper proposes a methodology to automate the conversion from topology optimization output to a CAD-compatible design for industrial components. Topology optimization usually produces a dense triangle mesh with a high topological genus for those objects. We present a method to automatically generate a collection of spline (tensor-product) patches joined watertight and test the approach on real-wise industrial components. The methodology is based on the use of quadrilateral patches which are built on the external surface of the components. Based on the tests carried out, promising results have been obtained. It constitutes a first step towards the automatic generation of shapes that can readily be imported and edited in a CAD system.

Keywords: topology optimization; spline; 3D modeling; additive manufacturing; computer-aided design

MSC: 65D17; 65D07; 68U07

Academic Editor: Behzad Djafari-Rouhani

Received: 24 November 2024
Revised: 26 December 2024
Accepted: 9 January 2025
Published: 20 January 2025

Citation: Beccari, C.V.; Ceruti, A.; Chudy, F. From Topological Optimization to Spline Layouts: An Approach for Industrial Real-Wise Parts. *Axioms* **2025**, *14*, 72. https://doi.org/10.3390/axioms14010072

1. Introduction

The modern industry requires lightweight structures. Aircraft, trains, or cars are used to commute in everyday life or travel long distances for work, leisure, or to carry goods: this has an overwhelming impact on the environment due to unsustainable fuel consumption and consequent pollution. Also, when electric energy is used, the reduction in structural masses allows for larger autonomy or increased payload. In the case of industrial applications, mass reduction can lead to increased accelerations, which can be obtained with constant stresses on components: in this way, automatic machines can increase production volumes with better resource exploitation [1]. Therefore, developing and producing lightweight structural components is an immediate and practical response to the needs of modern engineering [2], but it poses several research challenges. Topology

optimization (TO) can be used to obtain components with constant stress at each point, thus minimizing the weight-to-safety coefficient ratio. TO is based upon the following steps [3]: (1) meshing of the volume where the component can grow (control volume), (2) finite element structural analysis where the stress level (or stiffness) is evaluated in every node of the mesh (3) application of strategies to delete material where stress levels are low, and add it where stress is high (with filters to avoid sparse zones and allow continuity) (4) looping around steps (2) and (3) up to arriving at a point in which the geometry does not change and the convergence is reached. Therefore, the TO output is a mesh deriving from deleting tetrahedral or cubic elements to a 3D mesh. Typically, the shapes obtained from the topology optimization have a complex hollow structure; in mathematical terminology, they have a very high topological genus. Such shapes cannot be manufactured using traditional machining processes based on chip removal, such as lathing or milling, or with foundry due to undercuts and shape complexity. Only recently [4] have industries been able to produce objects of such complexity thanks to the widespread use of additive manufacturing machines [5]. Moreover, additive manufacturing is appealing because lattice structures and functionally graded structures can be produced [6].

TO techniques rely basically on three approaches. The Solid Isotropic Material with Penalization (SIMP) was introduced by [7] and is used to maximize the stiffness and optimize the stress distribution in a 3D part (called minimum compliance problem). SIMP updates the 3D model step by step, following gradient-based rules. With regard to material density, SIMP uses a continuous distribution. The Evolutionary Structural Optimization methodology (ESO) introduced in 1996 [8] relies on a dense control volume. The methodology is iterative, and the material is subtracted at each step. Bi-directional ESO (BESO), as proposed by [9] has been introduced to improve the ESO approach: material can not only be subtracted from the body but also added to the points where stresses are too high. In this way, if the subtraction of material weakens a zone, in the next iteration material can be added to optimize the stress distribution. In addition to the traditional SIMP and BESO methodologies for TO, other approaches have also been developed in the literature for structural shape optimization [10,11].

The workflow from a traditional "completely dense" design to an equivalent "hollow" optimized lightweight structure consists of two main stages. In the first stage, TO is applied: a designer defines a control volume, sets the forces acting on the object and constraints (to avoid rigid motion), and a set of finite element simulations is run: as already stated, this is necessary to determine where material can be subtracted because there is low or null stress. The optimization output is a tessellation made of voxels or tetrahedra, providing a rough indication of the final object's appearance. But this shape is far from being manufactured: there is no continuity between elements, flat surfaces may appear undulated, and cusps and other imperfections can be found; moreover, a superficial mesh is obtained, which can be smoothed but not edited. To cope with those problems, a second stage of the design process consists of redesigning the component manually from scratch in a computer-aided design (CAD) system, trying to imitate the result from TO as accurately as possible. This second stage is critical: time is lost in redesigning the object; with manual modeling, the result of topological optimization is approximated, and therefore, the obtained shape could be heavier or less resistant than the optimized one; in the case of complex geometries, it is challenging to replicate the shapes that come out from topological optimization with the available CAD tools. Often, features such as support surfaces, holes, and flat faces should be kept the same: a way to protect their shape is excluding some zones of the component from the control volume, but this leads to problems and discontinuity. There is thus a consistent gap between the first stage of topology optimization, which, though often not fully satisfactory, is an automatic procedure, and the second stage of shape reconstruction,

which is almost totally based on human work. This can increase the workload to apply TO in industrial applications, limiting it to high-value cases.

Several recent studies have focused on automating the process from TO's output to a fully redesigned 3D model. They provide mathematical and numerical methods to automatically generate a smooth surface from voxels or tetrahedral structures like those produced by TO. Ideally, this surface should not only resemble the hollow shape of the final object as much as possible, but it should also have specific mathematical properties (e.g., high continuity, adequate parametrization, interpolation of special features, and possibly others) to be suitable for its intended use. Last, a mathematical representation compatible with the CAD standards should be provided for simple communication, editing, and combining the results into more complex designs. The capability to edit an optimized model can be paramount in adding features in the modeled 3D part or implementing changes without needing another TO, which can require long computational times. State-of-the-art numerical methods for surface parameterization cannot effectively address all of these challenges. A strong interaction between mathematics/numerical analysis and industrial engineering is necessary to understand the problems faced in everyday design problems: the mathematical basis is useful to obtain refined methodologies showing numerical efficiency, precision, and robustness.

Existing studies have focused on specific objects, particularly beam-like structures or simple geometries, whose interest is more academic than practical. A more detailed discussion is provided in the next section. This research, instead, started from real test cases arising from applications in industrial and aerospace engineering. The aim of this paper is to provide mathematical and numerical methods to automatically generate a smooth surface from a triangular mesh produced by topology optimization. Our goal poses a significant challenge in several ways.

Figure 1 shows a typical triangular mesh from topology optimization. Such a mesh can be obtained using TO tools embedded in commercial CAD packages. Typically, it is a dense mesh made of several thousands of triangles, and the vertices are scattered irregularly; that is, they do not belong to a smooth surface. To improve the quality of this mesh, one may use smoothing methods such as [12–16]. However, those do not provide a mathematical representation of the surface, the features to be preserved can be affected, and the volume of the body can be altered. More specifically, smoothing tools do not produce a parametric surface but rearrange the position of the mesh vertices to reduce the noise and spikes. They output a triangulated surface which is adequate to prepare the path that the head of additive manufacturing machines should follow, or the movements required to lower the model once a layer has been completed, but cannot be edited using CAD software. Moreover, they often significantly alter the mesh's dimensions and, due to their global nature, such methods can be inefficient or even unfeasible for large datasets. In general, smoothing methods do not reduce the size of complex meshes with millions of vertices, even in the case of flat surfaces, which only four vertices could represent. Those embedded in commercial CAD software (such as OptiStruct and SolidThinking by Altair Engineering and Tosca by Dassault Systemes) are more effective in handling complex datasets. Still, the codes and procedures are not available in the literature, and no details have been published.

The paper reports research aimed at developing a methodology to obtain a CAD model from TO results, overcoming current limitations. As will be detailed in Section 1, methods in the literature show drawbacks such as they can be applied to a limited set of shapes; require human intervention (are semi-automatic); cannot recognize holes, flat surfaces, or other areas that should be preserved. Unlike existing methods, the approach presented in this paper is fully automatic, applies to objects of any shape and genus, and preserves holes and flat faces.

