CSMA 2026
17ème Colloque National en Calcul des Structures
18-22 Mai 2026, Presqu'île de Giens (Var)

An Industrially Robust Quadrilateral Mesher for CAD Models

J.-F. Remacle[1], C. Geuzaine[2]

[1] *Université catholique de Louvain, Institut de Mécanique, des Matériaux et du Génie Civil, Avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgique*
[2] *Université de Liège, Institut Montefiore, Allée de la Découverte 10, 4000 Liège, Belgique*

Résumé — We introduce a new quadrilateral meshing algorithm integrated in the open-source mesh generator Gmsh. The method targets industrially robust meshing of complex CAD models with many trimmed patches. It combines : (i) a pattern-based strategy producing high-quality meshes when patch topology matches predefined templates ; (ii) a global quantization procedure enforcing compatible subdivisions across adjacent patches ; and (iii) an improved packing of parallelograms for general surfaces. A Winslow-type smoother on CAD geometries further enhances element quality, yielding accurate and simulation- ready quad meshes.
Mots clés — Quad Meshing, Gmsh, Winslow, Quantization.

1 Introduction

Quadrilateral meshes are widely used in industrial simulations due to their superior numerical properties, yet generating high-quality quad meshes on arbitrary CAD models remains difficult. Despite numerous approaches over the past decades—including paving [1], advancing fronts [1], cross-field parametrizations [2, 3], medial-axis methods [4], and pattern-based strategies [5]—robust, fully automatic meshing of large CAD assemblies is still an open problem. The main challenges stem from models composed of many trimmed patches of varying quality and from geometric complexities such as small features and anisotropy.

We introduce a new quadrilateral meshing algorithm integrated into Gmsh [6], designed to meet industrial robustness requirements. It combines : (i) a global quantization scheme enforcing compatible, even-valued subdivisions, (ii) a pattern-based meshing strategy, (iii) the use of packing-of-parallelograms algorithm for generating quad-dominant meshes on general surfaces, and (iv) a Winslow-type smoother on CAD geometries. Together, these components produce accurate and simulation-grade quad meshes with minimal user intervention.

2 Global Quantization

In Gmsh's meshing pipeline ("mesh flow"), we follow a surface-to-volume strategy : boundary curves are discretized first, surfaces are then meshed using the 1D discretization of the curves, and volumes are meshed using surfaces triangulations/quadrangulations. Curves are meshed independently according to the user-prescribed size field. For a surface to be meshed exclusively with quadrilaterals, the total number of subdivisions assigned to the curves bounding that surface must be even. While enforcing an even number of segments on every individual curve is a sufficient condition, it typically induces substantial changes to the 1D mesh. We therefore propose an alternative parity-enforcement strategy that achieves the required evenness while perturbing the boundary discretization as little as possible.

2.1 The solver

We consider a CAD model with S surfaces and C curves. We define the *face to curve* incidence matrix as

$$A \in \{0,1\}^{S\times C}, \qquad A_{ij} = \begin{cases} 1 & \text{if curve } j \text{ lies on the boundary of surface } i, \\ 0 & \text{otherwise.} \end{cases}$$

Each curve j has an initial/optimal number of subdivisions $n_j \in \mathbb{N}$. We allow a modification

$$n'_j = n_j + x_j, \qquad x_j \in \{-1, 0, +1\},$$

and we want every surface to have an *even* total number of subdivisions :

$$\sum_{j=1}^{C} A_{ij} n'_j \quad \text{is even for all } i = 1, \dots, S.$$

Since only parity matters, we take $x_j \in \{0, 1\}$, meaning we either flip or do not flip the parity of curve j. Define for each surface

$$b_i := \left(\sum_{j=1}^{C} A_{ij} n_j \right) (\mathrm{mod}\, 2).$$

The parity constraints become

$$\sum_{j=1}^{C} A_{ij} x_j \equiv b_i \pmod 2.$$

Then we have to solve the linear system

$$Ax = b \qquad \text{over } \mathbb{F}_2$$

where $\mathbb{F}_2$ is the finite field with two elements, $\{0, 1\}$, equipped with addition and multiplication modulo 2. We solve $Ax = b$ by Gaussian elimination over $\mathbb{F}_2$. It is quite straightforward to show, using simple arguments based on the Euler–Poincaré formula, that $S < C$, so the system above is over-determined. We therefore perform an incomplete Gaussian elimination, which allows us to obtain an initial/valid solution $x^{(0)}$ to the problem. From a practical point of view, short curves with a small number of subdivisions should not be given priority when adjusting their parity. Thus, modifying a curve with large n_j should be *cheaper* than modifying a curve with small n_j. We introduce positive weights $w_j > 0$, decreasing functions of n_j, for instance $w_j = \frac{1}{n_j}$. We thus eventually want to solve :

