Unstructured polyhedral mesh generator with improved grid quality metrics

Manuel Dias da Costa
manuel.d.costa@ist.utl.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

June 2018

Abstract

In this Thesis, a two dimensional mesh generator, named *Formiga* (acronym for Forced Mesh Improvement and Generation Algorithm), capable of creating triangular and polyhedral grids is developed with the end goal of increasing numerical accuracy in simulations with the Finite Volume Method. The main characteristic of the generator is the iterative smoothing algorithm based in the forcing approach and Finite Volume specific grid quality metrics, which displaces the vertices of the mesh so that the said metrics are improved.

Other important features include the change in mesh connectivities for different cell types and an algorithm to endow the polyhedral meshes with the ability of handling non-convex geometries.

Meshes over several geometries are presented to showcase the robustness of the developed algorithm and comparisons with other triangular and polyhedral mesh generators are established to prove the quality of the meshes obtained using the proposed approach.

These meshes are also used to perform the numerical simulation of problems with a known analytical solution to prove that the applied methods decreases the numerical error. Given that both triangular and polyhedral meshes are tested a comparison between cell shapes is also performed along with a review of four families of discretization schemes for unstructured grids.
Keywords: Mesh generation; Unstructured polyhedral meshes; Finite volume method; Grid smoothing algorithm; Grid quality metrics; Numerical accuracy

1. Introduction

Unstructured meshes do not obey a regular order of nodes or cells and aren't restricted to a particular shape of cell or element, existing in triangles, quadrilaterals and polygons with an arbitrary number of faces when in 2D. There's a wide variety of methods available to generate this kind of mesh depending on the shape of cell preferred, although for this work only triangular and polyhedral ones are studied.

For triangular meshes, the three main groups identified [1] are Delaunay triangulation, Advancing Front and Quadtree.

Polyhedral mesh generation algorithms are split between direct and indirect methods. The direct methods include the Centroidal Voronoi Tessellation (CVT), that may also be used to create triangular meshes. In [2] there's a more in dept description of this method by the same authors who made available the *Polymesher* code [3], 2D MATLAB polyhedral mesh generator that uses the CVT.

The indirect methods use a triangular mesh as a primal mesh from which the polyhedral one is created. In [4] two ways of achieving this goal are mentioned. In the first one, polygons are created by bisecting the lines between each triangle's central point and its neighbors. A full description of this method is given by [5].

The second indirect method is to compute the dual of a Delaunay triangulation, this method, often referred to as dual method, is explained in detail in the works [4, 6].

Studies comparing cell shapes [7, 8, 9] exist and conclude that polyhedral meshes require less cells than triangular ones to achieve the same level of accuracy.

The quality of a mesh, independently of its cell shape is also important. Several metrics exist to judge grid quality but some are dependent on the cell shape. The following metrics work for cells with arbitrary number of faces. Juretić in [7] described three metrics: non-orthogonality angle, skewness and uniformity, which are directly related to the discretization schemes for unstructured meshes. These metrics measure some grid's characteristics, at the faces of the control volumes, known to affect the accuracy of numerical solutions. In [10] four metrics are addressed of which the *Hybrid mesh quality metric I* and *II* appear to be the most promising. These particular metrics were created with information about the cell's size and shape.

A way of improving mesh quality is by using a smoothing algorithm that alters node/vertex position without changing connectivities. Among these are optimization based algorithms [11], the Laplacian smoothing [12] and the *force-based method* [13].

Distmesh [13] is a Delaunay triangulation code that uses the *force-based method*. *Formiga* uses the triangular meshes from this generator and improves them by employing changes in connectivities and a novel smoothing algorithm based on the same method. *Formiga* also has the ability to generate polyhedral meshes by computing the dual of a triangular mesh and improving it by using the same smoothing algorithm and different changes in cell topology.

2. Background

2.1. Grid quality criteria

The mesh quality criteria used in *Formiga* are the ones proposed by Juretić [7]: non-orthogonality angle, skewness and uniformity. All of which are measured on the faces of the Control Volumes (CV) and are specific to the Finite Volume Method (FVM). Figure 1 introduces the nomenclature used.

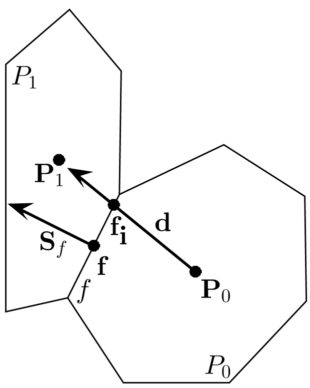


Figure 1: Two polyhedral cells with significant entities represented.

A face is considered non-orthogonal when the angle α_N between vectors $\mathbf{d}$ and $\mathbf{S}_f$, which may be expressed by Eq. 1, is different from zero. With $\mathbf{d} = \mathbf{P}_1 - \mathbf{P}_0$ and $\mathbf{S}_f$ exterior face normal. $\mathbf{P}_1$ and $\mathbf{P}_0$ are the coordinate vectors of the geometric centers of cells P_1 and P_0, respectively.

$$\alpha_N = \arccos\left(\frac{\mathbf{d}.\mathbf{S}_f}{||\mathbf{d}||\,||\mathbf{S}_f||}\right) \tag{1}$$

This quality criterion, that may also be referred to as warp angle, is evaluated in every face of the mesh, including boundary faces. In these situations vector $\mathbf{d}$ is defined by , $\mathbf{d} = \mathbf{f} - \mathbf{P}_0$, where $\mathbf{f}$ represents the face geometric centre.