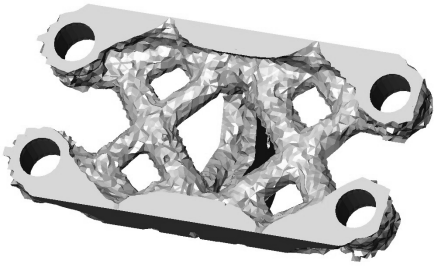

Figure 1. A typical triangular mesh resulting from topology optimization: it is a dense mesh whose vertices are affected by spurious undulations.

Essentially, the proposed approach aims to convert the TO mesh into a tensor-product spline surface. Generating a spline layout from a triangulated data structure is a complex task since spline surfaces are typically made of quadrilateral parametric polynomial (macro) patches stitched together with some order of continuity. First, generating a suitable quadrilateral layout from a dense and noisy mesh is complicated. For instance, all approaches that start with the extraction of a skeleton (or medial axis) (see, e.g., [17] and references therein) fail for objects with high topological genus (the *genus* is, roughly speaking, the number of holes). Even when a quadrilateral layout has been established, generating a spline surface that does not suffer from the noisy nature of the mesh data are nontrivial.

The method proposed here consists of two stages: first, we segment the TO mesh and extrapolate a quadrilateral layout. Successively, a spline patch corresponding to each quadrilateral is created in such a way that the resulting surface is globally watertight (i.e., the surface patches are glued together with at least C^0 continuity) and is free from the spurious oscillations present in the data. To do so, we rely on a combination of algorithms proposed in the context of mesh and surface processing [18–20], adapted to our context. The resulting spline surface can be resampled, thus generating a triangulation ready for additive manufacturing. Moreover, algorithms that use the polynomial (spline) representation can further smooth it.

The remainder of the paper is structured as follows: In Section 1, we review the existing literature and the downsides of existing methods. Section 2 presents an overview of the process for reconstructing a spline surface from the initial dense mesh. Numerical results are illustrated in Section 3. Finally, Section 4 discusses the results and future research directions.

Approaches from Topology Optimization to CAD Models

A wide literature supports the need for automatically obtaining a CAD model from TO, considering structural needs. The diagram in Figure 2 visually illustrates the characteristics of different approaches in comparison.

A first approach to deal with a smooth geometry after a TO process has been addressed in [21]. After applying the SIMP TO algorithm, two methodologies have been evaluated to shift from a density distribution to a 3D body. The first relies on setting a threshold density, keeping the elements with higher density, and applying smoothing algorithms such as the one described in [22]. The second one includes the computation of iso-density surfaces from a continuous approximation of the density along the three directions x, y, and z. In this case, better quality of the surfaces is obtained with respect the first method, but a smoothing algorithm must be applied anyway. It is worth remembering that this method can be applied to SIMP TO algorithms only, requires raw data, and relies on filtering algorithms that change volumes and shapes that should be preserved.

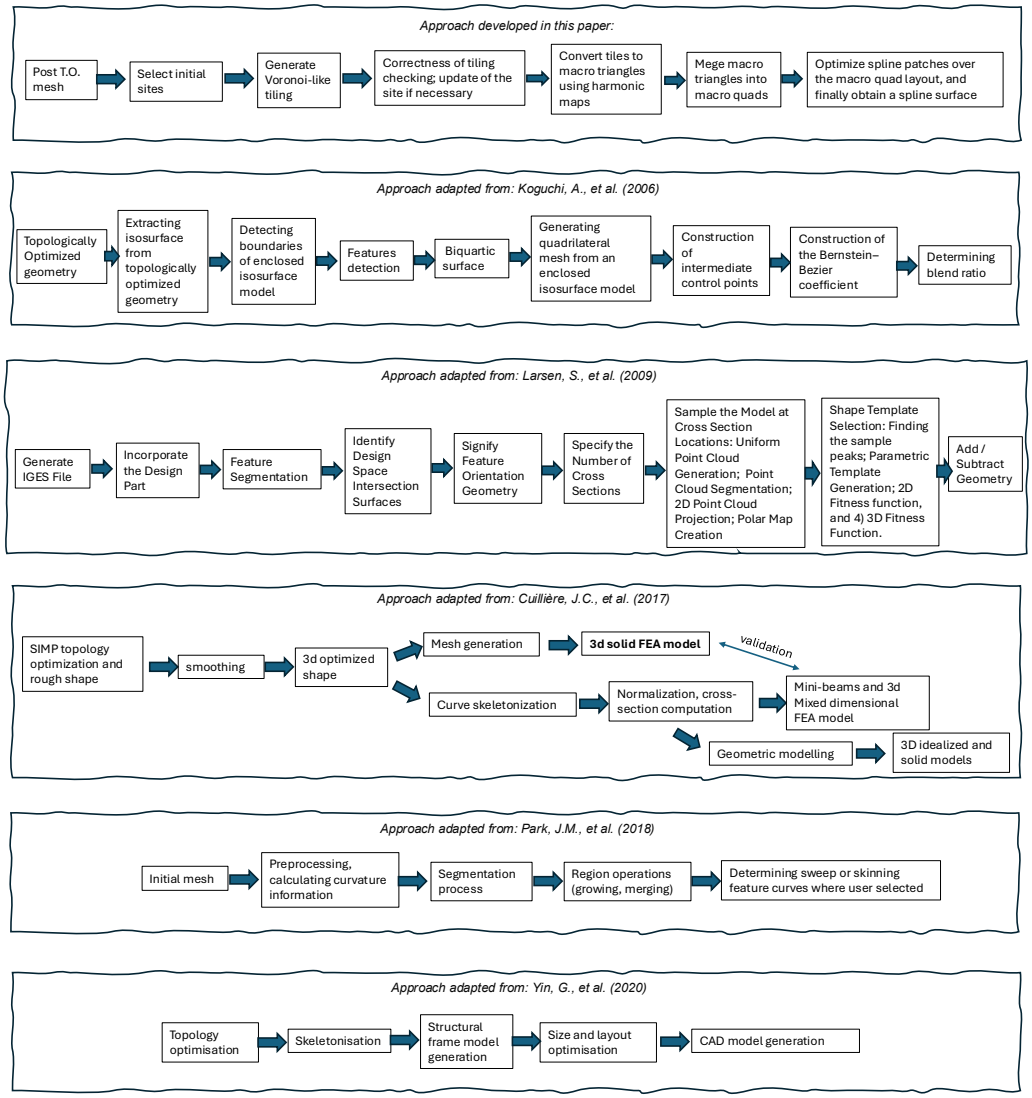

Figure 2. Workflow of different approaches for generating CAD compatible models from TO, in particular [23–27].

The advancement of the above procedure can be found in [28] where the Unified Topological Model (UTM) has been introduced to achieve better integration between CAD, FEA, and TO methods. An ensemble of boundary representation (B-Rep), mesh generation, and other geometric tools was developed for the correct management of non-manifold geometry and multi-dimensional models.

The evolution of the two above-cited papers can be found in [23,29]. The authors describe fully automated methods to obtain a 3D CAD model from TO that tends towards beam-like structures. At first, raw TO results are translated into a smooth triangulation. Secondly, a curve skeleton is obtained from the triangulation, which is normalized to generate a CAD model made of an assembly of straight beams. An FEA validation has been carried out to show that the converted CAD model shows similar structural behavior with respect to the pristine geometry. The limit of this kind of approach lies in the fact that only thin geometries can be translated into beams, while dense parts of the 3D model can be cumbersome to treat with this methodology. Moreover, flat zones cannot be approximated well with cylindrical beams.

More recently, a new strategy was tested in [30], where a fully automatic process to convert 3D TO into CAD models is presented. Curve skeletonization and boundary triangulation are applied to obtain smooth shapes: Cross-sections are interpolated using

cubic B-spline-based lofting functions to obtain 3D models with unchanged structural performances with respect to the original triangulation. This approach's main limitation is that it can be applied only to cases where the output of TO can be approximated with beam-like structures.

