Automatic mesh generation for structural analysis in naval architecture

Luka Grubišić[a], Domagoj Lacmanović[a], Pero Prebeg[b], Josip Tambača[a]

University of Zagreb, Faculty of Science, 10000 Zagreb, Croatia
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, 10000 Zagreb, Croatia

Abstract

We present a frontal, quad based, Delaunay meshing algorithm adapted to the structures appearing in the ship structural analysis. The main contribution of the paper is the preprocessing algorithm which resolves the geometry of the object and divides it into plates where all of the intersections of the boundary edges are guaranteed to be the nodes of any mesh generated for such a geometry. The result of this preprocessing algorithm is a discrete description of the geometry which can be meshed by the automatic frontal mesher according to the requirements of the classification society. Further algorithmic solutions for the inclusion of openings in the plates are also presented. The resulting algorithm is wholly implemented in python, using open source gmsh system together with the opencad module. It is capable of generating as many as ten thousand quads per second. This is the efficiency comparable to the standard triangular mesh algorithms and is due to the efficient implementation of the Delaunay quad algorithm in gmsh. We present several models from the engineering praxis to illustrate our claims.

Keywords: mesh generation; finite element analysis; low order shell elements;

1. Introduction

In this paper we are concerned with the automatic mesh generation for the structural analysis of marine structures. According to the acceptance criteria by the classification societies [1, 12] the rules for the finite element analysis include restrictions on the type of elements allowed as well as rules for generating and assessing the quality of the wire-frame model (mesh). For instance, a mesh should mostly consist of quadrilaterals close to a square. Up to 5% triangles are allowed and the angles of all elements should be between 45° and 135°. The preferred elements to be used are dominantly low degree shell elements. Such elements are sensitive to locking issues which necessitate the use of quadrilateral meshes which mostly consist of squares or almost squares. We have summarized the requirements on the mesh in Appendix A (coarse mesh description is in Appendix A.1. , and the specifications for the fine mesh regions are in Appendix A.2.).

Meshes based on quadrilateral elements can either be generated directly by some sort of advancing front algorithm such as [3] or by a subdivision of a region in quadrilaterals using a regular grid-based methods (quadtrees), [4]. Alternatively, a quadrilateral mesh can be generated indirectly by a recombination of a triangular mesh. However, 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 [5,6]. 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 mesh generator so that the generated triangular mesh can be efficiently recombined 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. In this way the triangles which are generated by the Delaunay algorithm will tend to be closer to right angle triangles than in the L^2 case where the algorithm tends towards equilateral triangles, e.g. with angles closer to 60°, see [7].

The use of triangle recombination based quadrilateral mesh generator is not new in marine engineering. A notable example is the utilization of the `qmorph` algorithm in [8,9]. However, it was observed in [2] that such meshes can include lower quality quadrilaterals in a way which is not easy to control. The method proposed in [2], called Stiffener-Based Mesh Generation, achieved the control of the quadrilateral meshes by subdividing the parts of the geometry adding extra stiffeners. A further improvement of the mesh was achieved by proactively modifying the geometry, e.g. by removing challenging parts of the geometry, in the procedure called *idealization*.

Our approach to the problem of generating high quality quadrilateral meshes for the structural analysis of marine structures is bases on the L^∞ advancing front mesh generation from [7] in a combination with a minimally aggressive geometric preconditioning step. This preconditioning is primarily aimed to implement the restrictions posed by the rules of the classification societies restricting mesh generation, such as requiring at least four elements on a girder, and restricting angles, in almost all cases, to the range between 45° and 135°. Also, as a consequence of the utilization of the modeling software there are frequent instances of overlapping plates which need to be idealized.

Our general approach to idealization was that the definition of the geometry should be respected as much as possible. The holes in the plates were localized by virtual stiffeners primarily for the purpose of enforcing the preference for quadrilaterals which have angles close to 90° and have sides parallel to boundaries of a plate in which they are punctured. In general, Blossom algorithm produces a quadrilateral mesh form any triangular mesh with even number of the triangles. Furthermore, the marching front quadrilateral Delaunay mesh generator produces triangles close to the right-angle triangle and so typically compensates inexactness in the definition of the surface, even in challenging situations, with at most a couple of acute angle triangles.

The makeup of the paper is as follows. We will first present materials and methods where we will describe the virtual stiffener algorithm for the control of the quadrilateral meshes generated by the algorithms from [5,7]. We will then present results of the applications of these algorithms to two characteristic examples from practical modeling. As a first example we consider three pontoons and in the second we consider a section of the deck and a bulkhead. In the discussion section we will present statistical evaluation of the proposed algorithm. Finally, we will draw some conclusions about the use of open-source meshing software in naval architecture.