Skewness or eccentricity is a measure of the distance between a face geometric centre and the intersection of vector $\mathbf{d}$ with said face, $\mathbf{f_i}$ represents this intersection point.

In this work this criterion shall be computed through Equation 2 which was adapted from [14] for 2D cases. Where $\mathbf{m}$ is defined as $\mathbf{m} = \mathbf{f} - \mathbf{f_i}$. The reasoning behind this Equation is that in this form it's easy to gage the position of $\mathbf{f_i}$. When $\psi = 1$, means that $\mathbf{f_i}$ coincides with one of the vertices that defines the face and when $\psi = 0$, $\mathbf{f_i}$ coincides with $\mathbf{f}$.

$$\psi = 2\frac{||\mathbf{m}||}{||\mathbf{S}_f||} \tag{2}$$

In the way it is defined it only makes sense to measure skewness values, unlike non-orthogonality, in interior faces. The same goes for the following and quality metric, uniformity.

Uniformity is a measure of the relative position of a face in regards to the cells that share it. If a mesh is uniform Eq. 3 will return $f_x = 0.5$.

$$f_x = \frac{||\mathbf{f_i} - \mathbf{P}_1||}{||\mathbf{d}||} \tag{3}$$

All three grid quality criteria have a close relation to the numeric schemes used in the FVM. Skewness is closely related to the convective schemes while the warp angle is associated with diffusive schemes. There are families of schemes that propose methods to correct or mitigate the effects caused by these grid quality metrics.

2.2. *Force based method*

This method is employed in *Distmesh* [13] to help improve vertex distribution during mesh triangulation.

This method is an iterative smoothing algorithm inspired by the analysis of trusses. By definition [15] "A truss consists of straight members connected at joints. Truss members are connected at their extremities only; thus no member is continuous through a joint.". The resemblance with computational meshes is striking and one could almost change the words 'truss', 'members' and 'joints' for 'mesh', 'edges' and 'points/nodes', respectively, and obtain a, though incomplete, mesh definition. Taking advantage of this analogy the method developed by Persson and Strang [13] applies forces to the points/nodes of a mesh, that are later converted in displacements to help the improvement of mesh quality throughout the generation process.

The conversion from forces into displacements is done through Equation 4. In [13] and this work the same equation is approximated with the Forward Euler method (Eq. 5).

$$\frac{d\mathbf{p}}{dt} = \mathbf{F}(\mathbf{p}) \tag{4}$$

$$\mathbf{p_{n+1}} = \mathbf{p_n} + \mathbf{\Delta t}\,\mathbf{F}(\mathbf{p_n}) \tag{5}$$

The force, $\mathbf{F(p)}$, used in *Distmesh* is modeled after Hooke's law. The idea behind it is to make every edge in the mesh comply to the specified *edge length* function `fh`. This is done by computing the difference between each edge's desired and actual length. For cases where this difference is larger than zero the computed value is multiplied by a constant k to obtain a force. If the above mentioned difference is smaller than zero the resulting force is zero.

Any vertex found outside the domain after the forces are applied is projected back to the boundary.

3. Grid generator algorithm

In the following sub-sections the main building blocks of *Formiga* are presented.

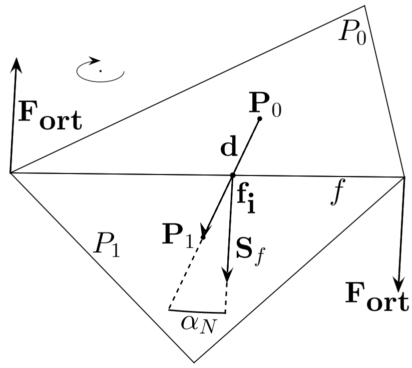
3.1. Grid quality corrections

Three corrections are proposed next, one for each of the presented quality metrics. All are directly proportional to a relevant metric concerning the respective quality criterion.

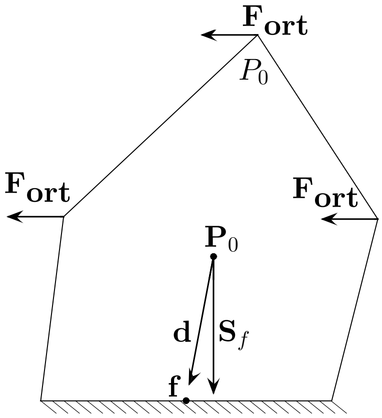
To reduce non-orthogonality or warp angles the strategy adopted was to apply forces with equal intensity but opposite directions to the vertices that compose the edge, f. These forces are perpendicular to f and their intensity is directly proportional to the angle, α_N, resulting in the rotation of the face. Figure 2(a) is an illustration of the described process.

This approach achieved the best results however there are other forms of achieving the same goal. One possibility might be applying a displacement to one of the edge's points instead of both. The main advantage against the former method is that in situations where the improvement of this quality criterion directly reduces the quality of other metrics, only moving one point allows more freedom for the other corrective forces to work with.