Another original method to obtain a CAD model from TO has been developed in [24]: in this case, a semi-automatic procedure for converting TO into a smooth and parametric CAD model is described. A set of predefined 2D shape templates are fit to the geometry from TO, and sweeping is applied to a surface to create 3D features. However, this procedure requires user intervention, and it is not automatic. Moreover, numerous datasets of predefined 2D shapes must be developed and integrated to apply the methodology to whatever shape.

In the paper [25], a surface reconstruction procedure has been developed to obtain a CAD model from the outcome of a topology optimization process. An enclosed isosurface model is extracted from the TO result after a threshold density has been set: the marching cubes method for tetrahedral, pentagonal, and hexagonal elements is applied. Afterward, a quadrilateral patch model is built on the isosurface. Tests with this approach have yet to be carried out to check whether holes or other features are preserved.

A method based on Constructive Solid Geometry (CSG) is proposed in [26]. It describes an automated and topologically accurate methodology where the topology-optimized structure is converted into a spatial frame structure and then regenerated in a CAD system using standard CSG operations. Voxelization and the combination of primitive solids like spheres and cylinders are subject to boolean operations. The final solid model is a B-Rep made of trimmed non-uniform rational B-spline (NURBS) surfaces and curves. Currently, an automated process to extract a shell surface from the skeletonized voxel model has not been developed, and the only primitives that can be used are spheres and cylinders.

The paper [31] proposes a way to smooth the result from a TO process with a global approach. Still, the final aim of the work is to develop a strategy to alter the initial triangular preserving shapes such as holes or flat surfaces. Shifting from a triangular mesh to a CAD model needs to be addressed.

The most promising techniques for obtaining a CAD model from the TO outcome are based on spline-based surfaces and curves. A methodology for automatically modeling extruded 3D parts from topology optimization was developed by [32]. It is based on an image analysis technique called ASOS in which the contour of 2D optimized bodies (internal and external) is detected and interpolated with B-spine curves. The methodology was further improved in [33] with the development of the IASOS procedure. Both suffer from the limitation of applying only to extruded 2D shapes.

The paper [34] proposes a methodology where sections are described through B-Spines and a loft is then applied to obtain solid bodies inside CAD software. At first, the geometry from TO is converted into smooth parametric B-spline surfaces and curves. Those curves and surfaces are successfully imported into CAD parametric software, and a manual lofting operation is needed to obtain the solid body. Examples of 3D shapes obtained from extruding 2D shapes are presented in the paper.

A more general approach has been developed in [27], which can be applied to whatever 3D shape. Using the *k*-means clustering method, the mesh is segmented into a set of regions; then, each region is converted into a free-form B-spline surface. Finally, feature curves are calculated to create loft or sweep surfaces, minimizing errors in distances. This approach's main drawback is that it applies only to models without inner holes, significantly limiting its applicability to real industrial parts.

To sum up, the problem of integrating automatic procedures into the optimization loop to obtain a CAD model is still open. The approaches described in the literature work

well with sample bodies, but poor performances are usually noticed when real components are used. Typical industrial parts show a strong complexity, with a very high topological genus that is demanding from a theoretical and computational point of view. The literature lacks a methodology that can be automatically applied to whatever shape.

2. Methodology for Transitioning from Topology Optimization to Spline Patches

This section introduces the methodology for converting a dense mesh resulting from TO into a tensor-product spline surface. A two-stage approach is pursued (see Figure 3): the first stage consists of extrapolating a quadrilateral layout from the initial mesh. This quadrilateral layout is a mesh with quadrilateral macro faces. We use the adjective "macro" to identify this mesh, in contrast with the data mesh, whose triangles are relatively small and which we call a *dense mesh*. The computation of the quadrilateral layout uses ideas presented in [18,19] in the context of mesh processing. The first paper proposes a framework for generating a multiresolution form of a dense triangle mesh with arbitrary connectivity. The method produces a tessellation where each tile is formed by a set of nearby triangles. Then, it extracts a macro-triangulation from this tiling whose meshes corresponding to different levels of detail can be generated. The second paper deals with reconstructing a tensor product B-spline surface from a set of scanned 3D points. To do so, a quadrilateral layout is derived by coupling adjacent triangles of a triangulated mesh. The methods proposed in those papers (intended for meshes with relatively few triangles and free from unwanted oscillations) fail for complex meshes like those produced by TO. Consequently, we describe our implementation here, which allowed us to successfully process the real-wise objects targeted in our research.

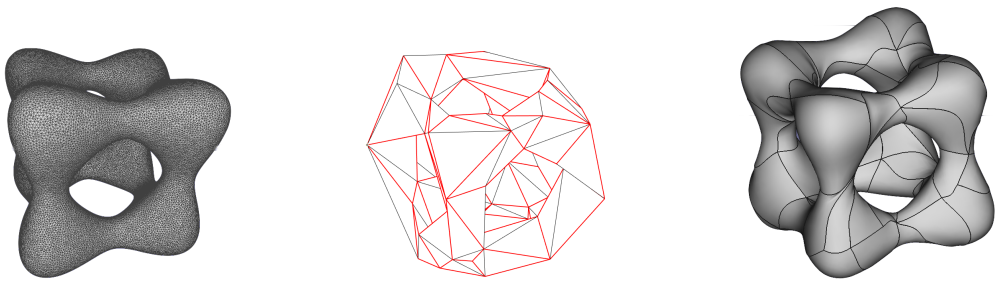

Figure 3. Stages of the algorithm's execution, from left to right: TO mesh (i.e., a dense mesh with triangular faces); quad layout (i.e., a mesh with quadrilateral macro faces); complete spline surface.

The second stage of our approach consists of constructing a spline surface, where each spline patch corresponds to one quadrilateral. The resulting surface must be globally watertight (i.e., the surface patches are glued together with at least C^0 continuity) and free from the spurious oscillations present in the data. To this aim, we pursue a local least-squares approach [20], where each surface patch is computed independently, thus allowing for parallel computation and reduced computer memory consumption, which are vital when dealing with large datasets as in our setting.

Figure 4 presents a flowchart of the algorithm to pass from a TO mesh to a spline layout described in the following subsections. The procedure, in particular, undergoes the following stages:

- Partition the TO mesh into tiles (Section 2.1).
- Generate a quadrilateral patch layout, i.e., a macro-mesh with quadrilateral faces (Section 2.2).
- Construct a spline patch corresponding to each quadrilateral of the previous layout (Section 2.3).

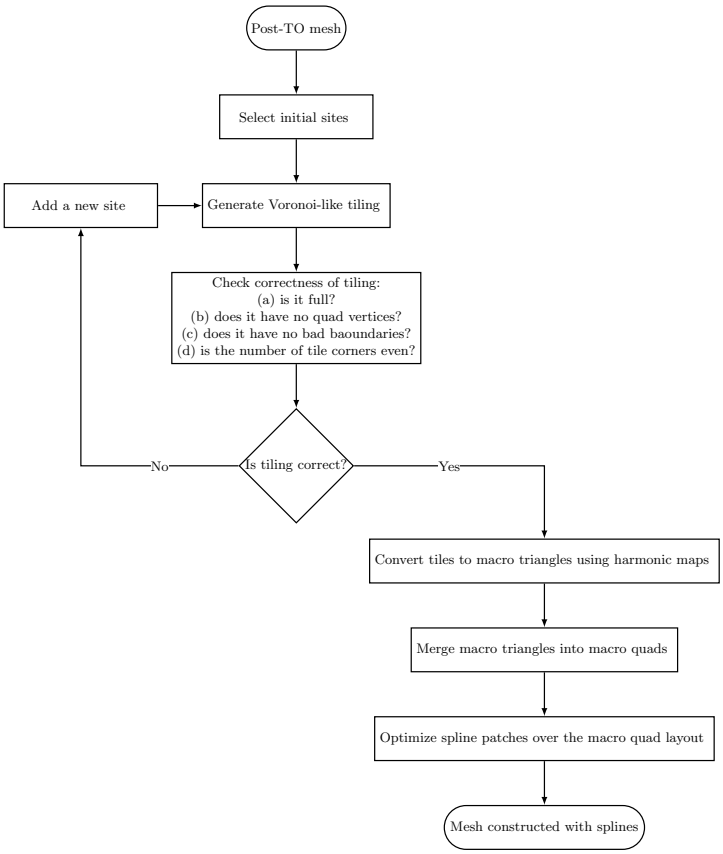

Figure 4. Flowchart of the algorithm to pass from a TO mesh to a spline layout.

