Unstructured Quad Meshing

Jean-François Remacle[1] and Christophe Geuzaine[2]

[1] Université catholique de Louvain (UCLouvain)

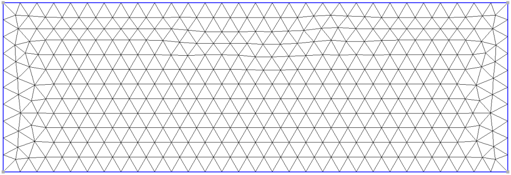
[2] Université de Liège (ULiege)

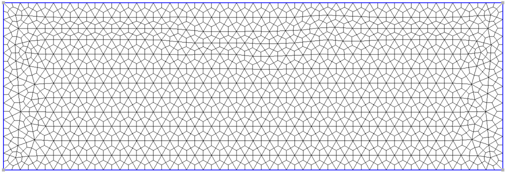
http://www.gmsh.info

October 7, 2024

Split Triangles – Full Quad

One triangle is divided in three quads, 20 lines of code, problem solved ?





Matching

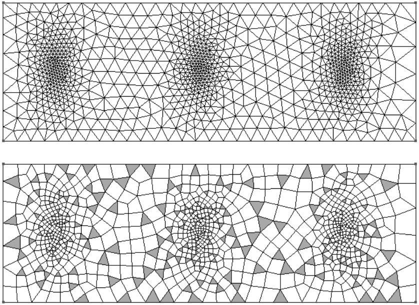
A quad q and its the four internal angles $\alpha_k, \quad k = 1, 2, 3, 4$. We define the quality (q) of q as:

$$(q) = \max\left(1 - \frac{2}{\pi}\max_k\left(\left|\frac{\pi}{2} - \alpha_k\right|\right), 0\right). \tag{3}$$

Greedy *quad-dominant* algorithm (Frey & Borouchaki, *Adaptive triangular–quadrilateral mesh generation*, IJNME, 1998).

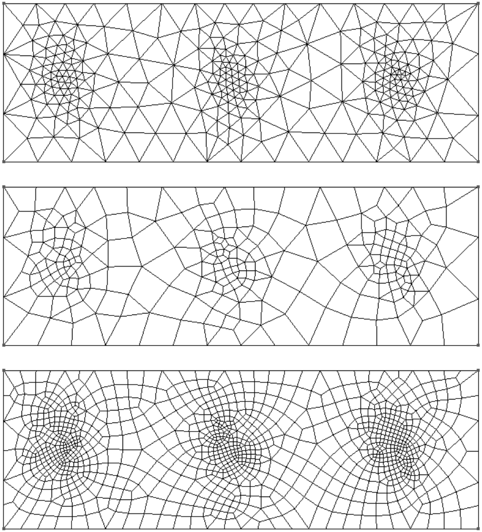
Rectangular domain of size 1×3 and a mesh size field defined by

$$h(x, y) = 0.1 + 0.08\sin(3x)\cos(6y).$$

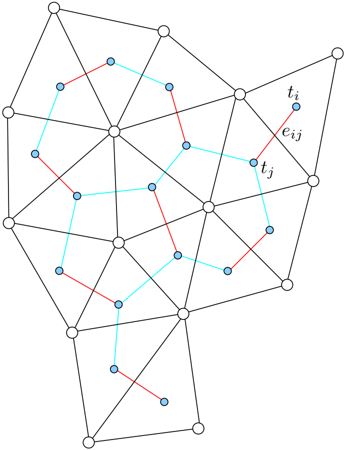


Full Quad

$2h(x, y)$ – Match – Split.



Perfect Matching



A mesh (in black) and its graph (in cyan and red). The set of graph edges colored in red forms a perfect matching.

Perfect Matching

In 1965, Edmonds (Edmonds, Jack. *Paths, trees, and flowers*. Can. J. Math., 1965) invented the *Blossom algorithm* that solves the problem of optimum perfect matching in polynomial time. A straightforward implementation of Edmonds's algorithm requires $\mathcal{O}(\#V^2E)$ operations.

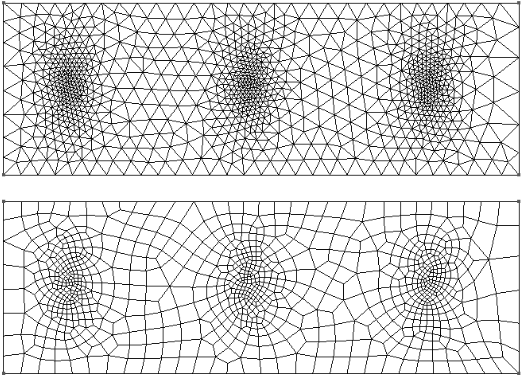
Since then, the worst-case complexity of the Blossom algorithm has been steadily improving. The current best known result is

$$\mathcal{O}(\#V(\#E + \log \#V))$$

Gmsh use the Blossom IV code of Cook and Rohe [1], which has been considered for several years as the fastest available

[1] Computer code available at http://www2.isye.gatech.edu/~wcook/blossom4/.

Perfect Matching



Try in Gmsh...

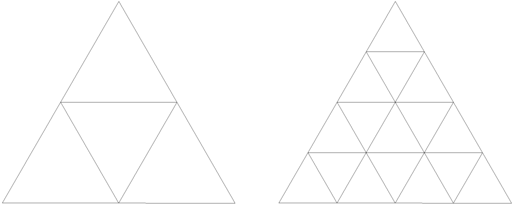
Existence of Perfect Matchings

$$n_t = 2(n_v - 1) - n_h.$$

An even nr. of triangles requires an even nr. of points on the boundary.

Even if n_t is even, *there is in general no guarantee that even one single perfect matching exists in a given graph.*

Tutte's theorem : A graph $G = (V, E)$ has no perfect matching if and only if there is a set $S \subseteq V$ whose removal results in more odd-sized components than the cardinality n_S of S, i.e., the number of elements in S (Pemmaraju S. and Skiena S, *Computational Discrete Mathematics*', 2003).



Planar Graphs

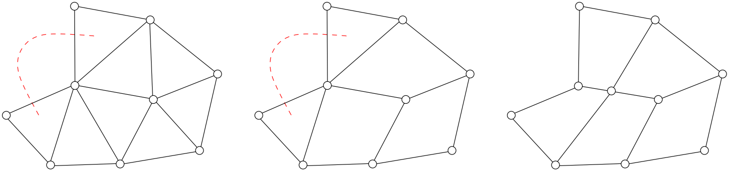
2D meshes are planar graphs. Gmsh only generates meshes in the parameter plane.