Another way to approach the situation is by going the opposite route and instead of trying to improve the warp angle by moving the face vertices, move the remaining cell's points so that the centroids position change and vector $\mathbf{d}$ becomes parallel to $\mathbf{S}_f$. This approach did not achieve better results than the previous one, however in situations where moving the face's vertices is not an option, when improving boundary edges, this is a valid option. As such, it is used to improve the non-orthogonality angle of boundary faces. Figure 2(b) showcases this method where, for clarity, the normal vector $\mathbf{S}_f$ was represented in the cell's centroid instead of in the actual face, for the image to be more readable. In the presented way its easier to gage the angle α_N. In the same spirit of making the images that make Figure 2, and the following, easily read the corrections shown are only for one face.



(a) Correction force in inner faces.

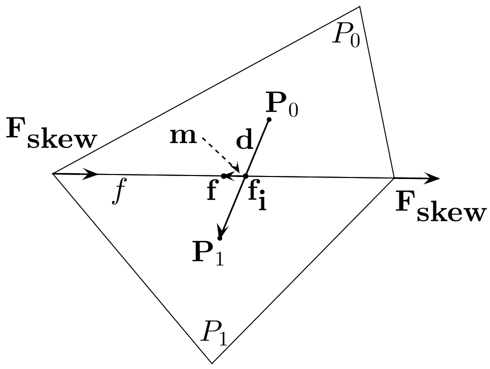


(b) Correction force in boundary faces.

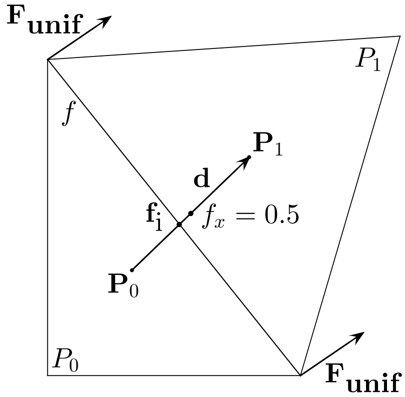
Figure 2: Corrections proposed for the non-orthogonality angle α_N.

From Eq. 2 stands that to reduce skewness values the distance between the edge's midpoint $\mathbf{f}$ and $\mathbf{f_i}$ must be shorten, and if possible, make them coincide. By applying a force in the opposite direction of $\mathbf{m}$ to the two vertexes that form the edge to be improved, Fig. 3(a), $\mathbf{f}$ is moving closer to $\mathbf{f_i}$. Note that by doing this $\mathbf{f_i}$ will change position as well. However, it shouldn't move at the same rate as $\mathbf{f}$, allowing for an approximation. Following the same principles as the forces above, this force is proportional to the length of vector $\mathbf{m}$.

The method used to improve the uniformity criterion follows the exact same guidelines as the previous ones and is shown in Fig. 3(b). Unlike the other two quality criteria where it's clear that the goal is to make them as close to zero as possible,



(a) Correction force for skewness at face f.



(b) Correction force for uniformity at face f.

Figure 3: Correction forces for skewness and uniformity.

uniformity poses a different problem as this is the only error that may be introduced voluntarily. For an uniform mesh is clear that the displacements applied must render $f_x = 0.5$ on all faces. However for an user-defined size distribution different than that its not as clear. To truly correct uniformity one would enforce $f_x = 0.5$ nonetheless but this would step over the user input resulting in an unwanted mesh.

Given this situation it was decided that these forces purpose would be to enforce the size distribution defined by the user and not mesh uniformity *per se*. The implications of this decision is that, depending on the situation, the forces applied might not correct uniformity at all.

In order to achieve the defined goal one must establish the reference values against which the uniformity will be measured. The relative size of each cell is measured using it's geometric centre and a ratio, is computed between its size and the neighbouring cells, with whom it shares a face, Eq. 6. In *Formiga* this is achieved by using the *Distmesh*'s [13] input variable `fh` that represents the *edge length function*. This ratio will be the that value the algorithm will try to enforce. To keep this reference values as unpolluted as possible this calculation is not performed in every iterative step but rather only in the first iteration and every time the grid's connectivities are modified.

$$\textit{cell ratio} = \frac{\texttt{fh}(\mathbf{P}_0)}{\texttt{fh}(\mathbf{P}_0) + \texttt{fh}(\mathbf{P}_1)} \tag{6}$$

Considering everything mentioned about the uniformity criterion, it was felt that though used in the smoothing process the numeric values of uniformity are hard to interpret and easy to misjudge so these won't be presented in Section 4.

3.2. Smoothing algorithm

In *Formiga* the proposed corrections are applied through the *force-based method* to already defined meshes. The four forces discussed until this point may be used together to create a function in the form of $\mathbf{F} = a\ \mathbf{F}_{\mathbf{ort}} + b\ \mathbf{F}_{\mathbf{skew}} + c\ \mathbf{F}_{\mathbf{unif}} + d\ \mathbf{F}_{\mathbf{dist}}$. By adjusting the values a, b, c, d one may easily change the resulting mesh, adapting to the problem at hand and desired objective.

The stopping criterion may also be chosen, it was found that criteria based on the grid quality metrics work well. Therefore the most commonly used stopping criterion is to monitor both the average and maximum values of a specific quality metric and stop the iterative process when they improve less than a predefined value.