2.1. Partitioning the TO Mesh

The first necessary step is to construct a partition of the TO mesh into regions, or *tiles*, each consisting of several abutting triangles. Each tile is associated with a special point, called a *site*. In our case, the sites correspond to the centroids of the mesh triangles, see Figure 5 Left for a schematic representation.

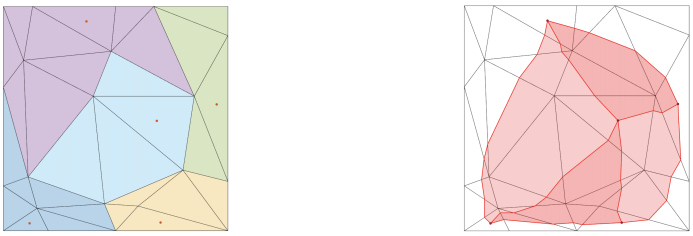

Figure 5. (**Left**): Partitioning of the TO mesh into tiles: each color identifies a different tile and the red dots correspond to the tiles' site; (**Right**): An excerpt from the macro-triangulation obtained by connecting the tiles' sites: the figure shows four macro-triangles colored in different shades of red. The red thick lines represent the macro-triangles "edges", which are sequences of edges in the TO mesh.

Given a finite set of sites S, the Voronoi diagram for S is the partition of the mesh that associates a region $V(p)$, consisting of a set of triangles, to each point $p \in S$ such that the centroids of all triangles inside $V(p)$ are closer to p than to any other point in S. The distance function that we use is the length of the shortest path in a weighted neighborhood graph $G = (V, E)$ where V is the set of all faces and, for $f_1, f_2 \in V$, $(f_1, f_2) \in E$ if and only if the triangles f_1 and f_2 have at least one edge in common. The weight function is then the Euclidean distance between the centroids of f_1 and f_2.

The partition must have the additional properties introduced in [18], and therefore, from now on it will be called an *augmented Voronoi tessellation* since it is a particular subtype of Voronoi diagrams. In [18] a partition satisfying the properties in Definition 1 is called a Voronoi-like diagram. We use different terminology because, despite our algorithm's result being a Voronoi-like partition, the methods do not have to yield the same partition due to some passages of the algorithm being different.

Definition 1 (Augmented Voronoi tessellation)**.** *Given a set of sites, we define an augmented Voronoi tessellation to be a Voronoi partition of the mesh that has the following properties:*

(a) *each tile is homeomorphic to a disk;*
(b) *each vertex of the mesh is incident to at most three tiles;*
(c) *no pair of tiles may share more than one contiguous set of edges;*
(d) *the tessellation constitutes a complete covering of the mesh, i.e., each face of the original mesh is included in exactly one tile.*

Clearly, given an arbitrary mesh and an arbitrary set of sites, it is not guaranteed that an augmented Voronoi partition based on the assigned sites exists. Hence, to successfully generate such a partition, the sites (or, more precisely, most of them) are not chosen a priori. More precisely, the method is initialized by specifying an initial set of sites, which can be empty. Hence, the tessellation is grown by dynamically inserting the sites one by one. In the remainder of the section, we describe our implementation of the procedure, which has proven to effectively generate the tiling for objects of our target level of complexity.

Initialization—To start the algorithm, an initial set of sites is necessary. We denote specific vertices as points of interest. Each vertex is greedily associated with the closest unassociated face (w.r.t. the vertex-centroid distance). If fewer than seven vertices have been specified, additional faces are randomly chosen as sites to reach that number. We chose seven sites because the simplest possible quadrangulation is a cube, and to obtain it, we need eight sites, seven picked here and one more later.

Growing the tiles—Once initialization is complete, the following loop begins, which ends when the tiling passes all the checks below. First, a new site is added (the first site is added randomly; the successive steps in the loop will determine the next one). The Voronoi tiles are grown (see next paragraph). Then, a list of vertices that do not comply with condition (b) (which we call *quad vertices*, i.e., vertices where four or more tiles meet) and bad boundaries with respect to condition (c) is compiled. The tiling is checked for failures, which are the cases below. Note that adding a site through *Case X* means that no sites have been added in *Cases 1* to *(X − 1)*, and no sites will be added in *Cases* (*X* + 1)–*end*.

Case 1: Tiling does not cover the whole mesh, i.e., it does not satisfy (d). This can happen in two situations: One is when tile growth fails when adding a new face to a tile. As explained in detail in the next paragraph, the failure occurs when a tile does not have the disk homeomorphism property—the tile growth is stopped upon the first face, the addition of which would violate the disk homeomorphism property. The faces that stopped the tiles' growth are considered candidates for a new site. From the faces, the one closest to its corresponding site is chosen—if choosing a face would guarantee the creation of a quad vertex, it is omitted, and another one is chosen. In this case, a list of faces that are edge-incident to the tile that failed is created. For each of them, one of their vertices is chosen as a candidate new site (if, for a face, all vertices are used for other sites, the face is omitted). Then, one of the pairs $(face, vertex)$ is chosen randomly as a new site. The other situation is when there are no more faces to process while not all faces have been added. We have

never encountered such a situation in practice. Should it happen, the face that is furthest from any site is chosen.

Case 2: There are bad boundaries, i.e., not satisfying condition (c). A bad boundary between two tiles occurs if their boundary is not continuous. The candidates for new sites are faces that are incident to a bad boundary. If a bad boundary fragment has at least one edge, faces that have a common edge with it are considered. If a bad boundary fragment consists of just one vertex, vertex-incident faces are added to the list. Faces that would guarantee a quad vertex are discarded; from the remainder, a new one is picked randomly as a new site.

Case 3: There are quad vertices, i.e., not satisfying (b). Faces that are vertex-incident to at least one quad vertex are determined as possible locations for new sites. Ones that are sites or that would cause a quad vertex are filtered out. Then, one is picked at random.

Case 4: Odd number of triangles. A (macro) triangulation is formed by connecting the abutting tiles sites, as detailed in Section 2.2. At this stage, we count how many triangles would be obtained in this way (this count is performed by counting the number of tile corners). If the number of triangles is odd, then the face which is the furthest from any site is chosen. The number of triangles must be odd because, at a later stage, the (macro) triangles are paired up into (macro) quads.

How the tiles are grown—When initializing a tiling, i.e., placing the first site, the distance of all the faces to the site is computed (distance is a sum of Euclidean face–face distances between centroids of neighboring faces) using a shortest path algorithm (more precisely, a multi-source variant of Dijkstra's algorithm). The purpose of this algorithm is to divide the mesh into zones, as an optimization of retiling the mesh in each iteration. When there is only one site, the whole mesh will be one zone. For two sites, we will have two zones, and so on. Basically, the partition into zones is a proper Voronoi partition, i.e., without the additional constraints introduced in Definition 1.

When adding a new site, the shortest path algorithm is rerun, redrawing the zones. The effect is usually local, so not many zones will be redrawn. For each zone that was modified, a tiling algorithm is run.

The tiling algorithm first checks whether the zone that needs to be tiled is homeomorphic with a disk by checking the Euler characteristic of the zone, i.e., if $v - e + f = 1$, where v is the number of vertices, e is the number of edges, and f is the number of faces in a zone. If it is, the whole zone becomes a tile. If not, we would tile the zone until failure.