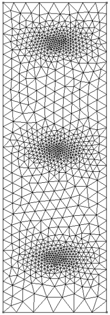
There exists an efficient algorithm (i.e., in polynomial time) that counts perfect matchings in a planar graph.

Cubic graphs, also called trivalent graphs, are graphs for which every node has exactly 3 adjacent nodes. Every cubic graph has at least one perfect matching (Oum S., *Perfect Matchings in Claw-free Cubic Graphs*). It can be proven that the number of perfect matchings in a cubic graph grows exponentially with $\#V$.

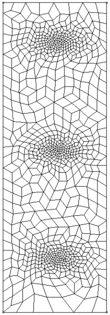
On closed surfaces, every triangular mesh has a perfect matching!

Extra Edges

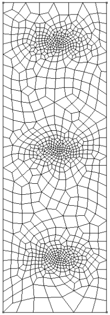




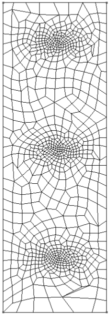
Initial triangulation



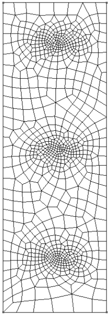
Raw Blossom application



Vertex smoothing

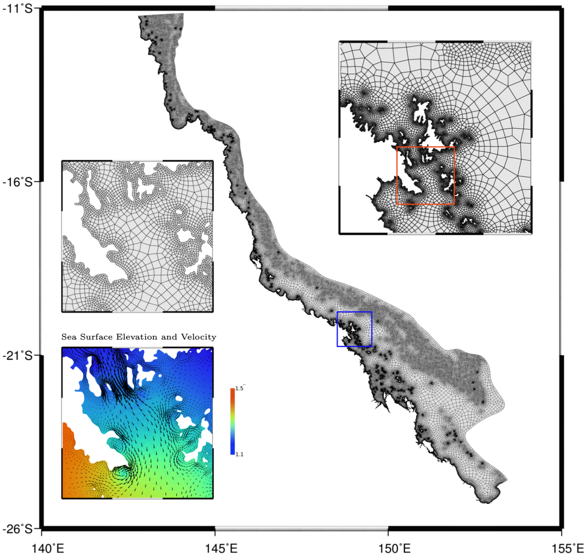


Topological optimization



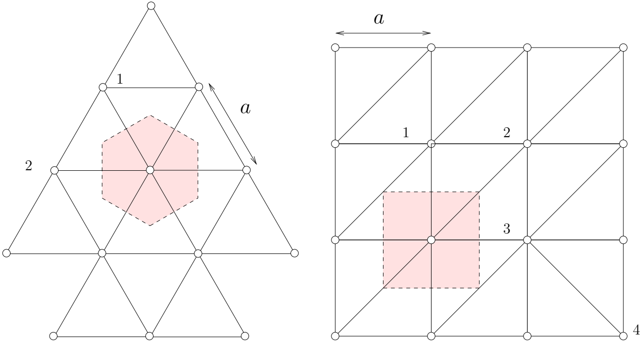
Final mesh

Great Barrier Reef



Gmsh's second attempt: delquad

JFR et al. *A frontal Delaunay quad mesh generator using the L^∞ norm*, IJNME, 2010.



Left : The Voronoi cell of each vertex is an hexagon of area $a^2\sqrt{3}/2$

Filling R^2 with equilateral triangles requires thus $2/\sqrt{3}$ times more vertices (i.e. about 15% more) than filling the same space with right triangles.

Gmsh's second attempt: delquad

Gmsh's surface mesher is a delaunay-frontal algorithm. Largely inspired by Rebay, S. (1993). *Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm.* Journal of computational physics, 106(1), 125-138.

Combine the robustness of Bowyer-Watson and triangle quality control of frontal algorithms.

Extension to surface meshing and the devil is in the details. One of Gmsh's oldest algorithms.

An example speaks louder than a long speech.

I'll have to hack Gmsh *to the bone*.

Gmsh's second attempt: delquad

Gmsh's frontal delaunay algorithm tries its best to make equilateral triangles.

A front edge e separates triangles that are "done" and other ones that are "not done".

A new point is added on the orthogonal bissector of e to eventually create an equilateral triangle.

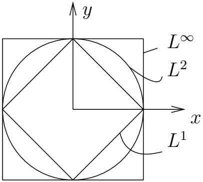
It is possible to very slightly modify the frontal algorithm to create right triangles.

Triangulation in the L^∞-norm

The L^∞-norm distance

$$\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|_\infty = \lim_{p\to\infty} \left\|\mathbf{x}_2 - \mathbf{x}_1\right\|_p = \max\left(|x_2 - x_1|, |y_2 - y_1|\right).$$

Unit cirlces.

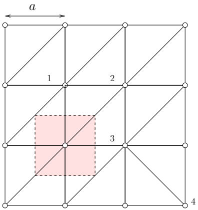


The 2-norm is the only norm that is rotationally invariant.

We thus use a cross field to define a local frame at point $\mathbf{x}$.

Triangulation in the L^∞-norm

In the L^∞ norm, the following mesh is made of equilateral triangles only.

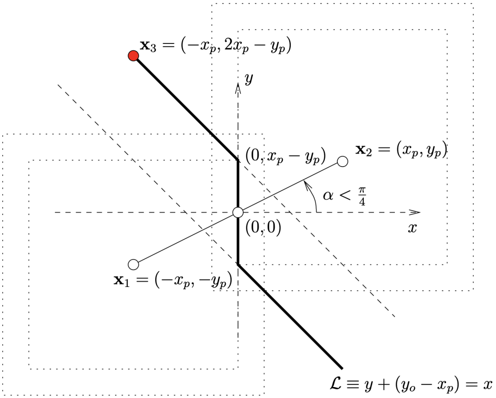


It is possible to use the same frontal-delaunay algorithm by computing orthogonal bisserctors in the L^∞-norm.