When improving triangular meshes, vertices that leave the domain are projected back with the same method used in *Distmesh* while polyhedral meshes use a different approach.

This new approach uses the boundary defined by the edges of the primal mesh to create a 1D space where the dual mesh boundary vertexes may move. In practice the forces applied to the dual mesh boundary vertexes are projected onto the primal edges to where they belong.

3.3. Changes in triangular meshes connectivities

A characteristic that was identified in several triangular meshes, from several mesh generators, that result in poor quality criteria according to the Finite Volume specific quality criteria introduced in Section 2. It is the presence of right-angled triangles on the domain's boundary. The term 'right-angled triangle' is employed loosely in this context since the triangles don't have in fact to be right-angled for this topology to be problematic, a more correct way of describing the situation would be: two boundary defining cells that also share an edge amongst them. Despite the remark this term shall be used during this work when referring to this problem.

This mesh topology results in faces with high warp angles and higher skewness values. In Fig. 4 one can see a large (for triangular mesh standards) distance between $\mathbf{f}$ and $\mathbf{f_i}$ (skewness) on the edge shared by cells P_1 and P_0 and big values of warp angle (over 20°) on the edges that these cells share with the boundary (again, $\mathbf{S}_f$ is represented in the cell's center for easier interpretation).

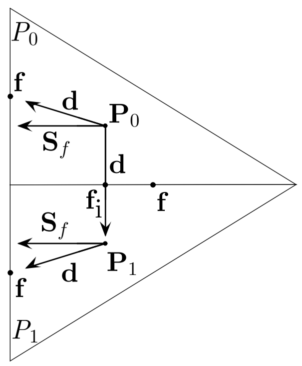


Figure 4: Grid quality criteria in 'right-angled triangles'.

The smoothing algorithm is not able to solve this predicament so several changes in the mesh topology are performed starting with the elimination of the edge shared by both cells in discussion. This alone would be enough to improve both the non-orthogonality and skewness values however would result in a very large cell. If the smoothing algorithm is capable of resizing this cell no further modifications are imposed.

For situations where this is not possible, a constrained Delaunay triangulation is performed, where the constraints are new edges created inside the aforementioned large triangles. These constraints are shown in Fig. 5, where the dashed gray lines represent the original triangular mesh and the black lines the applied constraints. The MATLAB function `delaunayTriangulation` is used to perform this triangulation.

On the same subject of low quality cells it was noted that when "corner" cells are shaped like the ones in Fig. 6 the boundary faces have high warp angle values and are also difficult to improve with the smoothing algorithm. To avoid this topology, edge swaps are performed resulting in "corners" shaped like the ones in Fig. 10(b).

In *Formiga* all triangular cell "corners" are shaped like the ones in the mentioned Figure, the only exception is when these triangular meshes are the basis for a dual polyhedral mesh. In this case the corners are forced to be like the example in Fig.

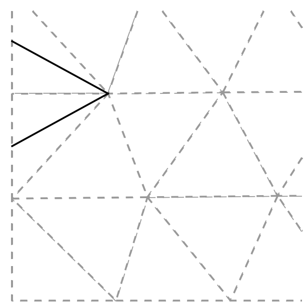


Figure 5: Constraints used to avoid 'right-angled triangles'.

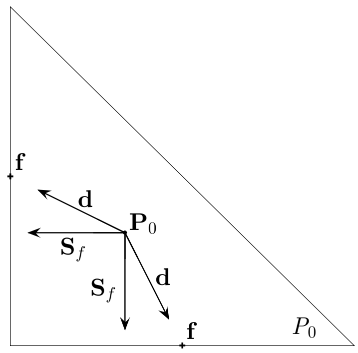


Figure 6: Quality criteria in bad "corner" cells.

6. The reasoning behind this is that the polyhedral mesh attained by computing the dual of meshes like the one in this figure have "square" cells in it's corners while the other topology results in pentagons. This is relevant because these pentagons are not regular, having one significantly smaller edge than the rest. Being smaller the chances of having significant skewness values are higher.

3.4. Concave cell elimination

A known issue in polyhedral meshes is the occurrence of concave cells when there are concavities in the domain of interest. Figure 7 shows a clear example. Popular CFD softwares are able to handle these non-convex geometries and in [16] three ways to "eliminate" these concave cells are presented for 3D meshes. Despite being possible to convert any of them to be applied in 2D it was decided to use a new approach in *Formiga*.

This novel approach was designed to work together with the already introduced smoothing algorithm meaning that the alterations described over the following lines won't render an high quality mesh until the smoothing algorithm is applied.

It's not uncommon in severe cases of concavity for the cell's centroid to be found outside the cell

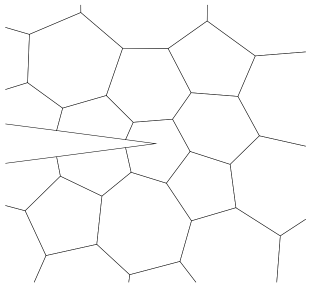


Figure 7: Example of concave cell with non trivial solution.

itself, resulting in an invalid cell. The first step is to ensure this cell is properly defined and respects the domain. Once this is achieved the following algorithm may be applied:

1. The vector that connects the centroid of the wedge found outside the domain to the concave point is computed;
2. Compute all cell's edges midpoints and search for the one whose vector formed by itself and the concave point is closer to making an angle of 180 degrees with the one computed in the previous step;
3. Once found, the edge to which this midpoint belongs is deleted and its two vertices connected to the concave point, effectively creating two new edges;
4. Delete the "concave cell" from the cell list and add two new ones that will split amongst each other the points used by the deleted cell.