The tiling of a zone works as follows: A site is added to a tile. Then, the shortest path algorithm is run to obtain an order in which we add faces to the tiling. If adding a new face violates disk homeomorphism, the tiling fails, returning the ID of the face in which the failure occurred.

2.2. Generating a Quadrilateral Patch Layout

The property (d) in Definition 1 makes it so that two abutting tiles have a set of contiguous edges in common, which we call a *tile boundary edge*. In addition, thanks to property (b), a macro-triangulation is determined by connecting the sites of each couple of abutting tiles across their common boundary edge. This structure is dual to the augmented Voronoi tiling, forming a Delaunay triangulation. The edges of the Delaunay triangles are sequences of mesh edges (see Figure 5 Right, for a schematic representation) computed as follows. Each tile is converted into a planar polygon using the harmonic map procedure described in [18] (with the polygon corners being the vertices at which three Voronoi tiles meet). Then, the centroid is connected, using a segment, with the midpoint of each side of the resulting polygon. The faces intersecting with the segment are refined so that the

segment consists of a sequence of mesh edges. If a segment intersects a face's interior and its corner, a simple refinement into two triangles follows. However, if it intersects two sides of the face, the face is divided into a triangle and a tetragon. The tetragon is further triangulated in a way that minimizes the maximum angle. This procedure subdivides a Voronoi tile into several subpatches. Three subpatches, incident to the same polygon corner, are assembled to form a macro-triangle patch.

We guarantee that the number of macro-triangles is even. Therefore, the macro-triangles can be paired to create bigger macro-quad patches. To decide which triangles need to be paired, the concept of harmonic maps is used as proposed in [19]. An undirected weighted graph is created where the vertices denote the macro-triangles, and the edges indicate which macro-triangles are edge-adjacent. The edge is weighted using the harmonic energy formula, i.e., for two triangles (o_1, d_1, d_2) and (o_2, d_2, d_1), the harmonic energy h is given by the formula

$$h(o_1, d_1, d_2, o_2) = \frac{\|o_1 - d_1\|_2^2 + \|o_1 - d_2\|_2^2}{area(o_1, d_1, d_2)} + \frac{\|o_2 - d_1\|_2^2 + \|o_2 - d_2\|_2^2}{area(o_2, d_1, d_2)},$$

where $area(a, b, c)$ is the area of the triangle having vertices a, b, c. The harmonic energy is expected to be the lowest when the two triangles, when put together, are the most square-like. The aim is to pair up the macro triangles in a way that minimizes the total harmonic energy of the connections. In order to do so, a minimum-weight maximum matching algorithm is used.

The macro-quads need to satisfy additional conditions in order to use the spline construction method used further in the process. Namely, every pair of macro-quads needs to have at most two corners in common. This condition can be violated in three distinct ways:

1. Three corners in common; two consecutive macro-edges in common.
2. Three corners in common; one macro-edge in common.
3. Four corners in common.

In the first case, the remaining four macro-edges form a boundary of a bigger macro-quad and thus the two macro-quads can be combined into it. In the second and third cases, the connections that formed the problematic macro-quad are marked as problematic. Then, one of the problematic connections is banned and the matching algorithm is rerun with backtracking.

The resulting macro mesh consists of quadrilateral faces, and each pair of faces has at most two common corners.

2.3. Constructing a Spline Patch Corresponding to Each Quadrilateral

To construct a spline patch for each quadrilateral generated in the previous section, we pursue the method in [20], which we summarize here for completeness. We chose that approach because of its local nature. Each spline patch is determined by solving a small size minimization problem, which can be solved numerically in a very reasonable computational time. In this way, large datasets are processed efficiently. In addition, the locality of the method also allows for computing several spline patches simultaneously.

A spline space is identified by a nonnegative integer d, representing the degree, and by a knot vector, which is a sequence having the form $\Xi = \{0, \ldots, 0, \xi_1, \ldots, \xi_m, 1, \ldots, 1\}$, where the ξ_i are nondecereasing real values in the interval $(0, 1)$, and 0 and 1 are repeated $d + 1$ times. The space of spline functions determined by d and Ξ has dimension $M = d + 1 + m$ and possesses a B-spline basis, which we denote by $N_{i,d}$, $i = 1, \ldots, M$. A tensor product spline patch (see, e.g., [35]) is defined by the following equation:

$$\mathbf{S}(u,v) = \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} \mathbf{s}_{i,j} N_{i,d_1}(u) N_{j,d_2}(v), \quad (u,v) \in [0,1]^2,$$

where d_1 and d_2 are the degrees corresponding to the two grid directions, Ξ_1 and Ξ_2 are the relative knot vectors and M_1, M_2 the dimensions of the spline spaces. The B-spline coefficients $\mathbf{s}_{i,j}$ are referred to as control points.

In our context, each quadrilateral of the quadrangulation determined in Section 2.2 gives rise to a different spline patch of which we need to define the control points. According to [20], the control points are computed as the solution of a least squares minimization problem so that the spline patch has a minimum distance from the mesh vertices in a nearby area. In particular, let $\mathbf{p}_i$ be the points to be approximated. Then, we first assign each point $\mathbf{p}_i$ parameter values (u_i, v_i). Hence, the spline coefficients are determined to minimize

$$\lambda \sum_i \|\mathbf{S}(u_i,v_i) - \mathbf{p}_i\|_2^2 + \mu \int_0^1 \int_0^2 \|\mathbf{S}_{uu}(u,v)\|_2^2 + 2\|\mathbf{S}_{uv}(u,v)\|_2^2 + \|\mathbf{S}_{vv}(u,v)\|_2^2 du dv, \quad \mu = 1-\lambda$$

for some $\lambda \in (0,1)$. The first term above minimizes the distance between the surface and the data points; the second term describes the bending energy of a thin plate, and thus its minimization promotes a smooth-looking solution. For each patch, the points $\mathbf{p}_i$ are the vertices of the TO mesh inside the corresponding macro-quadrilateral (indicated in light-blue in Figure 6) and the adjacent ones (in grey in the same figure). The problem is solved by using a Local Multi-Step Algorithm, which undergoes three subsequent stages corresponding, respectively, to computing the control points close to the corners of the patch, those along the boundaries and finally the control points in the interior. Since we want to obtain a watertight (i.e., C^0-continuous) connection between the spline patches, we additionally require that the control points of two B-spline surfaces along a shared boundary curve are equal.

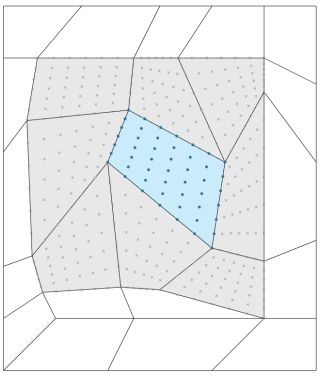

Figure 6. Area involved in the construction of a spline patch: For the patch indicated in light-blue and its control points marked by circles, the neighboring patches (colored gray) and their control points (marked with x-es) are used.

One may introduce additional continuity constraints to impose G^1 (i.e., tangent plane) continuity between abutting patches. At the moment, we do not deal with G^1 continuity constraints as they make it much more difficult to find a solution, and, even more so, a solution may not exist for all meshes.

3. Numerical Results

We present the results of the proposed method on real-wise objects representing, respectively, a bridge, a bell crank and a support.

For the bridge, the mesh obtained from TO consists of 28,182 triangles (14,069 vertices). The Voronoi tiling, described in Section 2.1, generates the partition in Figure 7 Left,

consisting of 202 tiles. The dual of the Voronoi triangulation is illustrated in the second figure from the left. The third figure shows the quadrilateral mesh (224 quadrilaterals) generated as discussed in Section 2.2. Finally, Figure 7 shows the spline surface obtained by least squares minimization, as discussed in Section 2.3. Each spline patch has degree 3 and the knot partition $\{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$ in both grid directions (the same setting is used in all the examples in this section). The algorithms were run on a laptop with `Intel Core i5-6300U` CPU at `2.40 GHz` processor (a capacity to run four threads in parallel) and `8 GB RAM`. The execution times are 263.3 s for the Voronoi tiling, 32.5 s for the macro quad patch partition and 46 min and 12 s for the computation of the spline patches. Compared to the computation of the spline patches, the computation times for forming the dual of the Voronoi tiles and for the quadrangulation are negligible.

