Article

Preconditioning the quad dominant mesh generator for ship structural analysis

Luka Grubišić [1,‡*], Domagoj Lacmanović [2,‡] and Josip Tambača [3,‡]

[1] University of Zagreb Faculty of Science; luka.grubisic@math.hr
[2] University of Zagreb Faculty of Science; domagoj.lacmanovic@math.hr
[3] University of Zagreb Faculty of Science; josip.tambaca@math.hr
* Correspondence: luka.grubisic@math.hr
‡ These authors contributed equally to this work.

Abstract: This paper presents an algorithm for the fully automatic mesh generation for the finite element analysis of ships and offshore structures. The quality requirements on the mesh generator are imposed by the acceptance criteria of the classification societies as well as the need to avoid shear locking when using low degree shell elements. The meshing algorithm will be generating quadrilateral dominated meshes (consisting of quads and triangles) and the mesh quality requirements mandate that quadrilaterals with internal angles close to 90° are to be preferred. The geometry is described by a dictionary containing points, rods, surfaces and openings. The first part of the proposed method consist of an algorithm to automatically clean the geometry. The corrected geometry is then meshed by the frontal Delaunay mesh generator as implemented in the gmsh package. We present a heuristic method to precondition the cross field of the frontal quadrilateral mesher. Also the influence of the order in which the plates are meshed will be explored as a preconditioning step.

Keywords: mesh generation; quad dominated surface meshes; finite element analysis; shell elements;

1. Introduction

This paper is concerned with automatic mesh generation in the process of the structural analysis of ships and off-shore structures. The ship structural analysis by the finite element method is governed by the acceptance criteria of the classification societies [1]. One of the possibilities to describe an input geometry, for a large class of such structures, is by the use of a dictionary of elements. The dictionaries which we will be considering in this paper consist of points, rods, surfaces, and openings. The points in the dictionary describe positions where the loads (for the finite element analysis) are going to be applied or the measurements are going to be taken. Rods describe the panel stiffeners or pillars. The set of surfaces will consist of three types of entities: web surfaces (parallelograms with one dimension much smaller than the other), regular surfaces (convex quadrilaterals defined by four co-planar nodes) and warped surfaces (closed loop surface defined by four corners which need not be co-planar) [2,3]. The preconditioning method which we are going to describe extends and improves the meshing algorithm from [4]. The algorithm from [4], called `pyREMAKEmsh`, is designed for the dictionaries that do not contain the warped surfaces. The present algorithm is extended to include the warped surfaces and also to improve the quality of the mesh by further preconditioning of the geometry. Our algorithm is intended for full automatic meshing of details of ship structures (such as superstructures, decks, ...) according to the rules of classification societies.

In a typical situation a description of a geometry by a dictionary of elements contains many modeling inconsistencies such as overlapping plates or plates not in contact. Note that a stiffened panel as a constructive element is modeled by a plate (shell) in contact with a rod or a web surface. The constructed mesh should not contain any hanging nodes. However, it can happen, due to modeling inconsistency, that e.g. a stiffener or a web surface does not overlap with a panel (model by a regular surface) which it is suppose to stiffen. Removing such modeling irregularities is fundamental both for the use of meshing algorithms which require that the surfaces are described as closed loop surfaces as well as by the subsequent finite element discretization procedures which require that there are no hanging nodes in the mesh, [2,5]. The algorithm from [4] utilized boolean operations (with a tolerance) from openCASCADE geometric kernel to solve this problem. The points in the dictionary represent locations where loads are applied or measurements are taken and so they need to be a part of the final mesh. This further complicates the meshing task. The requirements on a mesher can be summed up as:

- The elements with angles less than 45° and more than 135° should be avoided
- In the high stress areas such as e.g. web surfaces, the use of triangular elements should be avoided.
- The use of triangular shell elements should be kept to the minimum
- Quadrilateral shell elements with high aspect ratio as well as distorted elements should be avoided - aspect ratio for quadrilaterals is to be kept close to 1 but should not exceed 3 for 4 node elements and should not exceed 5 for 3 node elements (triangles).
- Web surfaces should be modeled with at least four elements allong the shorter dimension, and ideally with precisely four elements.

These requirements pose a challenge for implementing fully automatic meshing routines. Namely, these rules most often pose local linear constraints on the mesh and as such clash with the geometry optimization algorithms which search for globally optimal tessellations, see [5]. We relax those restrictions in order to accommodate more complex geometries typically encountered in ship structural analysis.