2. **Materials and methods**

Let us note the following geometrical observation from [7, Section 4.3.]. An equilateral triangle in the L^∞ norm is isosceles in the L^2 norm. In fact, an equilateral triangle in the L^∞ norm is half of the square. The marching front L^∞ Delaunay algorithm generates a marching front of L^∞ equilateral triangles, e.g. having angles close to the right angle, and pushes – by design – degenerate triangles to the boundary of the region to be tessellated. A subsequent application of the perfect matching algorithm from [5] yields a regular quadrilateral mesh with all of the degenerate triangles pushed to the boundary.

Modern quadrilateral meshing algorithms are designed around the notion of the cross field, [11]. This is a heuristic function defined based on the geometry and it models locally the preferred orientation of the mesh. For quadrilateral mesh generating algorithm this is taken to be parallel to the boundaries of the domain. To compute the desired orientation of the mesh in a given node, the frontal meshing algorithms propagate this cross-field information from the boundaries and insert a point in the mesh so as to form an equilateral triangle. In the quad oriented algorithm, one uses the Delaunay approach in 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 to the model.

Let us define the quadrilateral mesh quality measures. For a quadrilateral q with internal angles $\alpha_i(q),\ i = 1,2,3,4$ we define the quality measure

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

This quality measure is equal to one for the perfect rectangle and it is zero in the presence of angles which are greater than the straight angle.

Let $G(V, E, c)$ be an undirected weighted graph. Here V is the set of n_V vertices and E is the set of n_E undirected edges and $c(E) = \sum c(e_{ij})$ is an edge-based cost function. A matching is a subset E' of the set of edges such that a node V has at most one incident edge in E'. A matching is called perfect if each node of V has exactly one incident edge in E'. As a consequence, a perfect matching has exactly $n_{E'} = n_V/2$ and can be only found in graphs with an even number of vertices. A matching is optimal if $c(E')$ is minimal among all possible perfect matchings.

In the case of triangle recombination, the graph $G(V, E, c)$ is defined by taking V to be the set of triangles and E to include those triangle pairs having a common edge. The matching is perfect if all triangles have been matched in quadrilaterals and it is optimal if

$$c(E') = \sum_{(e_i,e_j)\in E'} (1 - \eta(q_{ij})) \tag{2}$$

is minimal over all subsets E'. Here q_{ij} is the quadrilateral obtained by merging a triangle pair (e_i, e_j) from E'.

2.1. Automatic mesh generation based on virtual stiffeners

We will summarize the algorithm which we implemented in the following flow chart. On a high level, the algorithm consists of three parts. First, we preprocess the geometry to remove the overlapping surfaces. We further subdivide the surfaces along the stiffeners and ensure that all of the nodes on the stiffeners are nodes in the mesh. This step is implemented using the `pygmsh` library to access the highly optimized Boolean operations of the Open CASCADE 3D modelling kernel. Second, we introduce virtual stiffeners to enforce the cross-field control on the quadrilateral elements. The virtual stiffeners are not bona-fide one dimensional elements. They are not returned in the dictionary of one-dimensional elements and are only introduced to control the properties of the mesh around the holes in the structure and force the splitting of a surface in two surfaces with the desired cross field.