For the bell crank, the outcome of the method is illustrated in Figure 8. In this case, the initial mesh consists of 45,122 triangles (22,519 vertices) and the execution times are 1429.1 s for the Voronoi tiling, 142.0 s for the macro quad patch partition and 1 h 53 min 15 s for the computation of the spline patches.

The outcome of the method is illustrated in Figure 9. In this case, the initial mesh consists of 22,812 triangles (11,398 vertices) and the execution times are 64.2 s for the Voronoi tiling, 15.3 s for the macro quad patch partition and 24 min and 1 s for the computation of the spline patches.

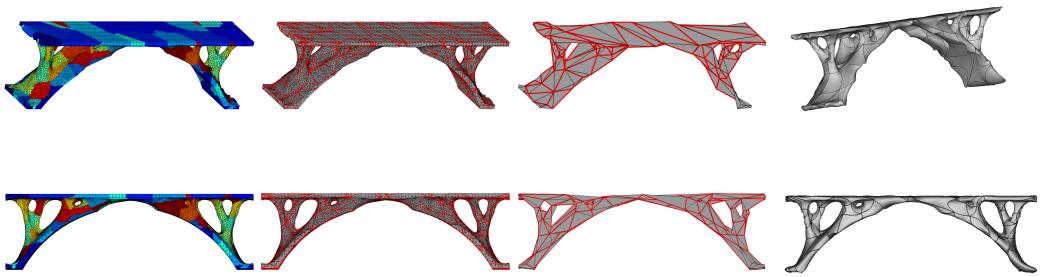

Figure 7. From **left** to **right**: TO mesh and augmented Voronoi tessellation where each tile is represented in a different color; TO mesh and macro triangles (the edges of the macro-triangles are drawn in red color); macro quadrangulation (the red segments represent the edges of the quadrilaterals); spline surface and boundary curves of the spline patches.

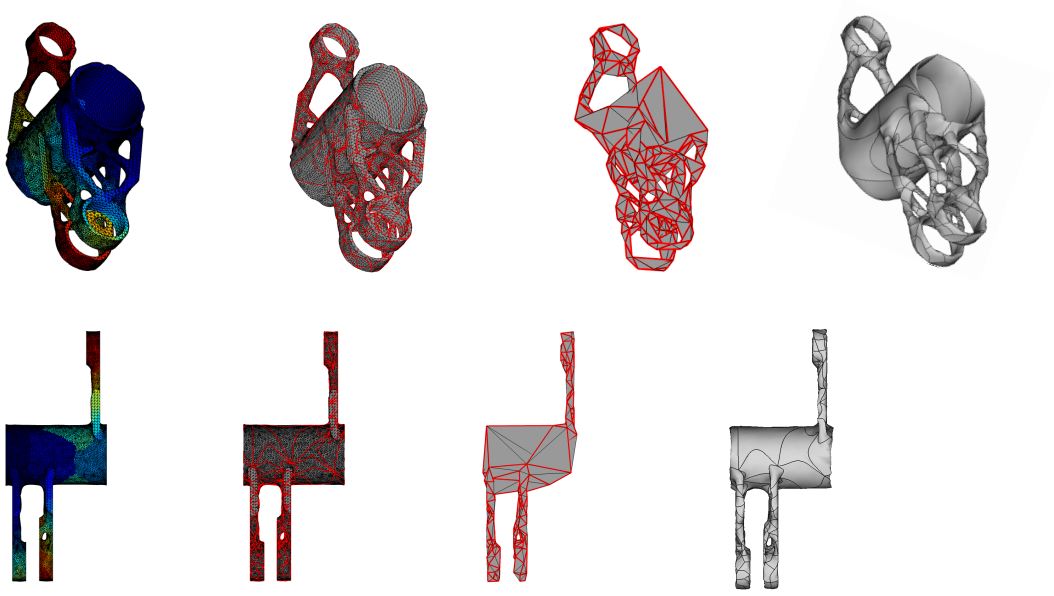

Figure 8. *Cont.*

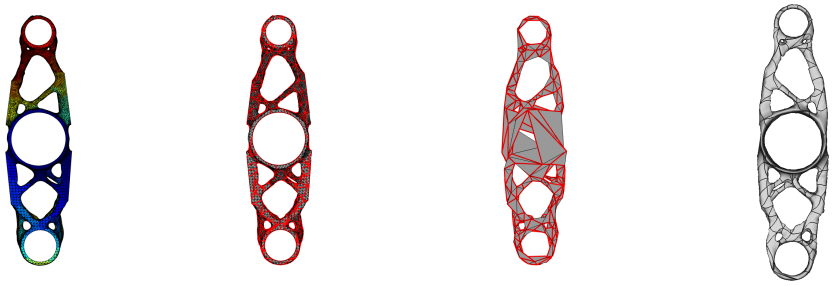

Figure 8. From **left** to **right**: TO mesh and augmented Voronoi tessellation where each tile is represented in a different color; TO mesh and macro triangles (the edges of the macro-triangles are drawn in red color); macro quadrangulation (the red segments represent the edges of the quadrilaterals); spline surface and boundary curves of the spline patches. The second and third line contain 3D images viewed from above.

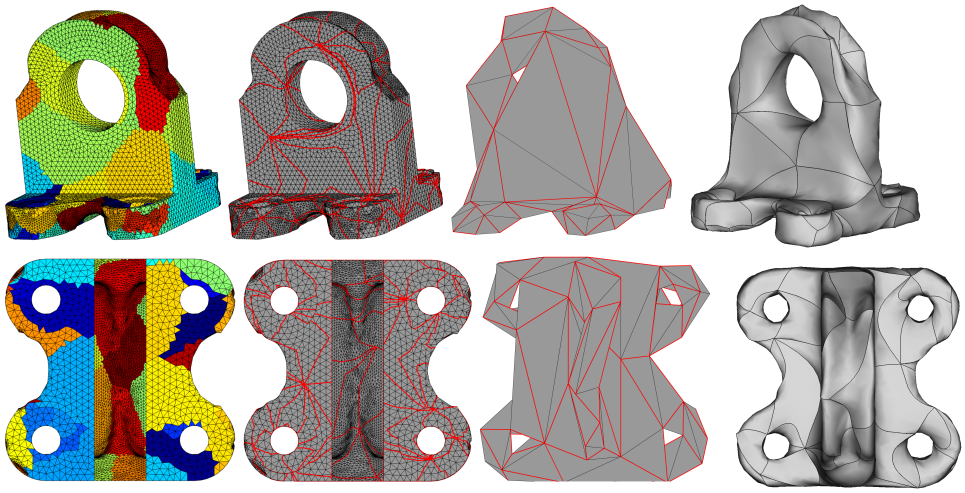

Figure 9. From **left** to **right**: TO mesh and augmented Voronoi tessellation where each tile is represented in a different color; TO mesh and macro triangles (the edges of the macro-triangles are drawn in red color); macro quadrangulation (the red segments represent the edges of the quadrilaterals); spline surface and boundary curves of the spline patches.

The graph in Figure 10 illustrates a curve that interpolates the computational times for the models tested in this study. It serves as an indication of the computational times one might expect for larger models (up to 100,000 triangles) using current hardware technology.

The examples show that the procedure is effective on bulky bodies, such as the support, hollow ones, such as the bridge, and bodies with both bulky and hollow parts, such as the bell crank. The procedure preserves the initial mesh's topology (number of holes). Moreover, the least squares approach used for computing the spline control points does not significantly alter the dimensions of holes and flat faces. One may also exactly enforce flat faces by flagging the spline patches that should be planar and setting all the spline control points to be on that plane.