Bisectors in the L^∞-norm

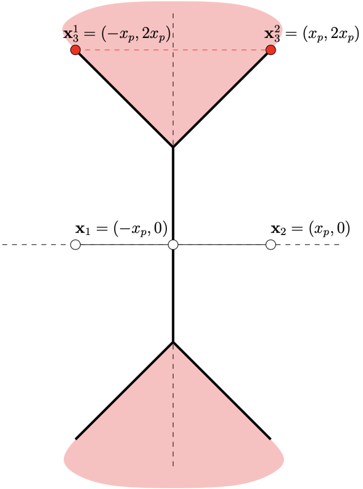
The perpendicular bisector, or bisector of the segment delimited by the points $\mathbf{x}_1 = (-x_p, -y_p)$ and $\mathbf{x}_2 = (x_p, y_p)$ is by definition the set of points $\mathbf{x} = (x, y)$ equidistant to $\mathbf{x}_1$ and $\mathbf{x}_2$.

It is the union of the intersections of circles centered at $\mathbf{x}_1$ and $\mathbf{x}_2$ and having the same radius.

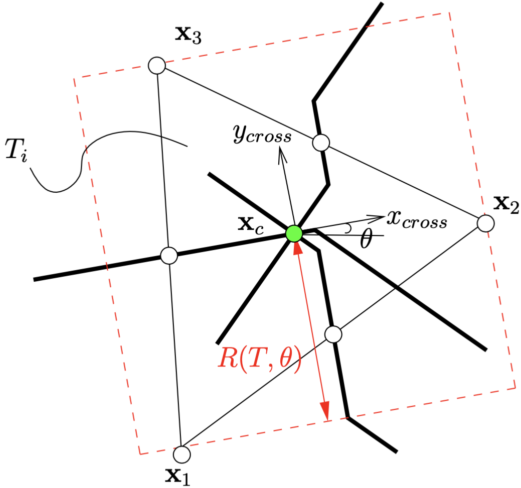


$\mathbf{x}_3 = (-x_p, 2x_p - y_p)$
y
$(0, x_p - y_p)$
$\mathbf{x}_2 = (x_p, y_p)$
$\alpha < \frac{\pi}{4}$
$(0, 0)$
x
$\mathbf{x}_1 = (-x_p, -y_p)$
$\mathcal{L} \equiv y + (y_o - x_p) = x$

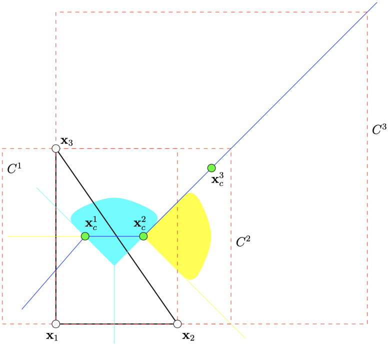
Bisectors in the L^∞-norm


$\mathbf{x}_3^1 = (-x_p, 2x_p)$
$\mathbf{x}_3^2 = (x_p, 2x_p)$
$\mathbf{x}_1 = (-x_p, 0)$
$\mathbf{x}_2 = (x_p, 0)$

Circumcenter in the L^∞-norm



Circumcenters in the L^∞-norm



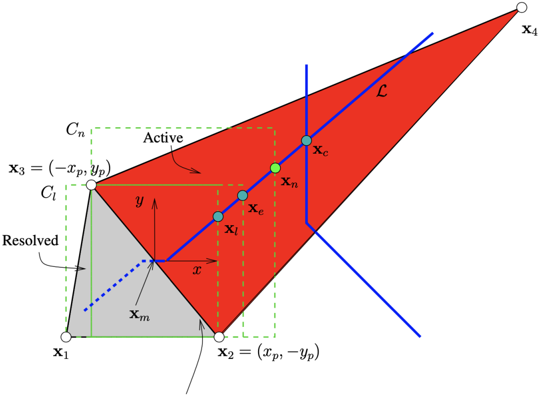
A right triangle. Perpendicular bisectors of the three segments are coloured in yellow (edge $\mathbf{x}_1\mathbf{x}_3$), blue (edge $\mathbf{x}_2\mathbf{x}_3$) and cyan (edge $\mathbf{x}_1\mathbf{x}_2$).

Points $\mathbf{x}_c^1$, $\mathbf{x}_c^2$ and $\mathbf{x}_c^3$ are three circumcenters that correspond to the three circumsquares C^1, C^2 and C^3.

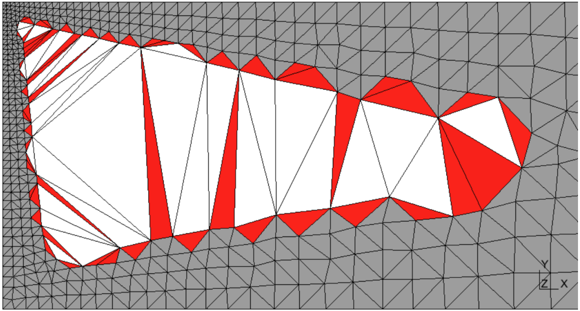
Circumcenter and circumsquare are unique when the points are in general position.

Delquad

- The new point should not be placed beyond the center $\mathbf{x}_c$ of the circumsquare of the active triangle (red triangle), as this would create a triangle with a small edge $\mathbf{x}_n\mathbf{x}_4$.
- The new point should not be placed below the intersection $\mathbf{x}_l$ of the bisector $\mathcal{L}$ and the circumsquare C_l of the resolved triangle $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$. Inserting a point inside C_l would make the resolved triangle invalid by means of the Delaunay criterion.
- If $\delta'(\mathbf{x}_m) = \|\mathbf{x}_3 - \mathbf{x}_2\|_\infty$, then the optimal point is $\mathbf{x}_n = \mathbf{x}_e$. It corresponds to the largest triangle $T_i(\mathbf{x}_e, \mathbf{x}_2, \mathbf{x}_3)$ that verifies $R_\infty(T_i, \theta) = \delta'(\mathbf{x}_m)$.



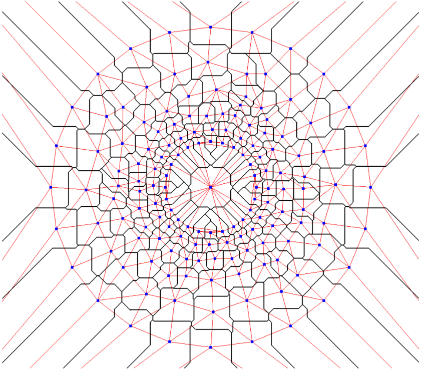
Delqual



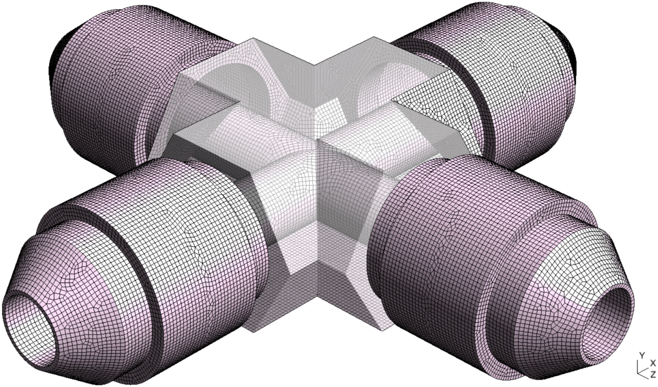
Delquad

We use standard Bowyer-Watson to connect the points i.e. we do Delaunay in the 2-norm.