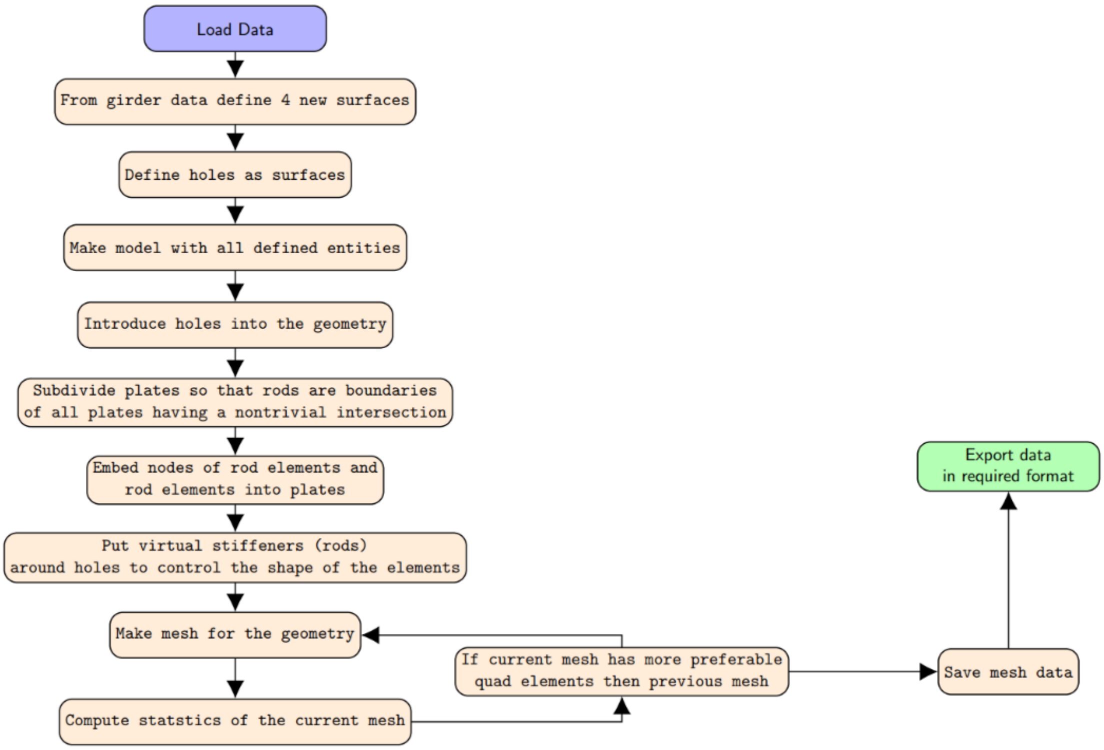

Fig. 1. Flow chart of the automatic mesh algorithm

The newly generated surfaces inherit all of the properties of the parent surface. As the last step we carry out a sequence of boundary-controlled refinements with the aim of determining the mesh with the lowermost number of nodes, in the sequence of refinements, with the highest percentage of regular quadrilaterals. For this part of the algorithm, we have used the python interface to the `gmsh` library of meshing routines. This allowed us to implement our algorithm based on the well tested and performance optimized routines – the Blossom perfect matching algorithm and the marching front Delaunay mesh generator. The implementation is described with the flow-chart on Figure 1.

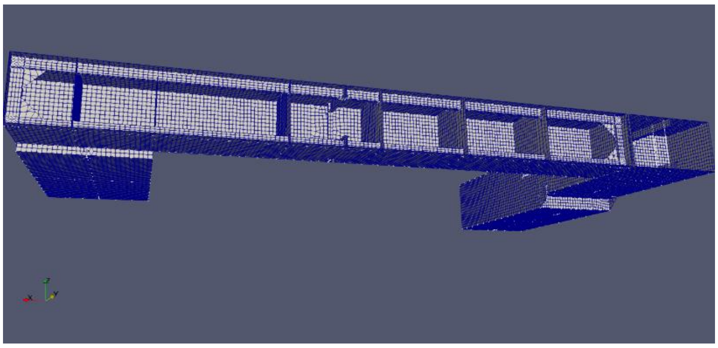

Fig. 2. A configuration of three pontoons.

3. Results

We will utilize two case studies for comparing the algorithms which we discussed in this note. The challenge in all of the models is to incorporate the line constraints in the mesh (e.g. stiffeners or girders intersecting with incommensurate dimensions), error in modeling as well as to generate a mesh which is mostly regular quadrilateral mesh. Further challenges are placed by holes in the plates as well as additional 1D elements – the stiffeners – which make restrictions on the quality of the mesh which can be achieved.

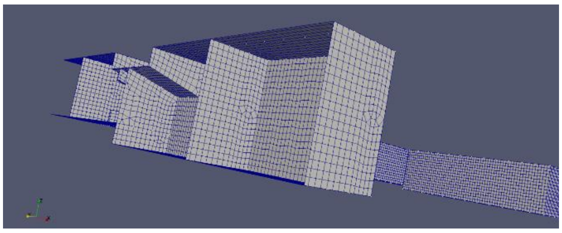

Fig. 3. A cut of the final finite element model of three pontoons.

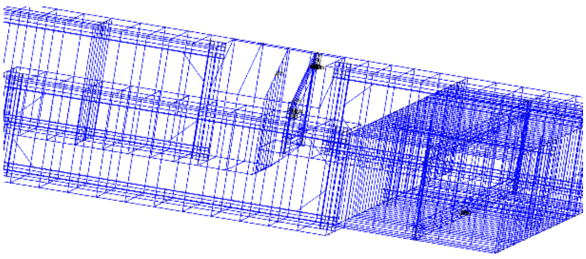

Fig. 4. Visualization of the location of the elements which do not meet the desired restrictions in the wire-frame model.