Let us briefly review the available results concerning fully automatic mesh generation for ship structural analysis. Meshes based on quadrilateral elements can either be generated directly by an advancing front algorithm, see [6] or by a hierarchical grid-based subdivision methods, [7]. In this paper we explore a third option, which the construction of a quadrilateral dominated mesh by a recombination of triangular mesh, [8,9]. An algorithm based on the preconditioned Blossom recombination approach [9] has been implemented in the `pyREMAKEmsh` algorithm presented in [4]. The method was designed for dictionaries which only contained co-planar plates. It was based on the control of the cross field, [10], using the insertion of virtual stiffeners. The virtual stiffeners are used to split a plate with which they intersect, but are not included in the dictionary as 1D elements.

In this approach it is necessary that the original triangular mesh be generated so that a recombination yields a regular quadrilateral mesh. Quadrilateral elements in `gmsh` are generated indirectly by the perfect matching algorithm based on the graph theoretic Blossom algorithm [9,11]. A default implementation starts from a standard Delaunay advancing front algorithm implemented in the L^2 metric. An important extension and improvement to the Blossom algorithm based quadrilateral mesh generation is realized by adapting the triangular advancing front mesher so that the generated triangular mesh can be efficiently matched into a regular quadrilateral mesh. It is achieved by modifying the advancing front Delaunay mesh generator with the change of norm from L^2 to the L^∞ norm. It is a geometric fact that an equilateral triangle in the L^∞ norm is in fact the right angle isosceles triangle, whereas an equilateral triangle in the L^2 (Euclidean) norm has internal angles equal to 60°, see [12]. With this change of norm the triangles which are generated by the Delaunay algorithm will tend to be closer to right angle triangles and so will be amenable to good recombination into a regular quadrilateral mesh, see [12].

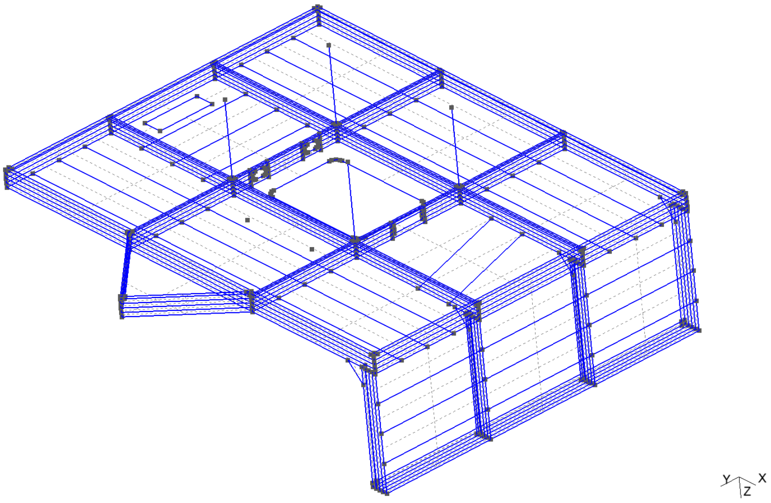


Figure 1. Input dictionary describing the geometry. It consist of points, rods, and co-planar plates (convex quadrilaterals which are a subset of a plane).

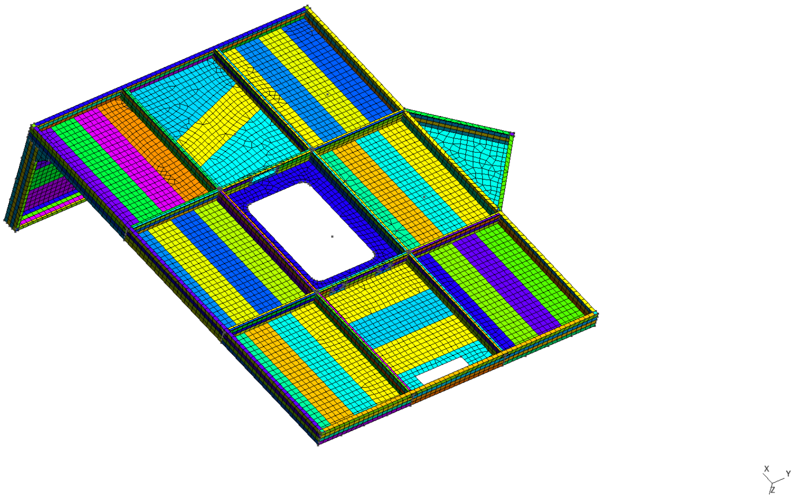


Figure 2. Meshing using the preconditioned gmsh algorithm Packing for Parallelograms from [13]

In this paper we will present a geometry improvement algorithm which can tackle geometries containing warped plates. The algorithm is based on the processing pipeline which utilizes boolean operations in the 3D geometry kernel OpenCASCADE to correct the geometry errors and to enforce local linear constraints. Further, we will introduce a preprocessing step to make the results of the meshing algorithm repeatable and to control the mesh quality.