$$\min_{x \in \{0,1\}^C} \quad \sum_{j=1}^{C} w_j x_j \qquad \text{subject to} \qquad Ax = b \quad \text{in} \quad \mathbb{F}_2.$$

The problem being over-determined, there exists a null space in A of size $d = C - S$. The nullspace basis vectors $v^{(1)}, \dots, v^{(d)}$ are obtained directly from the reduced row-echelon form of A over $\mathbb{F}_2$: after performing Gaussian elimination modulo 2, each free column of the matrix yields one basis vector by setting the corresponding free variable to 1, the others to 0, and back-substituting to determine the pivot variables. If $v^{(1)}, \dots, v^{(d)} \in \{0, 1\}^C$ denotes a basis of the null space of A over $\mathbb{F}_2$, that is $Av^{(k)} = 0$ in $\mathbb{F}_2$ for all $k = 1, \dots, d$, the general parametrization of solutions writes

$$x(\alpha) = x^{(0)} + \sum_{k=1}^{d} \alpha_k v^{(k)},$$

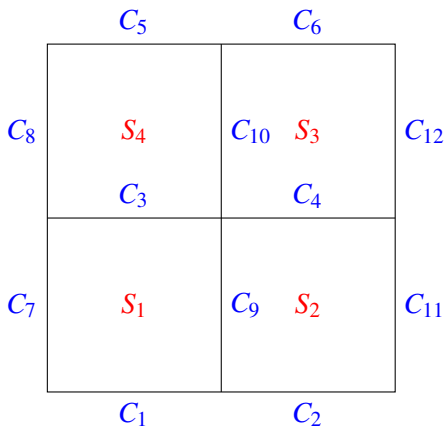
where we choose $\alpha \in \{0, 1\}^d$ minimizing the cost

$$J(\alpha) = \sum_{j=1}^{C} w_j x_j(\alpha).$$

If d is small, all 2^d choices of α can be enumerated. If d is larger, the problem is a small binary linear program and can be solved using simple greedy heuristics. Once x is chosen, we set

$$n'_j = n_j + \delta_j, \qquad \delta_j = \begin{cases} +1, & \text{if } x_j = 1, \\ 0, & \text{if } x_j = 0. \end{cases}$$

Optionally, if n_j must not decrease below 1 (i.e. if we do not allow curves with no mesh, which could e.g. be removed from the model), the sign of the correction ($+1$ or -1) can be chosen without affecting parity, since the parity constraint depends only on x_j mod 2.



$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

FIGURE 1 – A simple geometry with 4 squares.

2.2 Example

As an example, consider the simple geometry of Figure 1. Assuming that the initial edge parities are

$$n_{\text{init}} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}^T,$$

we obtain

$$A\,n_{\text{init}} = \begin{pmatrix} 1 & 3 & 1 & 2 \end{pmatrix}^T \;\to\; A\,n_{\text{init}} \,(\text{mod } 2) = \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix}^T = b$$

and surfaces S_1, S_2 and S_3 are initially odd. We form the augmented matrix $[A \mid b]$ and perform Gaussian elimination over $\mathbb{F}_2$. In reduced row-echelon form we obtain

$$[A \mid b] \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The pivot columns are $1,2,3,4$, so columns $5,\ldots,12$ are free. Setting all free variables to zero (which means we do not change parities of surfaces 5 to 12),

$$x_5 = x_6 = x_7 = x_8 = x_9 = x_{10} = x_{11} = x_{12} = 0,$$

the reduced system reads

$$x_1 = 1, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 1.$$

Thus one particular solution is

$$x^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad Ax^{(0)} \equiv b \pmod 2,$$

which means that, if parities of curves C_1 and C_4 are changed, the 4 global parity constraints are fulfilled. To describe the null space we solve $Ax = 0$ over $\mathbb{F}_2$. The reduced row-echelon form of A alone is