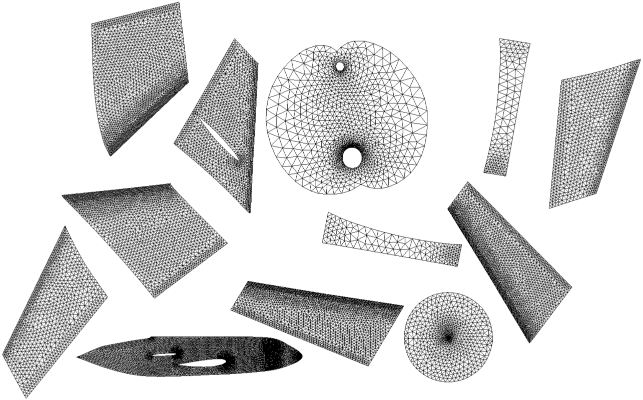
Yet, it has been observed experimentally that, in the case of finite element meshes with decent point distribution properties, the Delaunay kernel in the standard L^2-norm and the Delaunay kernel in the L^∞-norm give similar triangulations.



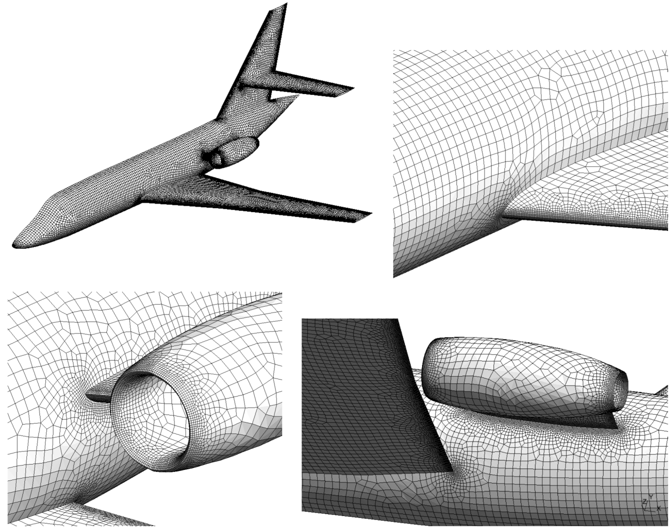
Delquad



Delquad

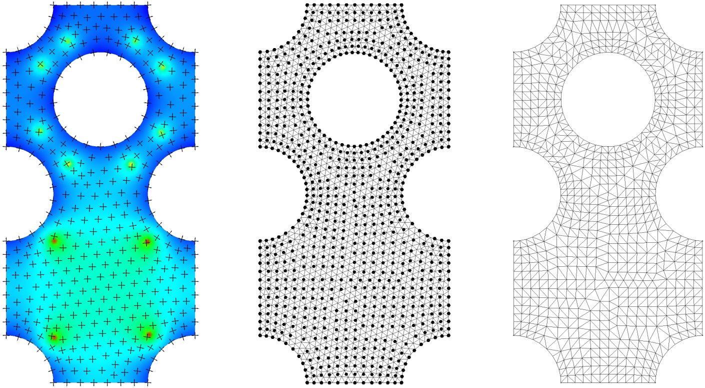


Delquad

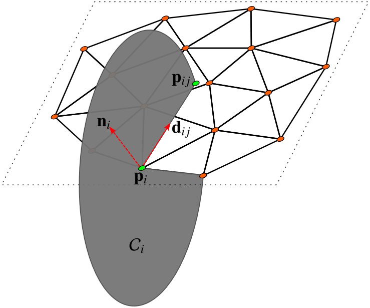


Gmsh’s third (& final) attempt: pack

Baudouin, T. C., Remacle, J. F., Marchandise, E., Henrotte, F., & Geuzaine, C. (2014). *A frontal approach to hex-dominant mesh generation.* Advanced Modeling and Simulation in Engineering Sciences, 1, 1-30.



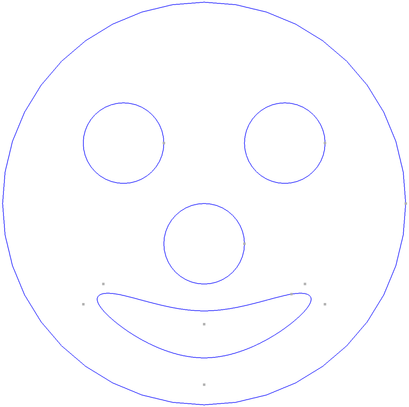
Gmsh's third (& final) attempt: pack



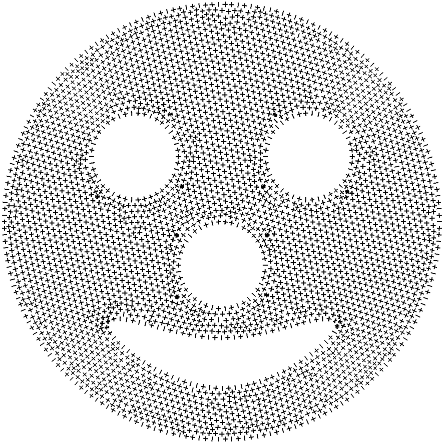
Algorithm 1 Frontal point insertion algorithm.

```
Input: Initial triangulation $\mathcal{T}_0$
       cross field $\mathbf{f}$
       mesh size field function $h(\mathbf{x})$
Output: Array of points $P$
Place boundary points in a queue
while queue is not empty do
    pop the first point $\mathbf{p}_i$ out of the top of the queue
    interpolate $\mathbf{f}$ and $h$ at this point
    for $2N_d$ directions do
        Compute point $\mathbf{p}_{ij}$ by intersecting $\mathcal{T}_0$ with a circle
        Find set of neighboring points $P_f$
        for $\mathbf{p}_f \in P_f$ do
            if $\|\mathbf{p}_{ij} - \mathbf{p}_f\| > \alpha h(\mathbf{p}_{ij})$ then
                add $\mathbf{p}_{ij}$ in $P$
                push $\mathbf{p}_{ij}$ in the back of the queue
            else
                delete $\mathbf{p}_{ij}$
            end if
        end for
    end for
end while
```