The makeup of the paper is as follows. We will first present materials and methods where we will describe the challenges and the necessary modifications to the algorithm necessary to be able to tackle warped plates. We will then present results of the applications of these algorithm to two characteristic examples from engineering practice. In the discussion section we will present statistical evaluation of the preprocessing algorithm.

2. Materials and Methods

Many modern quadrilateral meshing algorithms are designed around the notion of the cross field, [10]. This is a heuristic function defined based on the geometry and it models the preferred orientation of the mesh. For quadrilateral mesh generating algorithm this is taken to be parallel to the boundaries of the domain. Frontal meshing algorithms propagate this cross field from the boundaries towards the interior of a surface. The new point is inserted in a mesh so that it forms an equilateral triangle with two nearest previously inserted nodes. In the quad oriented algorithm, one uses the L^∞ norm, and so the equilateral triangle is actually isosceles with an angle close to the right-angle. These triangles can then be optimally matched into quadrilaterals using the graph theoretic approach which minimizes the measure of the departure of a quad (a pair of triangles) from regularity. Furthermore, the unmatched triangle will be pushed by the marching front to the boundary. In what follows we will describe the basic properties of the algorithm and our method of imposing further line restriction on the mesh by the addition of the virtual stiffeners (rods) to the model. Let us note that full recombination of the mesh is not always desirable since it might lead to the reduced quality of the quadrilaterals. The Blossom and the simple recombination algorithms will leave some triangles in the mesh in the case when recombining all the triangles would lead to low quality quads (either by the aspect ratio or angle criterion). In such cases, to generate full-quad meshes the full-quad recombination algorithm is used. The algorithm performs a subdivision of the elements, followed by the further recombination and smoothing. The smoothing is performed using the Lloyd's algorithm, and then the whole sequence is repeated until all of the triangles have been matched [13]. We will employ the full recombination algorithm only in the case when the returned mixed mesh has more than 5% triangles. Even then, we will return both meshes and raise the flag that the processing pipeline did not produce a mesh which meets all of the constraints. We will see that the fully recombined mesh forces the satisfaction of requirement for almost no triangles at the expense of reducing the quality of the quadrilateral mesh.

To this end, let us rigorously define the mesh quality measures which we will be using to check the compliance with the rules of the classification societies. For a quadrilateral q with internal angles $\alpha_i(q)$, $i = 1, \cdots, 4$ we define the quality measure

$$\eta(q) = \max\left[1 - \frac{2}{\pi}\max_i\left[\left|\frac{\pi}{2} - \alpha_i(q)\right|\right], 0\right].$$

This quality measure is equal to one for the perfect rectangle and it is zero in the presence of angles which are less than zero or greater than the straight angle (a non-convex quadrangle). Let now $\mathcal{T}$ be the mixed tesselation consisting of quadrilaterals and triangles. The set $\mathcal{T}$ is a disjoint union of the set of all quadrilaterals $\mathcal{T}_q$ and the set of all triangles $\mathcal{T}_t$ in the mesh. To define the stopping criterion we we introduce the following measures. Here $|\cdot|$ marks the cardinality of a finite set. The average η for the tesselation $\mathcal{T}$ is defined by

$$\overline{\eta}(\mathcal{T}) = \frac{100}{|\mathcal{T}_q|}\sum_{q\in\mathcal{T}_q}\eta(q).$$

The percentage of quadrilaterals with angles between 80° and 100° in $\mathcal{T}$ is marked with

$$\xi(\mathcal{T}) = 100\frac{|\{q \in \mathcal{T}_q \ : \ 80^\circ \leq \alpha_i(q) \leq 100^\circ, i = 1, 2, 3, 4\}|}{|\mathcal{T}|}$$

and the measure of the portion of the triangles in the mesh is defined as $\tau(\mathcal{T}) = 100(1 - \frac{|\mathcal{T}_t|}{|\mathcal{T}|})$. Finally, we define the quantity $\delta(\mathcal{T})$ as the percentage of irregular elements. Those are quadrilaterals with aspect ration larger than 3, quads with at least one angle less than 10° and more than 170°. Distorted triangles are triangles with aspect ratio 1 : 5 or more or with one internal angle less than 10° and more than 170°.