The methodology provides smooth surfaces and reduces the body's complexity. This can be useful with complex parts: in this case, saving the triangulated model obtained through TO as a solid body can lead to large models that are difficult to handle. This is especially true if the part should be included in a CAD assembly where several parts are obtained from TO.

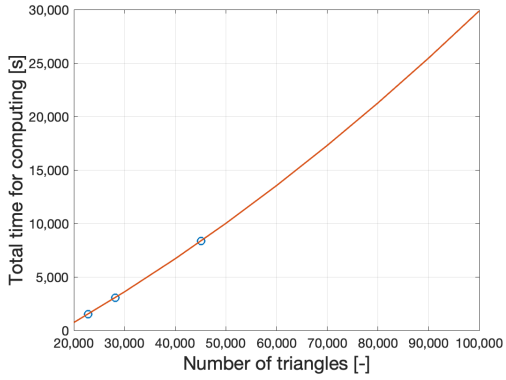

Figure 10. Trend of the total computational time with number of triangles.

One of the issues that arise after transitioning from the TO mesh to a spline model pertains to structural behavior. Various smoothing and filtering techniques can deform the original body, leading to a reduction in volume and, consequently, a decrease in strength. This effect is evident when performing finite element analysis (FEM) on both the original and the smoothed/filtered bodies. To conduct a more detailed examination of how our approach affects material distribution, we performed a mesh comparison. We utilized the CloudCompare software, version 2.14.alpha (https://www.danielgm.net/cc/ (accessed on 31 December 2024)) to overlay the TO mesh and the spline model and employ the distance tool to identify and analyze the differences between them. The reference model for this comparison is the original TO mesh, so all differences are measured against it. Figure 11 Left illustrates the TO mesh in green and the spline model in red. Visually, a good matching of the two models can be noticed. The graph in Figure 11 Right is obtained from a distance analysis between them. The length (maximum dimension) of the bell crank is around 190 mm, and the mean distance between the models is 0.098 mm.

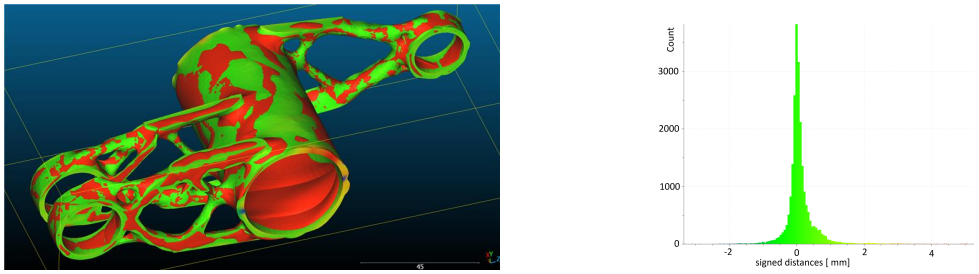

Figure 11. (**Left**): TO mesh (in green) and the spline model (in red) overlayed. (**Right**): Histogram of the distances between the two models, computed over 22,519 values.

4. Discussion

We have considered the problem of generating a polynomial patch layout from a mesh produced by a topology optimization process. Our study has focused on objects with complex shapes, particularly with a high topological genus, often resulting from topological optimization applied to real-world components used in the mechanical and aerospace industries. We have implemented and analyzed an approach based on two stages: the generation of a quadrilateral macro-mesh and the construction of a spline patch corresponding to each quadrilateral. The approach has proven viable for such complex objects and efficient in terms of computational time.

Topological optimization results in complex shapes that require additive manufacturing technologies for production. However, these AM techniques can be expensive, which limits the application of topology optimization to fields where reducing structural weights is crucial enough to justify the increased costs. The aeronautical industry stands to benefit

significantly from TO because, as suggested by the weight spiral law [36], a decrease in airframe weight leads to a magnified reduction in maximum take-off weight. Similarly, weight reduction is critical in space engineering since the costs of launching payloads into orbit are directly proportional to their mass. Usually, TO is applied to structural components where aerodynamic analysis is not carried out. However, if a Computational Fluid Dynamics (CFD) analysis is required, a surface mesh of the body is essential as the starting point for the surrounding volume meshing process. In this scenario, using a spline model offers greater flexibility for meshing than a mesh discrete model. Figure 12 demonstrates how the surface of the support and of the bridge can mesh with varying mesh sizes using the Gmsh software, version 4.13.1 (Christophe Geuzaine and Jean-Francois Remacle) [37], without needing to apply mesh decimation algorithms, which could negatively impact the body's features.

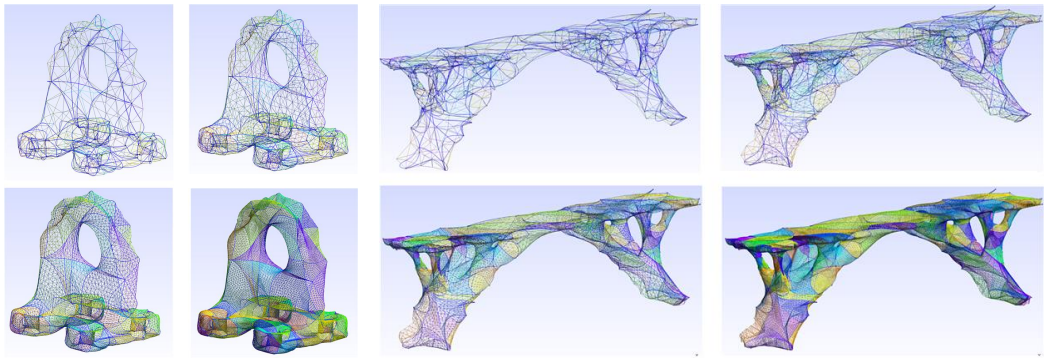

Figure 12. Support and bridge meshed with Gmsh with varying surface mesh sizes. For the support the number of surface mesh triangular elements in the four figures is, respectively, 936, 3744, 14,976, 59,904. For the bridge the number of surface mesh triangular elements in the four figures is, respectively, 1700, 6800, 27,200, 108,800.

The results in this paper represent a step toward the automatic generation of smooth spline surfaces that can be directly used in a CAD system and open up several future research directions. One is the generation of spline patches having boundaries aligned with the feature curves of the object to be generated. In the proposed method, such a goal can be pursued by suitably placing the initial sites of the Voronoi tessellation, and an automatic strategy to do so is one of our future topics of investigation. Moreover, CAD objects often contain flat faces and sharp or circular features, that should be maintained for functional requirements. The presented approach can exactly reproduce sharp features or flat faces if the corresponding edge or face is marked as sharp or flat; another topic of investigation is being able to incorporate these features automatically. Generating exact circles should also be a relatively simple task. Finally, we plan to continue our study to generate spline layouts with higher continuity. Typically, TO adds material in points where stress concentration can reduce the structural strength of that part: in those points, fillet features are obtained in the mesh, and C^1 continuity in the spline reconstruction can contribute to the reduction in stresses. So far G^1 (i.e., tangent plane) continuity has been attained in a few academic test cases but not in real-world industrial applications. In this respect, we aim to investigate whether improving the quadrilateral layout (e.g., reducing the number of vertices of high valence and suitably controlling their locations) may resolve the problem. We also plan to continue our study toward raising the continuity of the spline layout to G^2 (curvature continuity), which is desirable in aerospace and mechanical applications.

Author Contributions: Conceptualization, C.V.B. and A.C.; Methodology, C.V.B. and F.C.; Software, C.V.B. and F.C.; Validation, A.C. and F.C.; Resources, A.C.; Writing—original draft, C.V.B., A.C. and F.C. All authors have read and agreed to the published version of the manuscript.