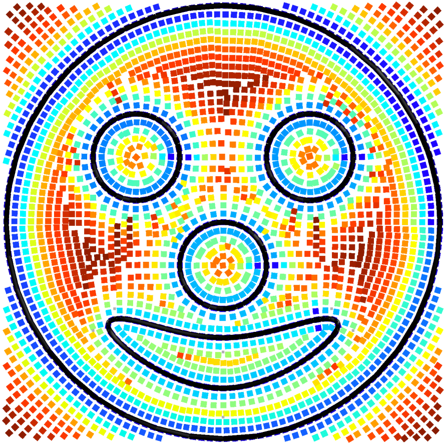
Gmsh's third (& final) attempt: pack



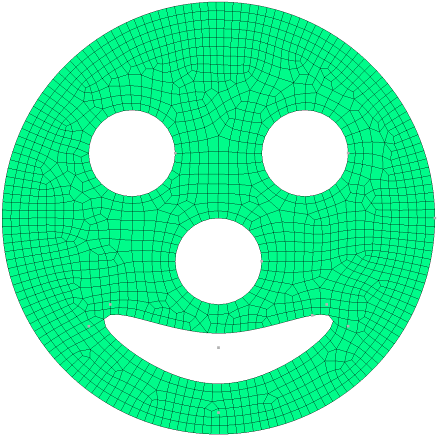
Gmsh's third (& final) attempt: pack



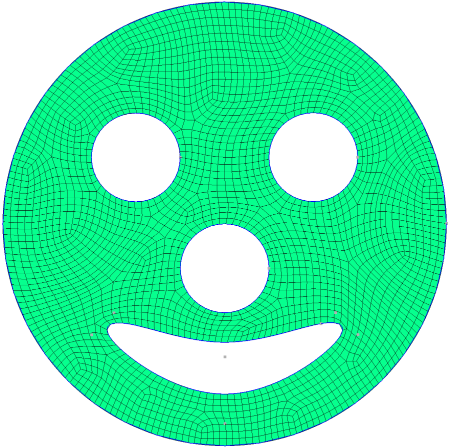
Gmsh's third (& final) attempt: pack



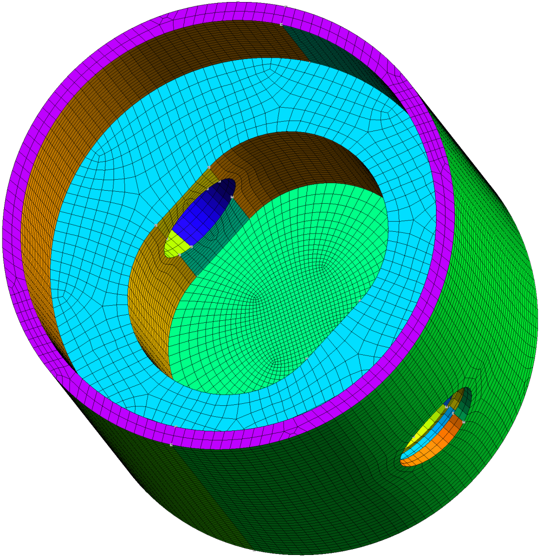
Gmsh's third (& final) attempt: pack



Improving pack: quadqs



Improving pack: quadqs



Improving pack: quadqs

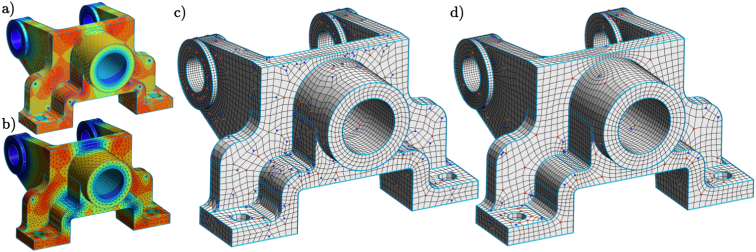
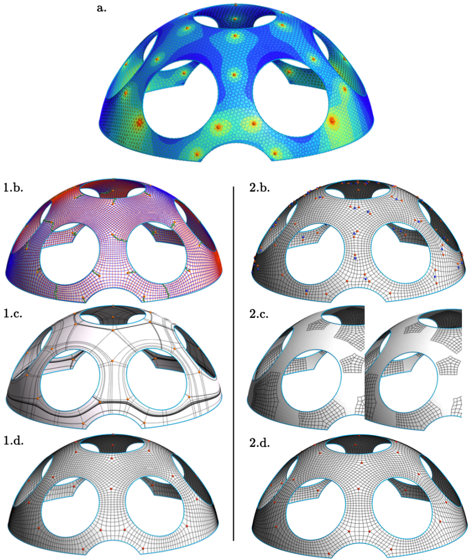
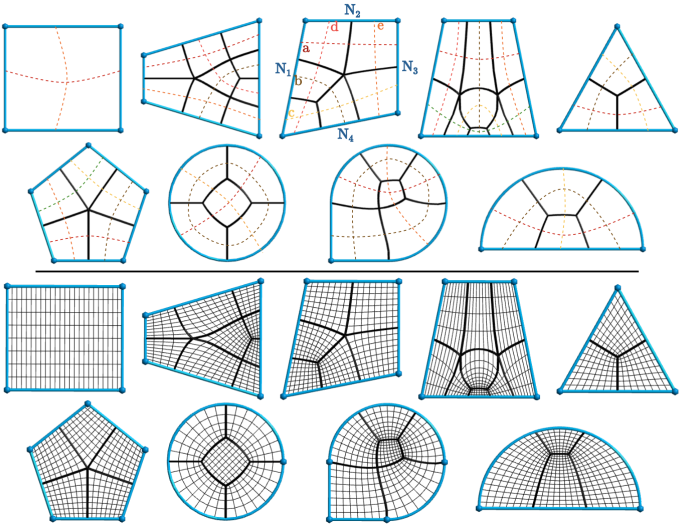


Fig. 1: The quasi-structured quad meshing pipeline applied on the model *M3* [24] (111 CAD corners, 169 curves, 60 faces). The cross field and its associated conformal scaling is computed on an initial triangulation (a). The size map (b) combines the cross field scaling and is adapted to the small CAD features. The initial unstructured all-quadrilateral mesh (c) is computed with a frontal approach, and contains 3429/54982 irregular vertices. After cavity remeshing, the final quasi-structured mesh (d) contains 606/56743 irregular vertices. 78 of them match the cross field singularities (a) and the others allow the mesh size transitions (b).