The controls which we use to improve the control the number of nodes used to resolve circular boundaries, and the resolution of the web surface. A hard constraint on the number of elements per smaller dimension of the web surface is at least four. To enforce this constraint we split each of the web surfaces along the shorter dimension into four strips. We use the integer vector $\mu = (n_1, n_2, n_3, n_4)$ to denote the rule with which a

doi:10.20944/preprints202112.0279.v1

web surface is split into strips. As a result the web surface will be split into strips with height ratio $n_1 : n_2 : n_3 : n_4$.

Since boolean operations are not defined for spline surfaces, the warped elements are appended at the end of the list of surfaces. These are to be distinguished from the web surfaces which are defined as regular parallelograms and are meshed first. Further, it has been noted that the resulting mesh does depend on the order in which the surfaces have been meshed. To this end we sort the regular surfaces, after boolean operations to correct the geometry have been performed, according to the descending $\eta(\cdot)$.

Algorithm 1 Preconditioning the Packing for Parallelograms algorithm from [13]

```
Require: A dictionary describing the geometry and the tolerance tol.
Split each web surface without openings along the smaller dimension into 4 surfaces ($\mu = (1,1,1,1)$).
Split each web surface with at least one opening along the smaller dimension in the four strips with the ratio $n_1 : n_2 : n_3 : n_4$ so that all openings are contained in the middle two strips.
Define the openings as surfaces and introduce them into geometry using boolean operations from the CAD kernel.
Subdivide plates which are co-planar with rods.
Index surfaces so that a quadrilateral $q$ with larger $\eta(q)$ comes before the one with the smaller.
Index warped surfaces so that they are appended at the and of the list of surfaces.
Introduce Virtual Stiffeners around openings on web surfaces
Generate the mesh $\mathcal{T}$ of the geometry $G$ using Packing for Parallelograms with the simple recombination algorithm.
Compute $\tau(\mathcal{T}_\mu)$
if $\tau(\mathcal{T}_\mu) \geq$ tol then
    Return $\mathcal{T}$ and the success flag.
else
    Return the mesh $\mathcal{T}$
    Return the fine mesh $\mathcal{T}_f$ obtained using the full recombination algorithm with refinement and smoothing.
    Return the fail flag.
end if
```

We compare the performance of an algorithm using this statistical criterion. Recall, there are hard constraints that the elements should have internal angles between 45° and 135° and that we are allowed up to 5% triangles in the mesh. We will relax this criterion in that we will measure the percentage of quadrilaterals in the mesh which satisfy this criterion. Also, we will extend this into the target function which accounts for the regular quadrilaterals (having the internal angles between 80° and 100°) and the overall percentage of the mesh covered by quadrilaterals.

3. Results

In this section we will present results of measuring the quality indicators for four geometries denoted by dictionaries with co-planar plates G_1 and G_2 and two dictionaries which also contain the warped plates W_1 and W_2. The geometry G_1 is presented on Figure 1, whereas the geometries G_2, W_1 and W_2 are presented on the figures 3 and 4.

The results will be summarized in a tables. We will also present details of some of the geometries. for instance on the geometries from Figure 1 and 3 we will be particularly concerned in the way in which the girders with holes have been resolved under the hard restriction of four elements per girder height. On the geometries with warped plates from Figure 4, we will concentrate on the effect of the cross-field – which will follow the warping of the plates – on the regularity of the mesh.