The mesh represented in Fig. 8 is the same as in Fig. 7 after the application of the current correction and of the smoothing algorithm.

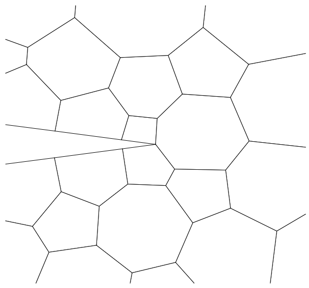


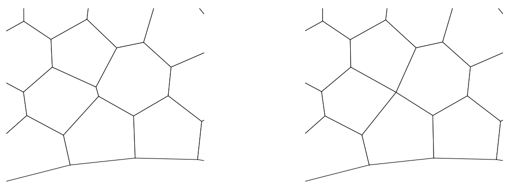
Figure 8: Results of introduced correction after the application of the smoothing algorithm.

3.5. Changes in polyhedral meshes connectivities

Like triangular meshes, polyhedral ones may suffer from problems left unresolved by the proposed smoothing algorithm. In general, cells of arbitrary number of faces are more manipulable than triangular cells so there isn't a particular topology that must be avoided in order for the smoothing algorithm to work. On the other hand its application may cause small edges or faces that lead to high skewness values.

The criterion defined to detect small edges is to compute the average length of a cell's edges and compare it against the length of each individual one. Case an individual edge is found to be smaller than a pre-established percentage of the average value, the edge is eliminated.

The process of edge elimination is in theory very simple, as one must only collapse a small edge into a single point keeping track of the changes caused in nearby cells. This approach of collapsing edges into points was also followed in the *Polymesher* code [3]. In Figure 9 there's an example of an edge elimination procedure.



(a) Initial mesh.

(b) Mesh after the edge elimination procedure.

Figure 9: Detail of a mesh before and after edge elimination.

Another unwanted scenario is to have a small cell. Despite being a very rare occurrence the ability to eliminate cells was added to *Formiga*. The criterion to eliminate a cell is its size. If a cell is smaller than it's neighbours average size, it is eliminated. Just like for the edge case, the eliminated cell is collapsed to a single point. In situations where the cell to be eliminated shares some edges with the boundary, this point must guarantee that the user's defined domain is respected. Otherwise the cell is collapsed to it's centroid.

A similar approach is also used for the edge elimination case. If an edge shares one vertex with the boundary or contains a fixed vertex it will collapse to that specific point. In any other case the edge will collapse to it's midpoint, instead.

In order to avoid unnecessary changes and to improve the chances of having a positive impact in the quality metrics the modifications are kept to a minimum. Cells are only eliminated if there are no suitable edges for elimination.

4. Results

In this Section, examples of meshes generated over two different geometries are presented. The first showcases most of the methods discussed here while in the second example a numerical solution is computed. Grid quality comparisons are established with *Distmesh* and its dual meshes to prove the improvements in the quality parameters.

4.1. L-shaped Domain

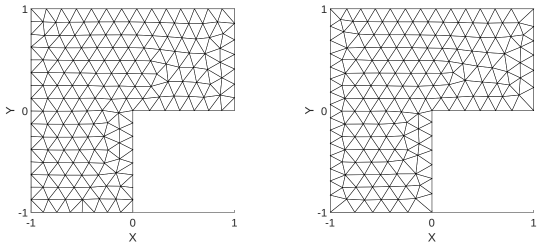
A good geometry to clearly show the improvements made is the L-shaped domain, defined by $[-1,\ 1]^2 \backslash ([0,\ 1] \times [-1,\ 0])$. What makes this a good example is the concavity, in which the introduced algorithm, may be applied and the fact that *Distmesh* frequently creates 'right angled triangles' in this domain. Fig. 10 shows the triangular meshes generated with *Distmesh* and *Formiga*. The main differences between them are the corner topology and the cells replacing the 'right-angled triangles'.

Table 1 details the improvement in grid quality metrics, that in the case of triangular meshes are mainly obtained through the presented changes in connectivities. The smoothing algorithm makes only small adjustments to further enhance the mesh quality.

		α_N *[deg]*		ψ	
	N.Cells	*Avg*	*Max*	*Avg*	*Max*
Distmesh	212	6.5178	32.1483	0.0476	0.3333
Formiga	206	3.6224	17.8596	0.0443	0.1965
Improvement		44.42%	44.45%	7.04%	41.05%
Distmesh	2124	2.4168	30.6401	0.0143	0.3333
Formiga	2119	1.6487	13.8971	0.0129	0.1864
Improvement		31.78%	54.64%	10.05%	44.08%
Distmesh	9922	1.1776	34.0130	0.0071	0.3333
Formiga	9915	0.9570	13.6313	0.0058	0.1956
Improvement		18.73%	59.92%	17.39%	41.31%
Distmesh	26740	0.6920	35.0352	0.0041	0.3333
Formiga	26733	0.5272	13.1603	0.0036	0.2107
Improvement		23.81%	62.44%	12.87%	36.79%
Average		29.69%	55.36%	11.84%	40.81%

Table 1: Grid quality comparison with *Distmesh* - L-shaped domain.