$$A \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Again, columns $1,2,3,4$ are pivots and $5,\ldots,12$ are free. Writing

$$x_5 = s_1,\ x_6 = s_2,\ x_7 = s_3,\ x_8 = s_4,\ x_9 = s_5,\ x_{10} = s_6,\ x_{11} = s_7,\ x_{12} = s_8,$$

the pivot variables are expressed as

$$\begin{aligned} x_1 &= s_1 + s_3 + s_4 + s_5 + s_6, \\ x_2 &= s_2 + s_5 + s_6 + s_7 + s_8, \\ x_3 &= s_1 + s_4 + s_6, \\ x_4 &= s_2 + s_6 + s_8, \end{aligned}$$

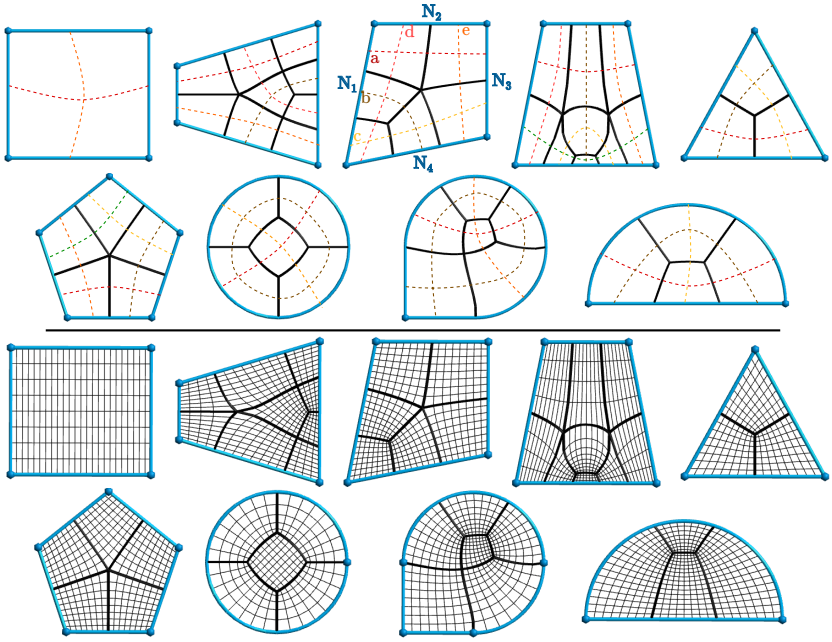


FIGURE 2 – Pattern-based topological quad meshing. The patterns are defined by coarse quadrilateral meshes (top). The topological chords (dashed lines) can be subdivided independently to build refined quadrilateral meshes (bottom).

with all additions taken modulo 2. Hence every vector in the null space can be written as

$$x(s_1,\ldots,s_8) = \begin{pmatrix} s_1+s_3+s_4+s_5+s_6 \\ s_2+s_5+s_6+s_7+s_8 \\ s_1+s_4+s_6 \\ s_2+s_6+s_8 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{pmatrix}, \qquad s_k \in \{0,1\}.$$

A basis of the null space is obtained by turning on one parameter s_k at a time. For example, for $s_1 = 1$ and all other $s_k = 0$, we get

$$v^{(1)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and similarly for $s_2,\ldots,s_8$. These eight vectors $\{v^{(1)},\ldots,v^{(8)}\}$ form a basis of $\ker(A)$ over $\mathbb{F}_2$.

3 Pattern based quadrangulation

Industrial CAD models generally contain a large number of faces—sometimes thousands. A significant number of these faces have the topology of a disk (they are contractible — all cycles can be reduced to a point, they are connected, have a single boundary, and an Euler characteristic equal to 1). For those simple faces, it is often possible to construct a high-quality quadrilateral mesh by applying suitable patterns.

All the patterns we consider are convex (see Figure 2). The boundary of each pattern may contain convex corners, i.e., abrupt changes in the direction of the unique contour of the patch. Our patterns are

essentially distinguished by the number of such corners : the first four patterns in Figure 2 have four corners, the fifth has three, the sixth has five, the seventh has none, and the eighth and ninth have one and two, respectively.