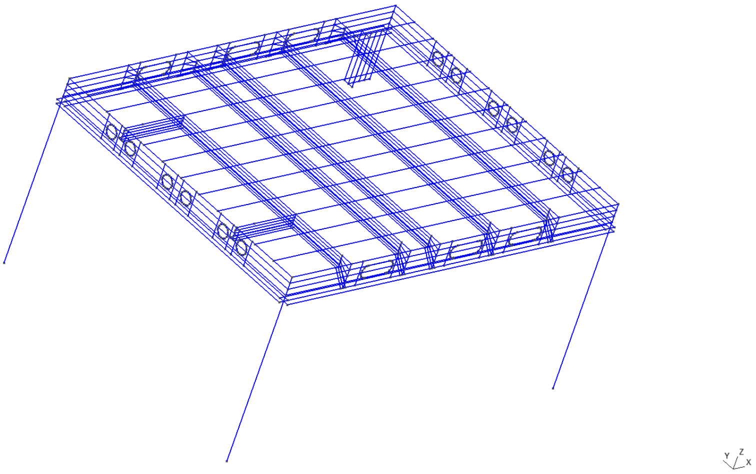
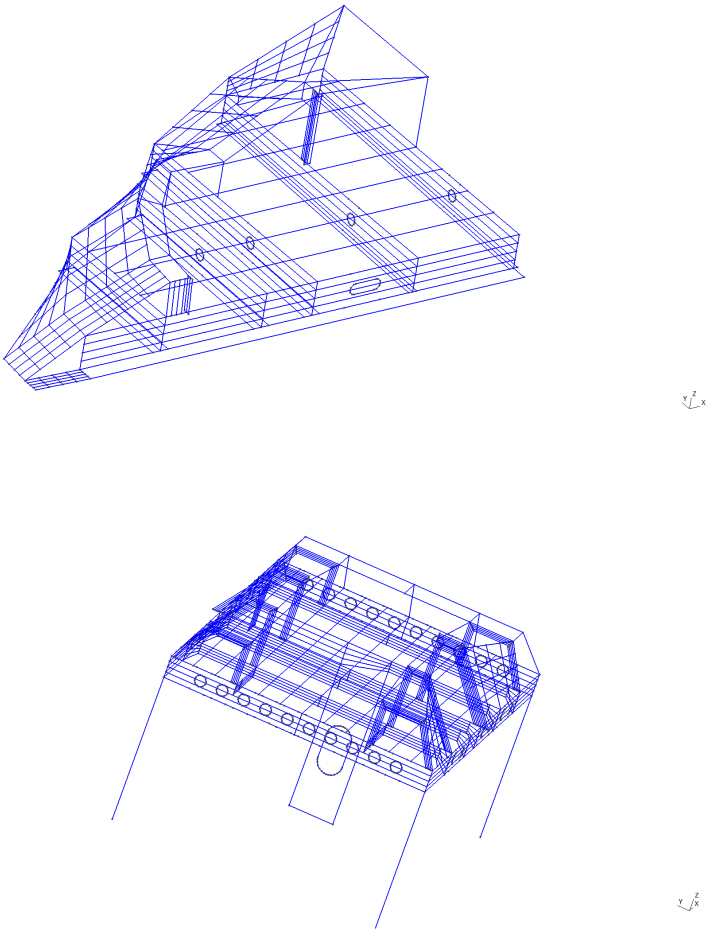


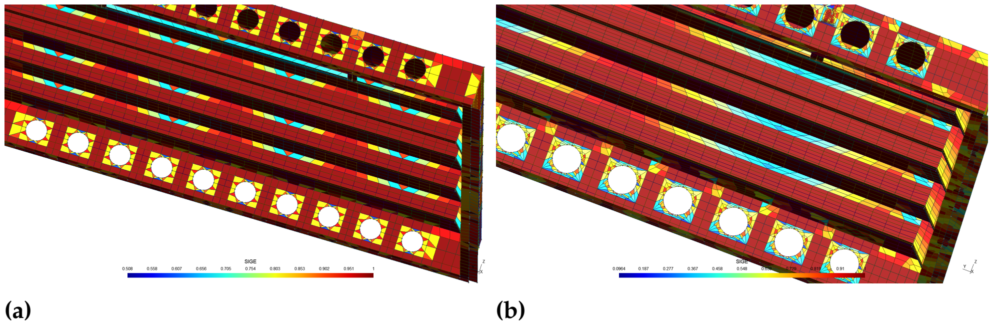
Figure 3. Geometry G_2



(a)

(b)

Figure 4. Input geometries with warped plates W_1 on figure(a) and W_2 on figure (b).



(a) (b)

Figure 5. A comparison of the meshes for a web surface with openings from the geometry G_2. We see that the full recombination algorithm produces an almost fully quadrilateral mesh, but the quality of the mesh, as measured by the SIGE indicator, for the full recombination algorithm is lower. In figure (a) we have the mesh generated by `pyREMAKmsh` and in Figure (b) the fully recombined mesh.

Table 1. Quality measures for the studied geometries. Algorithm: Preconditioned algorithm Packing for Parallelograms

Geometry dictionary	$\overline{\eta}(\mathcal{T})$	$\xi(\mathcal{T})$	$\tau(\mathcal{T})$	$\delta(\mathcal{T})$	$\lvert\mathcal{T}\rvert$
G_1	93	80	96	1.77	11,916
G_2	91	67	90	2.11	6,549
W_1	91	61	89	1.7	5,405
W_2	88	61	91	6.17	11,683

As a comparison we present, exemplary, the results for the mesh produced by applying the full recombination algorithm. Our experiments on the five industry relevant geometries of varying complexities show that the number of triangles in the mesh is much smaller, but the number of regular quadrilaterals is reduced significantly and the overall number of elements can be almost double (see Tables 1 and 2). On Figure 5-(b) we see that even when the full recombination algorithm is used, preconditioning the geometry by virtual stiffeners and adapting the splitting of the web surface so that openings are in the two central strips limits the propagation of the effect of the smoothing algorithm which reduced the number of triangles in the full recombination approach.