Funding: The research has been supported by the Alma Idea initiative from University of Bologna, Italy.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The first author gratefully acknowledges support from INdAM-GNCS Gruppo Nazionale per il Calcolo Scientifico. The first author is also member of the Alma Mater research center on Applied Mathematics (AM^2).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Montalti, A.; Galiè, G.; Pignatelli, E.; Liverani, A. Enhancing surface roughness of material extrusion additive manufacturing components via an innovative ironing process. *Virtual Phys. Prototyp.* **2024**, *19*, e2401929. [CrossRef]
2. Bacciaglia, A.; Ceruti, A. Efficient toolpath planning for collaborative material extrusion machines. *Rapid Prototyp. J.* **2023**, *29*, 1814–1828. [CrossRef]
3. Mantovani, S.; Campo, G.A.; Ferrari, A. Additive manufacturing and topology optimization: A design strategy for a steering column mounting bracket considering overhang constraints. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 1703–1723. [CrossRef]
4. Rosso, S.; Uriati, F.; Grigolato, L.; Meneghello, R.; Concheri, G.; Savio, G. An Optimization Workflow in Design for Additive Manufacturing. *Appl. Sci.* **2021**, *11*, 2572. [CrossRef]
5. Bacciaglia, A.; Liverani, A.; Ceruti, A. Efficient part orientation algorithm for additive manufacturing in industrial applications. *Int. J. Adv. Manuf. Technol.* **2024**, *133*, 5443–5462. [CrossRef]
6. Bacciaglia, A.; Ceruti, A.; Liverani, A. Structural Analysis of Voxel-Based Lattices Using 1D Approach. *Print. Addit. Manuf.* **2022**, 9, 365–379. [CrossRef]
7. Bendsøe, M.P. Optimal shape design as a material distribution problem. *Struct. Optim.* **1989**, *1*, 193–202. [CrossRef]
8. Xie, Y.; Steven, G. Evolutionary structural optimization for dynamic problems. *Comput. Struct.* **1996**, *58*, 1067–1073. [CrossRef]
9. Li, Q.; Steven, G.; Xie, Y. A simple checkerboard suppression algorithm for evolutionary structural optimization. *Struct. Multidiscip. Optim.* **2001**, *22*, 230–239. [CrossRef]
10. Mortazavi, A. A novel binomial strategy for simultaneous topology and size optimization of truss structures. *Eng. Optim.* **2024**, 1–35. [CrossRef]
11. Mortazavi, A. A novel binomial-based fuzzy type-2 approach for topology and size optimization of skeletal structures. *Adv. Eng. Softw.* **2025**, *199*, 103819. [CrossRef]
12. Desbrun, M.; Meyer, M.; Schröder, P.; Barr, A.H. Implicit fairing of irregular meshes using diffusion and curvature flow. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '99, Los Angeles, CA, USA, 8–13 August 1999; ACM Press: New York, NY, USA, 1999; pp. 317–324. [CrossRef]
13. Sorkine, O. Laplacian Mesh Processing. *Eurograph. (State Art Rep.)* **2005**, *4*, 53–70. [CrossRef]
14. Belyaev, A.; Ohtake, Y. A comparison of mesh smoothing methods. In Proceedings of the Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics, Tel-Aviv, Israel, 12–14 February 2003.
15. Wei, M.; Huang, J.; Xie, X.; Liu, L.; Wang, J.; Qin, J. Mesh Denoising Guided by Patch Normal Co-Filtering via Kernel Low-Rank Recovery. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 2910–2926. [CrossRef] [PubMed]
16. Taubin, G. Curve and surface smoothing without shrinkage. In Proceedings of the IEEE International Conference on Computer Vision-ICCV-95, Cambridge, MA, USA, 20–23 June 1995; IEEE Computer Society Press: Piscataway, NJ, USA, 1995. [CrossRef]
17. Usai, F.; Livesu, M.; Puppo, E.; Tarini, M.; Scateni, R. Extraction of the Quad Layout of a Triangle Mesh Guided by Its Curve Skeleton. *ACM Trans. Graph.* **2016**, *35*, 6. [CrossRef]
18. Eck, M.; DeRose, T.; Duchamp, T.; Hoppe, H.; Lounsbery, M.; Stuetzle, W. Multiresolution analysis of arbitrary meshes. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '95, Los Angeles, CA, USA, 6–11 August 1995; ACM Press: New York, NY, USA, 1995; pp. 173–180. [CrossRef]
19. Eck, M.; Hoppe, H. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '96, New York, NY, USA, 4–9 August 1996; pp. 325–334. [CrossRef]
20. Mosbach, D.; Schladitz, K.; Hamann, B.; Hagen, H. A Local Approach for Computing Smooth B-Spline Surfaces for Arbitrary Quadrilateral Base Meshes. *J. Comput. Inf. Sci. Eng.* **2022**, *22*, 11003. [CrossRef]
21. Cuillière, J.C.; Francois, V.; Drouet, J.M. Towards the Integration of Topology Optimization into the CAD Process. *Comput.-Aided Des. Appl.* **2013**, *11*, 120–140. [CrossRef]

22. Chen, C.Y.; Cheng, K.Y. A direction-oriented sharpness dependent filter for 3D polygon meshes. *Comput. Graph.* **2008**, *32*, 129–140. [CrossRef]
23. Cuillière, J.C.; François, V.; Nana, A. Automatic construction of structural CAD models from 3D topology optimization. *Comput.-Aided Des. Appl.* **2017**, *15*, 107–121. [CrossRef]
24. Larsen, S.; Jensen, C.G. Converting Topology Optimization Results into Parametric CAD Models. *Comput.-Aided Des. Appl.* **2009**, *6*, 407–418. [CrossRef]
25. Koguchi, A.; Kikuchi, N. A surface reconstruction algorithm for topology optimization. *Eng. Comput.* **2006**, *22*, 1–10. [CrossRef]
26. Yin, G.; Xiao, X.; Cirak, F. Topologically robust CAD model generation for structural optimisation. *Comput. Methods Appl. Mech. Eng.* **2020**, *369*, 113102. [CrossRef]
27. Park, J.M.; Lee, B.C.; Chae, S.W.; Kwon, K.Y. Surface reconstruction from FE mesh model. *J. Comput. Des. Eng.* **2018**, *6*, 197–208. [CrossRef]
28. Cuillière, J.C.; Francois, V. Integration of CAD, FEA and Topology Optimization through a Unified Topological Model. *Comput.-Aided Des. Appl.* **2014**, *11*, 493–508. [CrossRef]
29. Nana, A.; Cuillière, J.C.; Francois, V. Automatic reconstruction of beam structures from 3D topology optimization results. *Comput. Struct.* **2017**, *189*, 62–82. [CrossRef]
30. Amroune, A.; Cuillière, J.C.; François, V. Automated Lofting-Based Reconstruction of CAD Models from 3D Topology Optimization Results. *Comput.-Aided Des.* **2022**, *145*, 103183. [CrossRef]
31. Bacciaglia, A.; Ceruti, A.; Liverani, A. Surface smoothing for topological optimized 3D models. *Struct. Multidiscip. Optim.* **2021**, *64*, 3453–3472. [CrossRef]
32. Lin, C.; Chou, Y. Automated structural optimization system for integrated topology and shape optimization. *J. Chin. Inst. Eng.* **2008**, *31*, 745–756. [CrossRef]
33. Chou, Y.H.; Lin, C.Y. Improved image interpreting and modeling technique for automated structural optimization system. *Struct. Multidiscip. Optim.* **2009**, *40*, 215–226. [CrossRef]
34. Tang, P.S.; Chang, K.H. Integration of topology and shape optimization for design of structural components. *Struct. Multidiscip. Optim.* **2001**, *22*, 65–82. [CrossRef]
35. Piegl, L.; Tiller, W. *The NURBS Book*, 2nd ed.; Springer: New York, NY, USA, 1996.
36. Raymer, D.P. *Aircraft Design: A Conceptual Approach*, 4th ed.; AIAA Education Series; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2006.
37. Geuzaine, C.; Remacle, J.F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.* **2009**, *79*, 1309–1331. [CrossRef]