Between two successive corners, the one-dimensional mesh contains a certain number of subdivisions. If a patch has m corners, we denote by $N_1, \ldots, N_m$ the number of subdivisions between corners. It should be noted that the number of corners does not necessarily correspond to the number of model edges describing the boundary of the patch, although the parity constraint is still satisfied :

$$\sum_{i=1}^{m} N_i \equiv 0 \pmod 2.$$

Each pattern is associated with a corresponding quad layout. A quad layout is a partition of a surface into a small number of topological quadrilateral patches. The layout captures the global structure of a quadrilateral mesh without specifying the internal grid resolution of each patch.

The dual of the quad layout is composed of a set of c chords, shown as dashed red lines in Figure 2. Each chord is associated with an unknown variable representing the number of one-dimensional subdivisions along that chord.

Let us go back to our CAD face with m corners and subdivisions $(N_1, ..., N_m)$ in between corners. We check if there is a matching with one quad pattern with m corners in our pre-computed pattern list. For each pattern, we define a set of c integer unknowns $(s_1, ..., s_c)$ that represent the number of subdivisions of the c chords. There is a quadrilateral mesh if it is possible to find strictly positive chord subdivisions $s_j > 0, j \in [1, c]$ such that on each side, the sum of the subdivisions of the chords included in the side is equal to the side number of edges :

$$\forall i \in [1, m], \quad \sum_{j \in [1,c]} w_{ij}\, s_j = N_i \tag{1}$$

where w_{ij} is the number of times the chord j in inside the side i. In practice, w_{ij} is equal to 0 (the chord is not on the side), 1 (the chord goes through the pattern) or 2 (the chord starts and finishes on the side, turning inside).

After performing the global quantization of the edges, we inspect each CAD patch and verify that it has the correct topology (no holes) and that it contains only convex corners, with no more than five of them. We then check whether the subdivisions of the edges lead to positive subdivisions of the chords. If all these conditions are satisfied, we apply the pattern and smooth the mesh using an extension of the method proposed in [7] to curved surfaces, briefly described in Section §5. We then assess whether the resulting mesh is an improvement over the one automatically generated by the *packing of parallelograms* algorithm, which is briefly described in Section §4.

4 Packing of parallelograms

The *packing of parallelograms* (PoP) algorithm [8] is a fully automatic procedure for generating high-quality quadrilateral meshes on general surfaces. In a nutshell, the PoP algorithm creates a quadrilateral mesh by sampling points in an anisotropic metric aligned with a cross field, interpreting the resulting packing as a tiling by parallelograms, and converting this structure into a high-quality quadrilateral mesh.

The algorithm proceeds in several steps. First, a smooth cross field is computed on the surface, which prescribes the local orientation of the quadrilateral elements. This orientation field is combined with the target mesh size to define an anisotropic metric in which ideal quadrilaterals correspond to nearly regular parallelograms.

A point-sampling strategy is then applied in this metric space to produce a quasi-uniform packing of points. Although this packing is hexagonal in Euclidean space, it becomes, by construction, equivalent to a tiling by parallelograms in the anisotropic metric. A quadrilateral mesh is then built by constructing a regular adjacency graph on the packed points and extracting cycles of four edges to define the quadrilateral faces.

Finally, several optimization steps improve the quality of the mesh by enforcing orthogonality, regularizing element sizes, and ensuring compatibility with geometric boundaries.

Since 2014, we have continuously improved the algorithm by computing cross fields through the solution of the Ginzburg–Landau equations, by enhancing the initial point-generation strategy, and by fixing numerous issues present in the original implementation.

5 Winslow smoothing for curved surfaces

5.1 Triangles in the plane

Let us first recall the principles of the approach proposed in [7]. Consider the task of untangling a mesh of triangles composed of n nodes. The goal is to optimize the position of the nodes so that each triangle is correctly oriented and exhibits low distortion with relation to an ideal triangular shape. However, distortion cannot be completely eliminated in practice. One can only hope for a mapping that preserves angles (conformal) or areas (authalic), but not both at the same time. What is proposed in [7] is to achieve a compromise between the two.

Let $\mathcal{T}$ be a triangulation of a domain Ω of the plane (the principle trivially extends to tetrahedrizations in space). Each triangle $t = (\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ of $\mathcal{T}$ is associated with a triangle $(\mathbf{X}_c, \mathbf{X}_b, \mathbf{X}_c)$ of *ideal shape*. This idea is quite similar to that used in anisotropic mesh adaptation, where a metric field is used to define angles and areas and where the ideal element is an equilateral or unit triangle with respect to the given metric.