Table 2. Quality measures for the studied geometries. Algorithm: Full recombination algorithm with smoothing.

Geometry dictionary	$\overline{\eta}(\mathcal{T})$	$\xi(\mathcal{T})$	$\tau(\mathcal{T})$	$\delta(\mathcal{T})$	$\lvert\mathcal{T}\rvert$
G_1	86	67	99.8	15.45	19,489
G_2	78	52	99.9	31.54	13,386
W_1	82	49	99.5	8.05	10,504
W_2	79	46	100	24.78	21,848

Let us now compare the meshes generated by the preconditioned Packing for Parallelograms algorithm followed by a simple recombination as implemented in `pyREMAKEmsh` from [4] with the mesh obtained using the full recombination algorithm with smoothing, [13,14]. We are comparing the meshes of the web surface with openings using the SIGE (signed inverse gradient error for the finite element solution) indicator from `gmsh`, [3]. Elements marked by the lower SIGE indicator (color code from blue to green) do not meet the SIGE quality criterion. The elements marked in red are on the other hand of the highest quality.

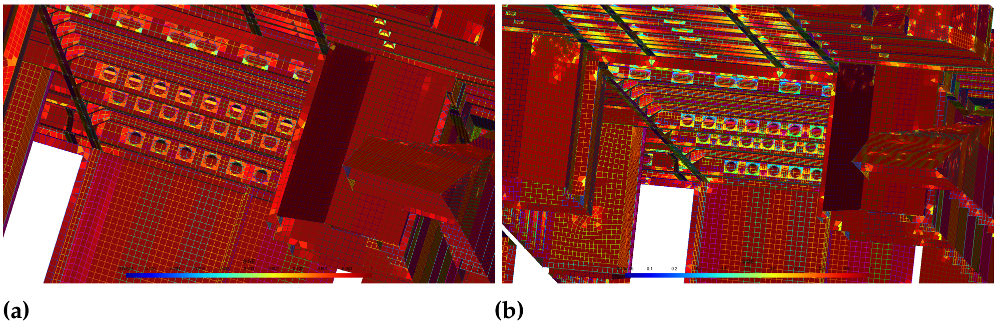


Figure 6. A comparison of the meshing algorithm on an example of a superstructure of a large yacht (geometry G_3). The detail of the web surfaces is compared using the SIGE indicator. In figure (a) we have the mesh generated by `pyREMAKmsh` and in Figure (b) the fully recombined mesh.

4. Discussion and a conclusion

Let us now consider a more involved geometry describing a superstructure of a large yacht. This geometry will be denoted by G_3 and is depicted on Figures A1 and A2. The generated mesh will have more than hundred thousand elements and we report the quality measures for the preconditioned recombination and the full recombination algorithms.

Table 3. The mesh quality of the superstructure of a large yacht (geometry G_3).

Geometry dictionary G_3	$\overline{\eta}(\mathcal{T})$	$\xi(\mathcal{T})$	$\tau(\mathcal{T})$	$\delta(\mathcal{T})$	$\lvert\mathcal{T}\rvert$
`pyREMAKEmsh`	94	78	94	1.95	$128,053$
Full recombination	85	60	99.8	18.37	$232,720$

The geometry, together with a detail of the performance of two algorithms on the web surfaces is presented in the Figure 6.

We see that the `pyREMAKEmsh` algorithm typically achieves the quality restrictions as posed by the classification societies. It should be noted that on smaller size challenging geometries (containing web surfaces with many openings) it is possible that the preconditioned Packing for Parallelograms algorithm (as implemented in `pyREMAKEmsh` can have as many as 10% triangles (geometries G_2, W_1 and W_2). However, those triangles do not affect, as measured by the SIGE indicator, the high stress area adversely. On the other hand, the full recombination algorithm can achieve and almost exclusively quadrilateral mesh at the expense of a lower element quality. This can in particularly be seen in Table 2 where the number of degenerate elements can be as high as 20%. Furthermore, the SIGE criterion shows that these lower quality elements can be found in a high stress area such as web surfaces. Also, on an example of a large geometry (120,000 elements for the `pyREMAKEmsh` algorithm and almost twice as many for the full recombination algorithm) we see that `pyREMAKEmsh` can produce the meshes with as few as less than 5 % triangles.