Improving pack: quadqs



Improving pack: quadqs



Improving pack: quadqs

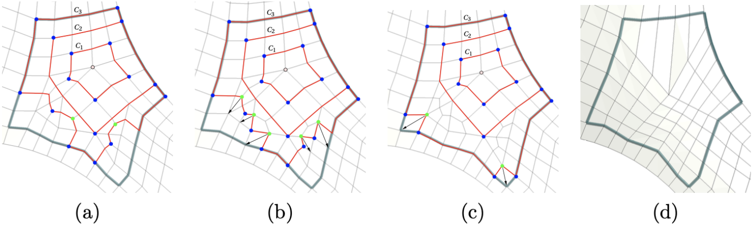


Fig. 6: Growing a cavity around one vertex of index −1 (in pink). Convex corners are in blue and concave corners are in green. The remeshed cavity (d) has one irregular vertex instead of eleven.

Improving pack: quadqs

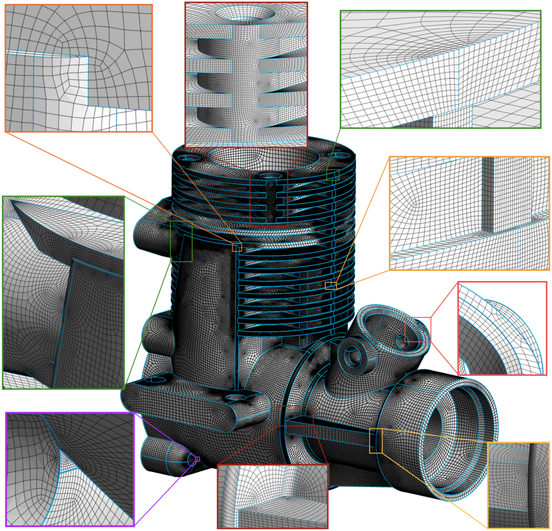


Fig. 9: Quasi-structured quad mesh (261.5k vertices, 261.6k quads) of the *Block* model (533 faces and 1584 curves). The average SICN quality is 0.87 and the minimum is 0.11. The initial unstructured quad mesh was generated in 58 seconds and the quasi-structured improvement took 33 seconds, both with 4 CPU cores on a laptop. The number of irregular vertices was reduced from 14.4k to 3.6k.

Unstructured Hex Meshing

Jean-François Remacle[1] and Christophe Geuzaine[2]

[1] Université catholique de Louvain (UCLouvain)

[2] Université de Liège (ULiege)

http://www.gmsh.info

October 7, 2024

Frontal approach

- Create a 3D frame/size field
- Generate points on surfaces & on volumes using the same approach
- Tetrahedralize the points (+ recover features)
- Subdivide tetrahedra into hexahedra
- Create a all-hex mesh?

```
gmsh Kolben.stp -clmin .3 -clmax .3 -hybrid -3 -nt 8
```

Subdividing a hexahedron into tetrahedra

Bounds on the number of tetrahedra:

$$n_v - n_e + n_f - n_t = 1$$

We have $n_v = 8$,

$$n_e = n_{ie} + n_{be} \text{ with } n_{be} = 12 + 6 = 18,$$

$$n_f = n_{if} + n_{bf} \text{ with } n_{bf} = 2 = 12$$

$$4n_t = 2n_{if} + n_{bf} \quad \to \quad n_{if} = 2n_t - 6.$$

All together (H. Edelsbrunner et al, *Tetrahedrizing point sets in three dimensions*, Journal of Symbolic Computation 10 (1990) 335–347)

$$8 - n_{ie} - 18 + (2n_t - 6) + 12 - n_t = 1 \quad \to \quad n_t = n_{ie} + 5.$$

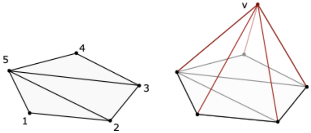
Since there are at most $n_{ie} = \binom{8}{2} - n_{be} = 10$ interior edges, we have the bounds

$$\boxed{5 \leq n_t \leq 15.}$$

174 Subdivisions

Pellerin, J., Verhetsel, K., & Remacle, J. F. (2018). *There are 174 Subdivisions of the Hexahedron into Tetrahedra*. ACM Transactions on Graphics (TOG), 37(6), 1-9.

A triangulation of the 2-sphere can be constructed from the triangulation of a 2-ball by building a cone.



The inverse transformation, the removal of one point v of the sphere triangulation as well as all triangles incident to v, permits to obtain the triangulation of a ball.

The 3-sphere is defined as the 3-dimensional boundary of a 4-dimensional ball. There are 1296 triangulations of the 3-sphere with 9 points

(Altshuler et al, *The classification of simplicial 3-spheres with nine vertices into polytopes and nonpolytopes.* Discrete Mathematics 31, 2 (1980), 115–124.)

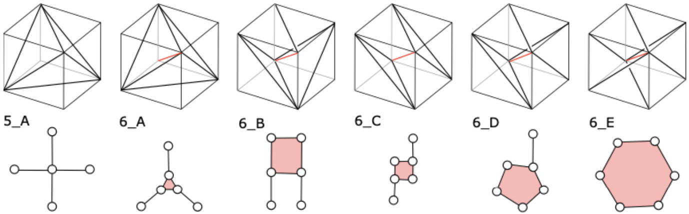


Figure 5: The six types of tetrahedrizations of the 3-cube and their dual complex representation.

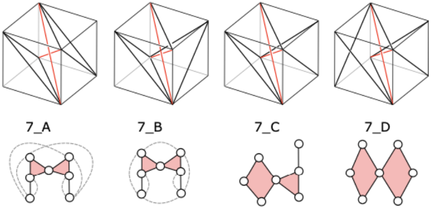
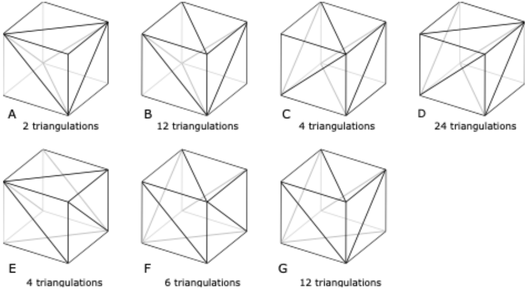


Figure 6: The four types of tetrahedrizations of an almost perfect cube into 7 tetrahedra proposed by [6] and their dual complex representation. Patterns 7_A and 7_B are differentiated by the edges linking vertices corresponding to tetrahedra that have a facet on the same hexahedron facet.