The jacobian matrix $\mathbf{J} = \frac{d\mathbf{x}}{d\mathbf{X}}$ can easily be computed as

$$\mathbf{J} = (\mathbf{X}_b - \mathbf{X}_a, \mathbf{X}_c - \mathbf{X}_a)^{-1} (\mathbf{x}_b - \mathbf{x}_a, \mathbf{x}_c - \mathbf{x}_a),$$

with determinant

$$J := \det(\mathbf{J}) = \frac{(\mathbf{x}_b - \mathbf{x}_a) \times (\mathbf{x}_c - \mathbf{x}_a) \cdot \mathbf{e}_z}{(\mathbf{X}_b - \mathbf{X}_a) \times (\mathbf{X}_c - \mathbf{X}_a) \cdot \mathbf{e}_z}.$$

In [7], authors propose to minimize

$$\lim_{\varepsilon \to 0} \sum_{t \in \mathcal{T}} f_\varepsilon(\mathbf{J}_t, \varepsilon) + \lambda g_\varepsilon(\mathbf{J}_t, \varepsilon) \tag{2}$$

where

$$f_\varepsilon(\mathbf{J}) = \frac{\mathrm{tr}(\mathbf{J}^T\mathbf{J})}{\chi(J,\varepsilon)} \text{ and } g_\varepsilon(\mathbf{J}) = \frac{J^2+1}{\chi(J,\varepsilon)}.$$

Here, $\chi(x,\varepsilon) = \frac{x+\sqrt{\varepsilon^2+x^2}}{2}$ is a regularization function that tends to the ramp function when $\varepsilon \to 0$.

The term f_ε is minimal when the mapping preserve the angles of the triangles, while minimizing g_ε amounts to preserving their area. Parameter $\lambda > 0$ controls the balance between the two and thus controls the tradeoff. To minimize Eq. (2), we rely on the same quasi-Newton scheme proposed in [7]. In particular, we do not need to compute the Hessian of the function but only the derivatives of J with respect to coordinates $\mathbf{x}_{a,b,c}$:

$$\frac{\partial J}{\partial \mathbf{x}_{a,b,c}} = \frac{(\mathbf{x}_{c,a,b} - \mathbf{x}_{b,c,a}) \times \mathbf{e}_z}{(\mathbf{X}_b - \mathbf{X}_a) \times (\mathbf{X}_c - \mathbf{X}_a) \cdot \mathbf{e}_z}. \tag{3}$$

One of the advantages of this approach over [9], for example, is the ease with which the gradient of f_ε can be calculated. Another advantage of using the regularization proposed in [7] is that f_ε is convex so the solution that is found is unique and a simple LBFGS allows to converge in every case.

5.2 Adapting the untangler for non planar quad-dominant meshes

We start from a mesh generated either from a pattern or from the unstructured algorithm. We then convert this mesh into a triangular mesh in which each quad is replaced by four triangles – a double covering of the quad, which is essential to avoid producing non-convex quads. One possible way to account for the curved nature of CAD surfaces would be to adapt the methodology so as to operate in the parametric plane of the surfaces. We experimented with this option with mixed success : the parameterizations of certain surfaces, such as spheres or some surfaces of revolution, are singular in CAD