The polyhedral meshes from *Formiga* are compared with the dual mesh of *Distmesh* (named *Dist dual*). Fig. 11 shows an example of each mesh. The main difference in these figures is in the concave point of the geometry, in here one can clearly see that the algorithm presented to deal with non-convex geometries requires the use of the smoothing algorithm. Without it the result is oddly shaped cells with low quality metrics. In addition boundary cells increase significantly in size, mainly due to the growth of what were significantly skewed faces.



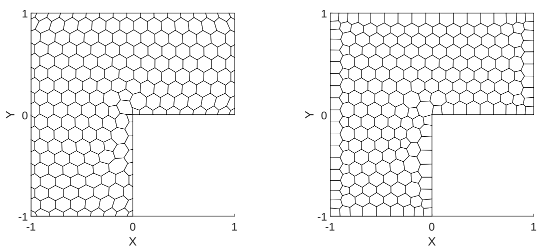
(a) Triangular mesh generated with *Distmesh*. (b) Triangular mesh generated with *Formiga*.

Figure 10: Examples of triangular meshes over an L-shaped domain.

		α_N *[deg]*		ψ	
	N.Cells	*Avg*	*Max*	*Avg*	*Max*
Dist dual	217	5.2730	28.4002	0.1141	0.7594
Formiga	219	0.4601	5.6465	0.0683	0.3863
Improvement		91.27%	80.12%	40.15%	49.14%
Dist dual	2209	1.2765	32.6320	0.0395	0.8071
Formiga	2207	0.5076	4.8324	0.0152	0.3536
Improvement		60.23%	85.19%	61.52%	56.19%
Dist dual	10782	0.7262	39.2428	0.0160	0.7992
Formiga	10783	0.0967	14.3688	0.0094	0.5367
Improvement		86.69%	63.38%	41.04%	32.85%
Dist dual	24138	0.4770	35.9490	0.0119	1.0500
Formiga	24138	0.0753	7.5522	0.0068	0.5694
Improvement		84.22%	78.99%	43.13%	45.77%
Average		80.60%	76.92%	46.46%	45.99%

Table 2: Grid quality comparison with *Dist dual* - L-shaped domain.

Table 2 shows large improvements in the quality metrics used, larger than in triangular meshes. The main factor for this higher improvement is the smoothing algorithm that takes advantage of a more manipulable topology to produce visible alterations.



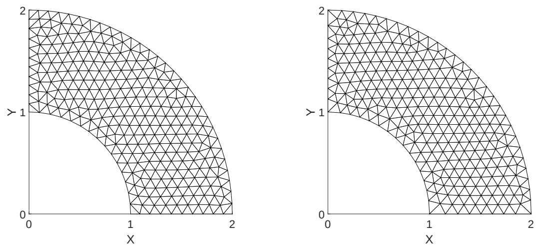
(a) Polyhedral mesh generated with *Dist dual*. (b) Polyhedral mesh generated with *Formiga*.

Figure 11: Examples of polyhedral meshes over an L-shaped domain.

4.2. Convective-Diffusive Equation - Quarter annulus domain

In this sub-section, the convective-diffusive equation is solved in an annulus segment centered at $(0,0)$, as seen in [17]. Tables 3 and 4 detail the quality metrics of the triangular and polyhedral meshes used. Given the nature of the problem the reduction of both, warp angle and skewness values is essential.

The triangular meshes used appear to have slightly better skewness values while polyhedral ones have better warp angle values.



(a) Triangular mesh generated with *Distmesh*. (b) Triangular mesh generated with *Formiga*.

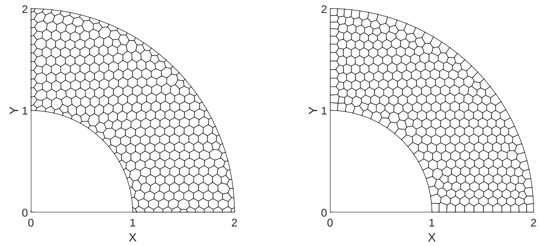
Figure 12: Examples of triangular meshes over a quarter annulus domain.

		α_N [deg]		ψ	
	N.Cells	*Avg*	*Max*	*Avg*	*Max*
Distmesh	503	4.5056	33.9281	0.0360	0.2383
Formiga	498	1.5174	18.1123	0.0322	0.2252
Improv.		66.32%	46.62%	10.43%	5.48%
Distmesh	1723	3.4162	32.7205	0.0205	0.3333
Formiga	1716	1.2006	17.7066	0.0176	0.1985
Improv.		64.86%	45.89%	14.09%	40.44%
Distmesh	12077	1.3896	32.0638	0.0073	0.3333
Formiga	12072	0.6823	17.6022	0.0061	0.1966
Improv.		50.90%	45.10%	16.63%	41.01%
Distmesh	53860	0.7136	32.0483	0.0035	0.3372
Formiga	53855	0.3902	18.9079	0.0030	0.1987
Improv.		45.32%	41.00%	15.41%	41.07%

Table 3: Quarter annulus - Triangular meshes quality metrics.