174 Subdivisions

Nine triangulations of the 3-ball with eight vertices can be built from each of the 1296 triangulations byremoving one of thevertices $v_i, i = 1, \ldots, 9$ and its link, i.e. all tetrahedra incident to v_i.

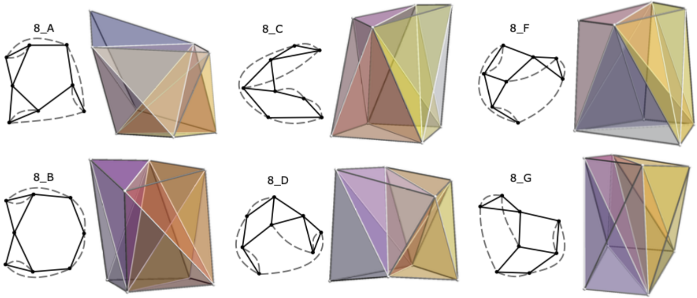
The triangulation of the boundary of a hexahedron has 8 vertices and 18 edges. Among these, 12 are fixed and there are 2 possibilities to place the remaining 6 diagonals of the quadrilateral facets. We have then $2^6 = 64$ possible triangulations. These triangulations can be classified into 7 equivalence classes, i.e. there are 7 triangulations of the hexahedron boundary up to isomorphism.



174 Subdivisions

The hexahedron has 174 combinatorial triangulations up to isomorphism that do not contain any boundary tetrahedra.

Among those 174 combinatorial triangulations, the 171 triangulations that admit an oriented matroid have a realization. The other ones cannot be realized.



174 Subdivisions

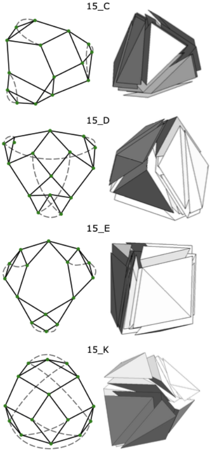
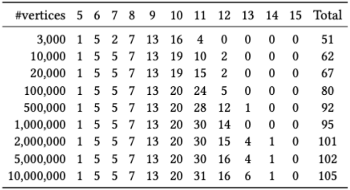


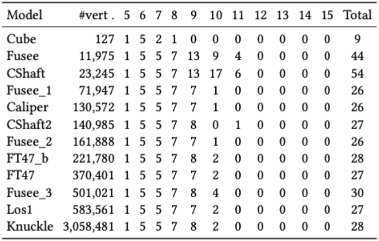
Fig. 13. The four realizable triangulations with 15 tetrahedra and with points in convex position. The hexahedra are valid, their Jacobian is strictly positive.

Table 4. Number of triangulations patterns per number of tetrahedra counted in the Delaunay triangulations of random point sets.



#vertices	5	6	7	8	9	10	11	12	13	14	15	Total
3,000	1	5	2	7	13	16	4	0	0	0	0	51
10,000	1	5	5	7	13	19	10	2	0	0	0	62
20,000	1	5	5	7	13	19	15	2	0	0	0	67
100,000	1	5	5	7	13	20	24	5	0	0	0	80
500,000	1	5	5	7	13	20	28	12	1	0	0	92
1,000,000	1	5	5	7	13	20	30	14	0	0	0	95
2,000,000	1	5	5	7	13	20	30	15	4	1	0	101
5,000,000	1	5	5	7	13	20	30	16	4	1	0	102
10,000,000	1	5	5	7	13	20	31	16	6	1	0	105

Table 3. Number of triangulations patterns per number of tetrahedra counted in the available input data of [Pellerin et al. 2018].



Model	#vert .	5	6	7	8	9	10	11	12	13	14	15	Total
Cube	127	1	5	2	1	0	0	0	0	0	0	0	9
Fusee	11,975	1	5	5	7	13	9	4	0	0	0	0	44
CShaft	23,245	1	5	5	7	13	17	6	0	0	0	0	54
Fusee_1	71,947	1	5	5	7	7	1	0	0	0	0	0	26
Caliper	130,572	1	5	5	7	7	1	0	0	0	0	0	26
CShaft2	140,985	1	5	5	7	8	0	1	0	0	0	0	27
Fusee_2	161,888	1	5	5	7	7	1	0	0	0	0	0	26
FT47_b	221,780	1	5	5	7	8	2	0	0	0	0	0	28
FT47	370,401	1	5	5	7	7	2	0	0	0	0	0	27
Fusee_3	501,021	1	5	5	7	8	4	0	0	0	0	0	30
Los1	583,561	1	5	5	7	7	2	0	0	0	0	0	27
Knuckle	3,058,481	1	5	5	7	8	2	0	0	0	0	0	28

Combining

Pellerin, J., Johnen, A., & Remacle, J. F. (2017). *Identifying combinations of tetrahedra into hexahedra: a vertex based strategy.* Procedia engineering, 203, 2-13.

1. A set of mesh vertices V is initially sampled in the domain.
2. A tetrahedral mesh T is built by connecting V, e.g. using a Delaunay kernel like.
3. A set H of potential that can be constructed by combining some of T is created.
4. A maximal subset $H_c \subset H$ of compatible is determined. It has been shown that this stage can be formally written as a maximal clique problem.
5. The , T', that are not combined into are combined into prisms, pyramids, or remain unchanged in the final *hex-dominant mesh*.

Combining

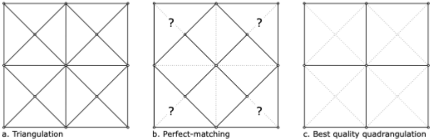


Figure 1: Combining pairs of triangles into quadrilaterals may not lead to the best quadrilateral mesh.

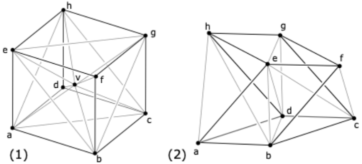


Figure 2: Two potential hexahedra that are not identified by existing combination methods. (1) A decomposition with an interior vertex v. (2) A decomposition into eight tetrahedra. This is a counter example to the claim of [7] that there is no decomposition into more than 7 interior tetrahedra.