Finally, let us not that the accuracy of the finite element solution for typical low degree shell elements can sometimes be reduced by more than 20% when elements whose internal angles are far away from the right angle are used. This is the reason why we have opted to return both a fully recombined mesh as well as a preconditioned quadrilateral mesh with the warning flag raised. It will be future work to perform a full empirical parametric study of the relationship between the lowest angle in the mesh and the accuracy of the finite element approximation for various types of appropriate shell elements. Also, a study of different (non standard) shell elements which might have a lesser dependence on the geometry of the mesh might prove to be a viable (albeit nonstandard) alternative.

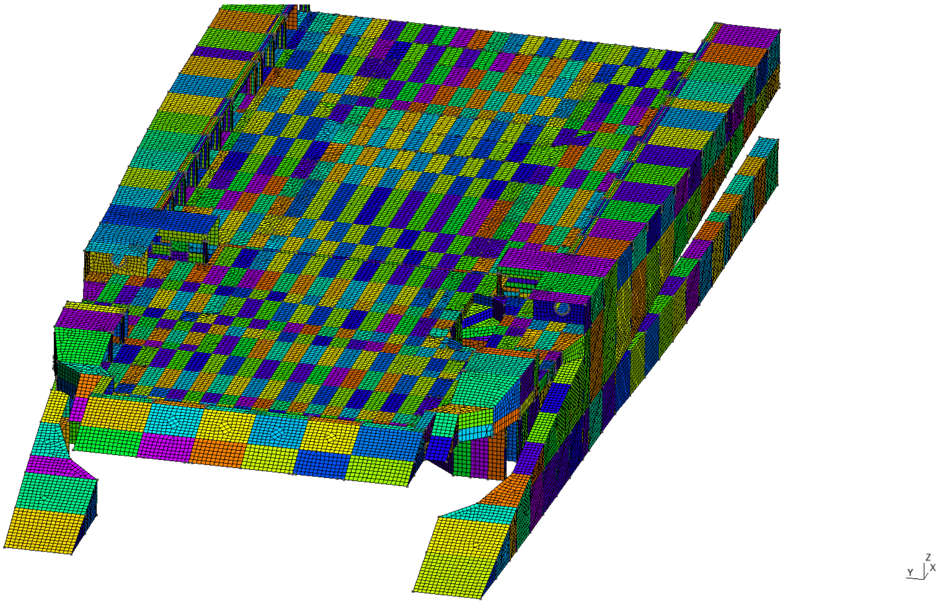


Figure A1. A `pyREMAKEmsh` mesh for G_3.

Author Contributions: Conceptualization, Luka Grubisic and Josip Tambača; Data curation, Domagoj Lacmanović; Funding acquisition, Luka Grubisic; Methodology, Luka Grubisic, Domagoj Lacmanović and Josip Tambača; Software, Domagoj Lacmanović; Supervision, Luka Grubisic and Josip Tambača; Validation, Luka Grubisic and Josip Tambača; Visualization, Domagoj Lacmanović; Writing – original draft, Luka Grubisic; Writing – review & editing, Luka Grubisic, Domagoj Lacmanović and Josip Tambača. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported in part by EU Regional Development Funds under the project: KK.01.2.1.01.0124 - Development of efficient methodology for finite element method based structural analysis of marine structures – REMAKE.

Data Availability Statement: he data which was used for testing purposes as well as the code is available from github `https://github.com/PMF-ZNMZR/pyREMAKEmsh`.

Acknowledgments: The authors would like to thank the REMAKE project coordinator Bonum d.o.o team for help during preparation of an automatic meshing module and for providing the geometrical models which are relevent from the engineering perspective.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SIGE	signed inverse gradient error for the finite element solution
web surface	a parallelogram with one dimension much smaller than the other
regular surface	quadrilateral defined by four co-planar corners
warped surfaces	generalized quadrilateral defined by four not co-planar corners

Appendix A

We present on Figure A1 the full mesh of the geometry G_3 as produced by the `pyREMAKmsh` algorithm and the mesh produced by the full recombination algorithm on Figure A2