In Fig. 12 examples of the triangular meshes used are presented and the only obvious difference between the meshes is the corner cell topology. In Fig. 13 besides the increase in boundary cell mentioned previously there are also an instance of edge elimination in the corner of coordinates $(0,1)$.

The conditions to perform this simulations are all defined in [17] along with its analytical solution. The numerical schemes used are defined in [18]. Though, in short, four families of second order accurate schemes are used where the first, $D-NRML, C-LIN$, does not use any correction for unstructured grids while the other three



(a) Polyhedral mesh generated with *Dist dual*. (b) Polyhedral mesh generated with *Formiga*.

Figure 13: Examples of polyhedral meshes over a quarter annulus domain.

		α_N [deg]		ψ	
	N.Cells	*Avg*	*Max*	*Avg*	*Max*
Dist dual	350	3.9515	23.9622	0.1023	1.1542
Formiga	349	0.3129	3.3118	0.0459	0.3937
Improv.		92.08%	86.18%	55.16%	65.89%
Dist dual	1723	2.3642	25.3238	0.0451	1.0572
Formiga	1725	0.1983	3.5399	0.0244	0.6988
Improv.		91.61%	86.02%	45.89%	33.90%
Dist dual	12151	1.0087	24.4224	0.0179	1.0619
Formiga	12145	0.3190	11.4282	0.0073	0.4420
Improv.		68.37%	53.21%	59.39%	58.37%
Dist dual	55636	0.5044	24.1743	0.0085	1.0257
Formiga	55631	0.1656	10.5087	0.0039	0.5020
Improv.		67.16%	56.53%	53.76%	51.06%

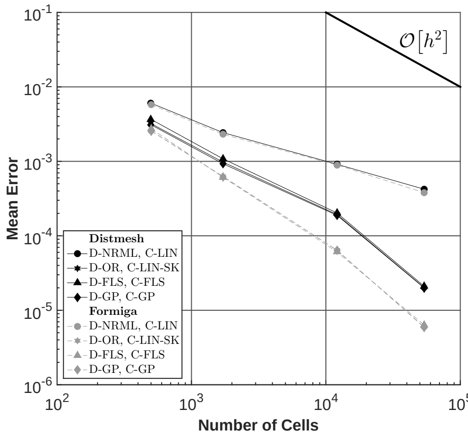
Table 4: Quarter annulus - Polyhedral meshes quality metrics.

have contingencies to deal with grid quality related issues.

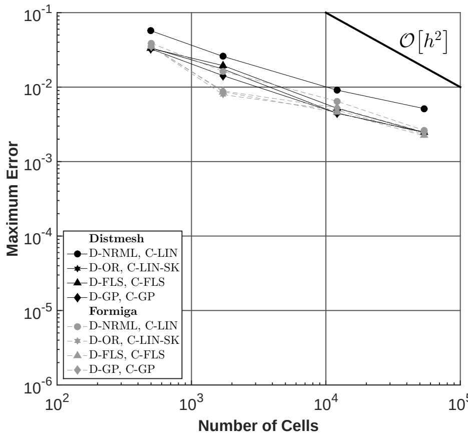
Figure 14 shows the results obtained with triangular meshes. *Formiga*'s meshes improve the average error by upwards of 40% on the three most accurate schemes used. The family of schemes, $D-NRML, C-LIN$, is not able to maintain the second order accuracy slope and scores the least accurate results of this test. The improvements in grid quality made, did not sort any major effect in the average results of this scheme but reduced its maximum error by more than 30%.

The results obtained with the polyhedral meshes are in Fig. 15. In this case the results for the family of classic schemes $D-NRML, C-LIN$ show that *Formiga*'s meshes had a positive effect on the results. Furthermore, they show sensibility to the mesh's grid quality. In the two most refined meshes, the warp angle values of *Formiga*'s meshes are larger than before, and their effect is clearly shown in Fig. 15(a).

The other three schemes keep their second order average accuracy slope, for all meshes tested. The modifications proposed score 50% less average error than polyhedral meshes with no improvements, and reduce the maximum error by over an order of magnitude, in the most refined meshes, used. These results show that the improvement of grid quality

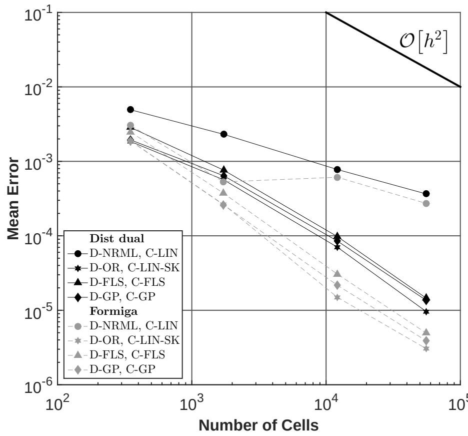


(a) Average error.