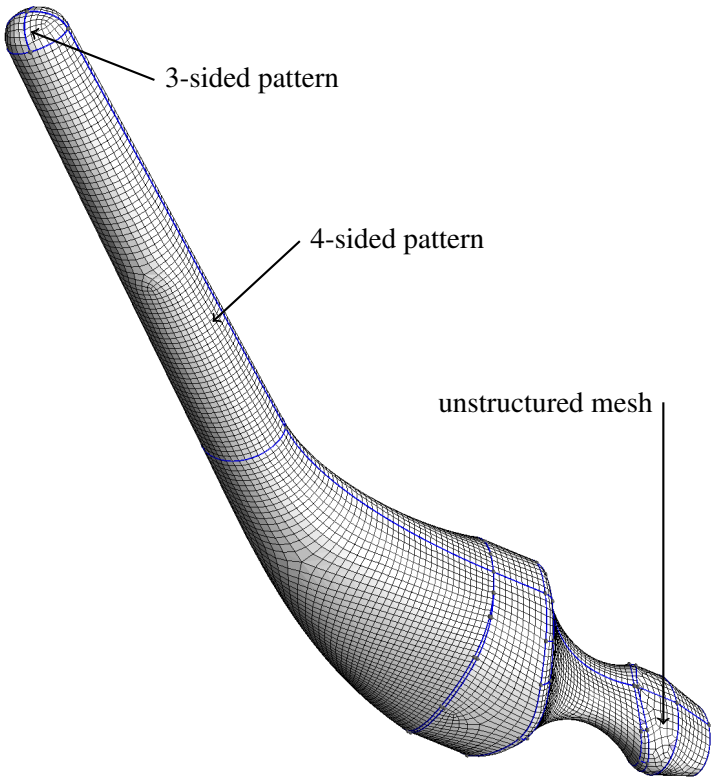


FIGURE 3 – Hip prothesis.

systems, and these distorted parameterizations cause the Winslow algorithm either to fail to converge or to produce poor-quality solutions. Indeed, a mesh may be valid in the parametric space but invalid in the physical space if the edges in the parametric domain are far from being geodesics [10].

The solution we propose here is very pragmatic. We choose a random node on the face to be smoothed. Starting from this node, we traverse its neighboring nodes using a breadth-first search. The traversal continues as long as the set of visited nodes remains sufficiently coplanar. We then project the elements of the resulting patch onto the average plane and apply the standard smoothing technique in this plane. The nodes are subsequently projected back onto the surface. This procedure is repeated until the positions of the smoothed mesh nodes stabilize.

6 Examples

We have used the new pipeline on many geometries; we expect that this pipeline will replace all of Gmsh's quad meshers in Gmsh 5.0. Figure 3 shows a very simple example of a hip prosthesis with $C = 164$ curves and $S = 52$ surfaces. We have generated 10164 quads and 26 triangles. A number of 18 parity flips were necessary to obtain a global even quantization. Among the 52 surfaces, 43 were meshed using patterns which is not unusual in complex CAD models. More surfaces could have been meshed with patterns if more 4-sided patterns were implemented that allow for steeper mesh size transitions. A total time of 9 seconds was necessary to generate the mesh, which includes the 1D quantization, the generation of an initial triangular mesh, the computation of 52 cross fields, the generation of 52 unstructured quad meshes and their smoothing, the generation of 43 quad patterns and their smoothing as well as IO's.

7 Conclusions

The development of robust quadrilateral meshing algorithms is a long-term effort. The method presented here represents the fifth major evolution of Gmsh's quad meshing pipeline, following Blossom