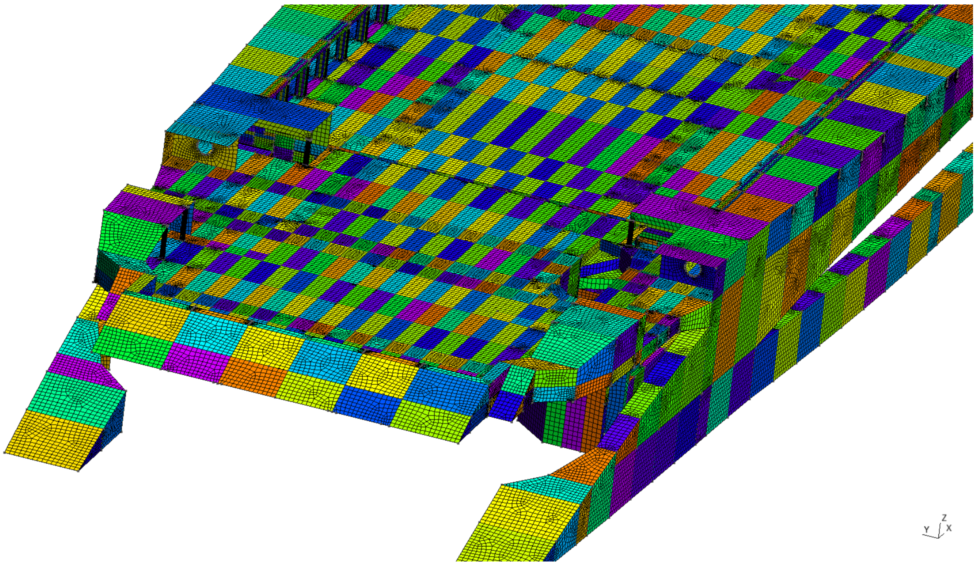


Figure A2. A full recombination mesh for G_3.

References

1. Nersesian, R.; Mahmood, S. International association of classification societies **2009**. pp. 765–774. doi:10.1163/ej.9789004163300.i-1081.675.
2. Hughes, O.; Paik, J.K. *Ship Structural Analysis and Design*; The Society of Naval Architects and Marine Engineers: New Jersey, 2010.
3. Geuzaine, C.; Remacle, J.F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* **2009**, *79*, 1309–1331, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2579]. doi: https://doi.org/10.1002/nme.2579.
4. Grubišić, L.; Lacmanović, D.; Tambača, J. Automatic mesh generation for structural analysis in naval architecture. *Proceedings of the International Conference on Ships and Offshore Structures ICSOS 2021.* **2021**, pp. 10, Paper 13.
5. Jang, B.S.; Suh, Y.S.; Kim, E.K.; Lee, T.H. Automatic FE modeler using stiffener-based mesh generation algorithm for ship structural analysis. *Marine Structures* **2008**, *21*, 294–325. doi: https://doi.org/10.1016/j.marstruc.2007.08.001.
6. Blacker, T.D.; Stephenson, M.B. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* **1991**, *32*, 811–847, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620320410]. doi: https://doi.org/10.1002/nme.1620320410.
7. Baehmann, P.L.; Wittchen, S.L.; Shephard, M.S.; Grice, K.R.; Yerry, M.A. Robust, geometrically based, automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering* **1987**, *24*, 1043–1078.
8. Pellenard, B.; Orbay, G.; Chen, J.; Sohan, S.; Kwok, W.; Tristano, J.R. QMCF: QMorph Cross Field-driven Quad-dominant Meshing Algorithm. *Procedia Engineering* **2014**, *82*, 338–350. 23rd International Meshing Roundtable (IMR23), doi:https://doi.org/10.1016/j.proeng.2014.10.395.
9. Remacle, J.F.; Lambrechts, J.; Seny, B.; Marchandise, E.; Johnen, A.; Geuzainet, C. Blossom-Quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. *International Journal for Numerical Methods in Engineering* **2012**, *89*, 1102 – 1119. doi:10.1002/nme.3279.
10. Lévy, B.; Liu, Y. Lp Centroidal Voronoi Tessellation and Its Applications. *ACM Trans. Graph.* **2010**, *29*. doi:10.1145/1778765.1778856.
11. Edmonds, J. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics* **1965**, p. 125.
12. Remacle, J.F.; Henrotte, F.; Carrier-Baudouin, T.; Béchet, E.; Marchandise, E.; Geuzaine, C.; Mouton, T. A frontal Delaunay quad mesh generator using the L^∞ norm. *International Journal for Numerical Methods in Engineering* **2013**, *94*, 494–512.

13. Baudouin, T.C.; Remacle, J.F.; Marchandise, E.; Henrotte, F.; Geuzaine, C. A frontal approach to hex-dominant mesh generation. *Advanced Modeling and Simulation in Engineering Sciences* **2014**, *1*, 8. doi:10.1186/2213-7467-1-8.

14. Du, Q.; Faber, V.; Gunzburger, M. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review* **1999**, *41*, 637–676, [https://doi.org/10.1137/S0036144599352836]. doi:10.1137/S0036144599352836.