Combining

Eight vertices of the tetrahedral mesh T define a potential hexahedron if (1) the twelve hexahedron edges are edges of T and if (2) the six quadrilateral hexahedron faces can be formed by merging two triangular facets of T.

This starting point is quite general and allows to automatically detect potential hexahedra without having to define a priori decomposition patterns into tetrahedra.

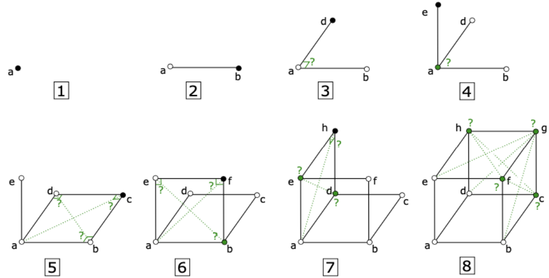


Figure 9: Vertex based search algorithm to build one hexahedron. Starting from one vertex a, the 7 other vertices are added one after another searching for vertices that are that are adjacent through edges of the tetrahedral mesh. Tests on the existence of triangulations of faces ensure the existence of the hexahedron boundary. Tests on the quality of 2D face angles and 3D hexahedron angles help the quick discard of bad hexahedra.

Maximal independent set

In graph theory, an independent set is a set of vertices in a graph, no two of which are adjacent.

Create a graph – nodes at the potential hexes and an edge exist between two hexes if they share a tet.

The optimization problem of finding a maximum independent set is a strongly NP-hard problem (in 2D, Blossom is polynomial!).

Greedy algorithm: choose the best hex h, remove all hexes that are connected to h i.e. that share a tet with h, choose the best remaining hex and so on.

Hex-dominant

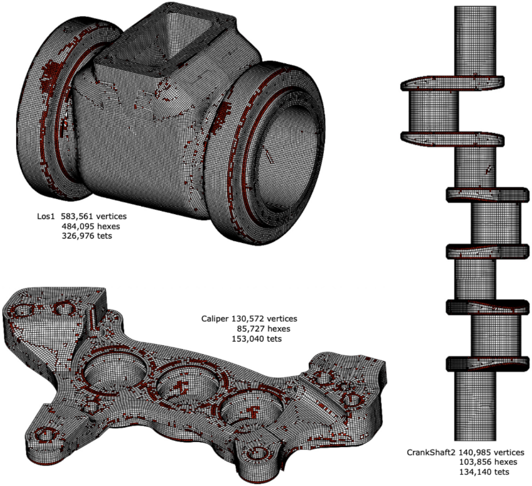
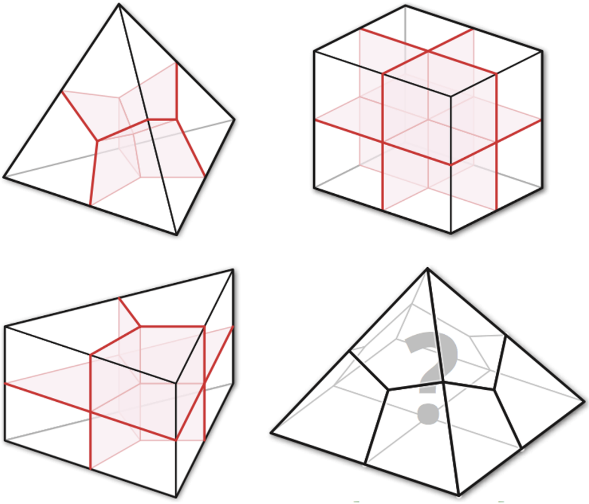


Figure 10: Hexahedral dominant meshes generated by greedy selection of the best quality hexahedra among those identified by our algorithm (white: hexahedra, red: tetrahedra). All potential hexahedra are identified in typically less than a minute, the greedy selection runs in a few seconds.

```
gmsh Los1.stp -clmin 1.5 -clmax 1.5 -hybrid -3 -nt 8
```

Full Hex?



Full Hex?

Verhetsel, K., Pellerin, J., & Remacle, J. F. (2019). *A 44-element mesh of Schneiders' pyramid Bounding the difficulty of hex-meshing problems. Computer Aided Design.*

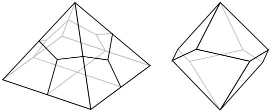


Fig. 1: Left: Schneiders'pyramid. Right: The octogonal spindle.

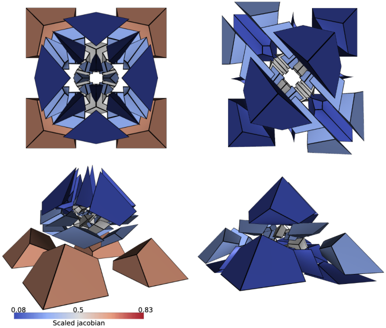


Fig. 2: Comparison of our 44-element mesh of Schneiders' pyramid (left) with the smallest known 36-element solution (right). Both admit two planar symmetries.

Full Hex?

Erickson J. (2014). *Efficiently hex-meshing things with topology.* Discrete & Computational Geometry 52, 3 (2014), 427–449.

Verhetsel, K., Pellerin, J., & Remacle, J. F. (2019). *Finding hexahedrizations for small quadrangulations of the sphere.* ACM Transactions on Graphics (TOG), 38(4), 1-13.

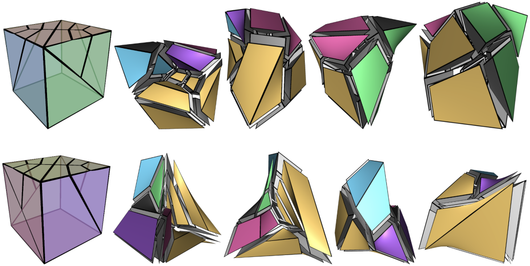


Fig. 10. Hexahedrizations of the two types of buffer cubes used to mesh arbitrary domains in the algorithm of Erickson [2014]. (top) 37 hexahedra to mesh the 20-quadrangle cell; (bottom) 40 hexahedra to mesh the 22-quadrangle cell. Colors correspond to the different sides of the original cubes (shown on the left).

Any ball-shaped domain bounded by n quadrangles can be meshed with no more than $78n$ hexahedra. This papier gives bounds that are very significantly lowers the previous upper bound of $5396n$.