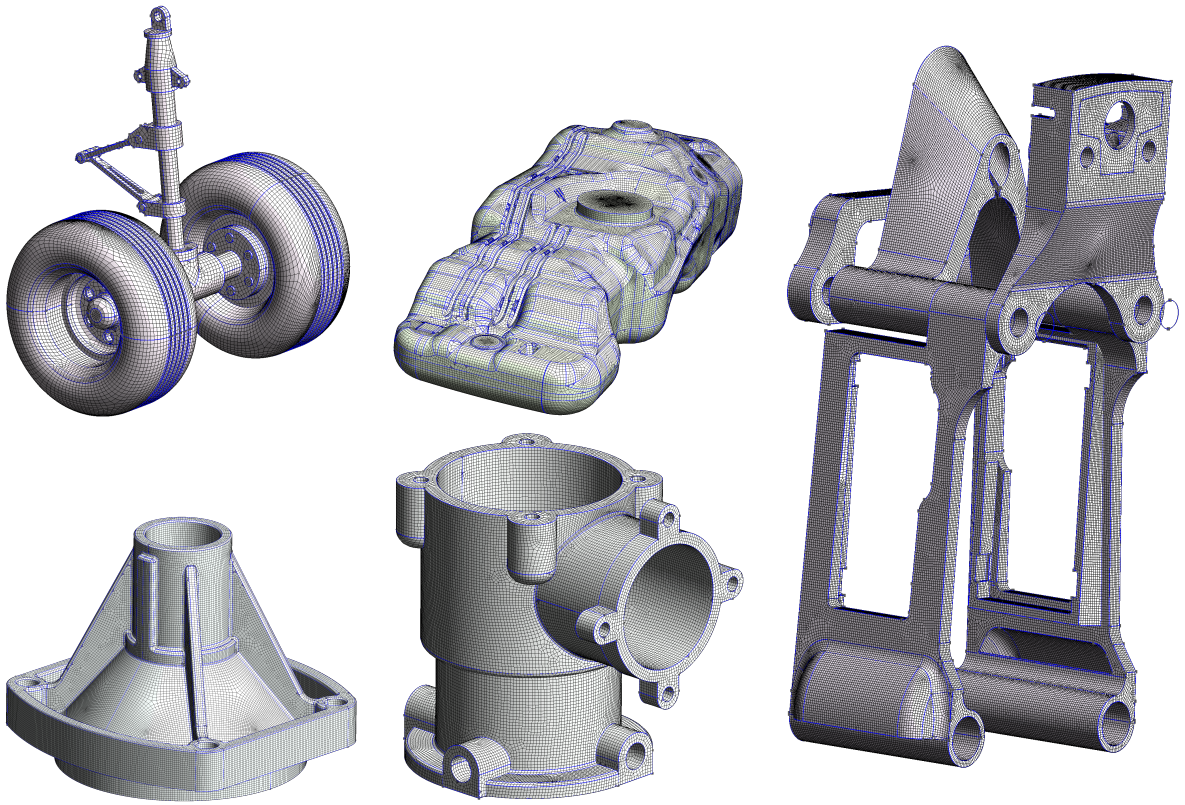


FIGURE 4 – High-quality meshes generated on industrial CAD models.

Quad [11], Frontal Quad [12], Packing of Parallelograms [13], and Quasi-Structured [14]. This new pipeline will become the default quadrilateral mesher in Gmsh 5.0, which is scheduled for release in 2026.

Références

[1] J. Z. Zhu, O. C. Zienkiewicz, E. Hinton, and J. Wu. A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(6) :849–866, 1991.

[2] N. Ray, W.-C. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Transactions on Graphics (TOG)*, 25(4) :1460–1485, 2006.

[3] D. Bommes, H. Zimmer, and L. Kobbelt. Mixed-integer quadrangulation. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09. ACM, 2009.

[4] T. K. H. Tam and C. G. Armstrong. 2D finite-element mesh generation by medial axis subdivision. *Advances in Engineering Software and Workstations*, 13 :313–324, 1991.

[5] K. Takayama, D. Panozzo, and O. Sorkine-Hornung. Pattern-based quadrangulation for n-sided patches. In *Computer Graphics Forum*, volume 33, pages 177–184. Wiley Online Library, 2014.

[6] C. Geuzaine and J.-F. 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, 2009.

[7] V. Garanzha, I. Kaporin, L. Kudryavtseva, F. Protais, N. Ray, and D. Sokolov. Foldover-free maps in 50 lines of code. *ACM Transactions on Graphics (TOG)*, 40(4) :1–16, 2021.

[8] T. Carrier-Baudouin, J.-F. Remacle, E. Marchandise, F. Henrotte, and C. Geuzaine. A frontal approach to hex-dominant mesh generation. *Advanced Modeling and Simulation in Engineering Sciences*, 1(1) :8, 2014.

[9] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254 :8–26, 2013.

[10] J.-F. Remacle and C. Geuzaine. Gmsh's approach to robust mesh generation of surfaces with irregular parametrizations. In *Mesh Generation and Adaptation : Cutting-Edge Techniques*, pages 95–112. Springer, 2022.

[11] J-F Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzaine. Blossom-quad : A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. *International journal for numerical methods in engineering*, 89(9) :1102–1119, 2012.

[12] J.-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine, and T. Mouton. A frontal delaunay quad mesh generator using the L^∞ norm. *International Journal for Numerical Methods in Engineering*, 94(5) :494–512, 2013.

[13] T. Carrier-Baudouin, J.-F. Remacle, E. Marchandise, F. Henrotte, and C. Geuzaine. A frontal approach to hex-dominant mesh generation. *Advanced Modeling and Simulation in Engineering Sciences*, 1(1) :8, 2014.

[14] M. Reberol, C. Georgiadis, and J.-F. Remacle. Quasi-structured quadrilateral meshing in Gmsh–a robust pipeline for complex CAD models. *arXiv preprint arXiv :2103.04652*, 2021.