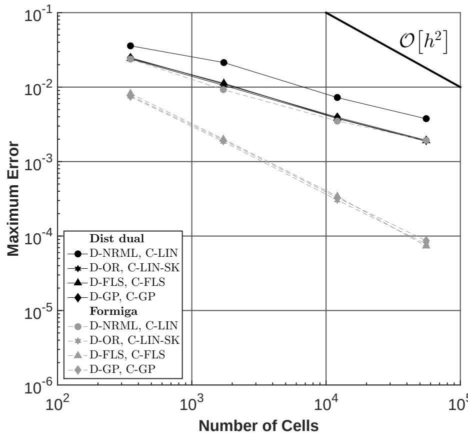


(b) Maximum error.

Figure 14: Quarter annulus - Triangular meshes numerical error.



(a) Average error.



(b) Maximum error.

Figure 15: Quarter annulus - Polyhedral meshes numerical error.

metrics is important, even for families of schemes with corrections terms for this issue.

5. Conclusions

A new mesh generator capable of producing high quality triangular and polyhedral meshes was presented. It's main characteristics include the identification and elimination of poor quality cells in triangular meshes, a new method to eliminate concave cells in polyhedral meshes and a smoothing algorithm based in force induced displacements driven by FVM specific grid quality metrics.

This smoothing algorithm shows good ability to make small adjustments in node position. Often seen in triangular meshes where the alterations are hard to identify but also to affect change in a significant way moving nodes over "large" distances as seen in the mesh's reconstruction after the elimination of concave cells, for example. It's fair to say that this algorithm is more effective in polyhedral meshes as it obtains higher improvements than in triangular ones.

The meshes generated with the routines presented in this work are more accurate than grids with worse quality metrics on the numerical solution tested. Improving both the average and maximum error, independently of the family of discrete schemes used, even those schemes for unstructured grids that depend on grid quality correction terms. This performance on numerical simulations ultimately validates both the choice of grid quality metrics and also the algorithm developed in this work.

The numerical problem solved confirms that polyhedral cells are more accurate than triangular ones for meshes of similar size.

References

[1] Steven J Owen. A survey of unstructured mesh generation technology. In *IMR*, pages 239–267, 1998.

[2] Cameron Talischi, Glaucio H. Paulino, Anderson Pereira, and Ivan F. M. Menezes. Polygonal finite elements for topology optimization:

A unifying paradigm. *International Journal for Numerical Methods in Engineering*, 82(6):671–698.

[3] Cameron Talischi, Glaucio H. Paulino, Anderson Pereira, and Ivan F. M. Menezes. Polymesher: a general-purpose mesh generator for polygonal elements written in matlab. *Structural and Multidisciplinary Optimization*, 45(3):309–328, 2012.

[4] Rao V Garimella, Jibum Kim, and Markus Berndt. Polyhedral mesh generation and optimization for non-manifold domains. In *Proceedings of the 22nd International Meshing Roundtable*, pages 313–330. Springer, 2014.

[5] F. Juretić. *Error Analysis in Finite Volume CFD*. PhD thesis, PhD Thesis, Imperial College London, 2004.

[6] G. Balafas. Polyhedral mesh generation for CFD-analysis of complex structures. *Master Thesis, TUM, Germany*, 2013.

[7] F. Juretić and A. D. Gosman. Error analysis of the finite-volume method with respect to mesh type. *Numerical heat transfer, part B: fundamentals*, 57(6):414–439, 2010.

[8] M. Perić. Simulation of flows in complex geometries:new meshing and solution methods. *NAFEMS Seminar*, May 3-4 2004. Germany.

[9] Man Young Kim, Seung Wook Baek, and II Seouk Park. Evaluation of the finite-volume solutions of radiative heat transfer in a complex two-dimensional enclosure with unstructured polygonal meshes. *Numerical Heat Transfer, Part B: Fundamentals*, 54(2):116–137, 2008.

[10] Jibum Kim and Jaeyong Chung. Untangling polygonal and polyhedral meshes via mesh optimization. *Engineering with Computers*, 31(3):617–629, Jul 2015.

[11] Jibum Kim, Myeonggyu Shin, and Woochul Kang. A derivative-free mesh optimization algorithm for mesh quality improvement and untangling. *Mathematical Problems in Engineering*, 2015, 2015.

[12] David A Field. Laplacian smoothing and delaunay triangulations. *International Journal for Numerical Methods in Biomedical Engineering*, 4(6):709–712, 1988.

[13] Per-Olof Persson and Gilbert Strang. A simple mesh generator in matlab. *SIAM review*, 46(2):329–345, 2004.

[14] J. P. Magalhães. An adaptive framework for the numerical simulation of environmental flows in complex geometries. *PhD Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa*, 2011.

[15] F. P. Beer. *Vector Mechanics for Engineers: Statics and dynamics*. McGraw-Hill Companies, 2009.

[16] Sang Yong Lee. Polyhedral mesh generation and a treatise on concave geometrical edges. *Procedia Engineering*, 124:174–186, 2015.

[17] Carl Ollivier-Gooch and Michael Van Altena. A high-order-accurate unstructured mesh finite-volume scheme for the advectiondiffusion equation. *Journal of Computational Physics*, 181(2):729 – 752, 2002.

[18] Duarte M. S. Albuquerque. Numerical computation of incompressible flows on adaptive and unstructured grids. *PhD Thesis, Instituto Superior Técnico, Universidade de Lisboa*, 2013.