On figure 4. we see the location of a section where two girders pierce each other causing an undesired overlap. The algorithm resolves this overlap by ensuring that all boundaries of structural elements (surfaces) are edges of elements in mesh. Subsequently the moving front algorithm deals with this problem by putting one degenerate triangle and thus marks the scope and the location of the problem. On Figure 5. we see a zoom in onto the configuration of two girders piercing each other. We also see the marked triangles which need to be removed in the idealization procedure. The decision is however entirely placed at the user's responsibility.

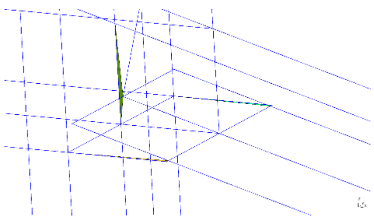

Fig 5. A zoom in on a description of the surfaces of two girders which are puncturing each other.

3.1. Should we idealize?

In Figure 4 and Figure 5 we see the locations of the triangles which do not meet the desired prescriptions on the mesh geometry. In most cases this was the consequence of the geometry description error. For instance, in Figure 5 we see two girders puncturing each other. This geometry error results in only three degenerate triangles. The rest of the geometry -- including the undesired overlap -- is meshed with regular quadrilaterals. We opt to leave these triangles in the mesh. A geometry idealization procedure, similar to the one described in [2], would simply modify the geometry by removing these sections. However, the right change in the model would have been to shorten one of the girders so that they are no longer puncturing each other. Important feature of a moving front mesh generator is that this situation will be detected by a presence of a single degenerate triangle (per intersection) and it is envisaged that a user modifies the geometry at these marked paces (rather than to have to search for them). The task of the algorithm is to mark the places which should be idealized. An automatic idealization algorithm, even though desirable, seems to be too aggressive in comparison to this human in the loop procedure. We have limited our approach to virtual stiffener-based mesh control solely to the cases of holes in the girder and the need to enforce right angle preference on the quadrilaterals. On Figure 3 we show the cut of the final model form the far side in relation to the overlapping girders.

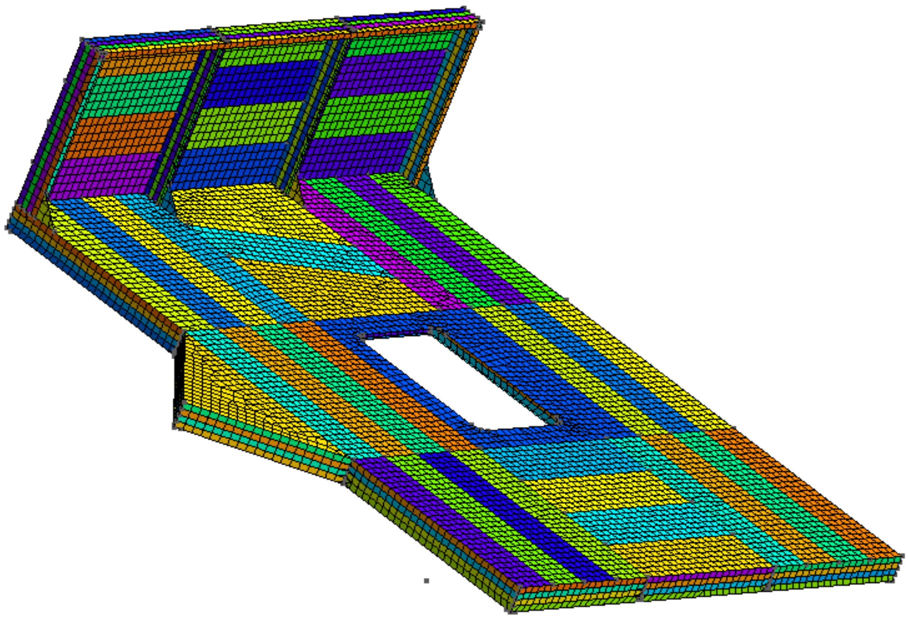

Fig 6. Section of a deck and a bulkhead.

3.2. Mesh refinement and stiffener-controlled refinement localization

The heavy lifting in this implementation is performed by the moving front Delaunay mesh generator. It is controlled by the notion of the cross-field function. Notice the interactions of the Delaunay fronts of quadrilaterals. This is to be contrasted with the hierarchical approach in Figure 12. The main challenges to scaling this approach is in dealing with incommensurate dimensions in the geometry which force the refinement of the mesh size. In general, there are two ways to deal with the challenges posed by unnecessarily fine details in the model. One is to isolate this portion of the mesh by placing virtual stiffeners to enforce the line restrictions, see Figure 8. In this setting we can opt to refine the isolated region according to the rules for the fine mesh Appendix B.

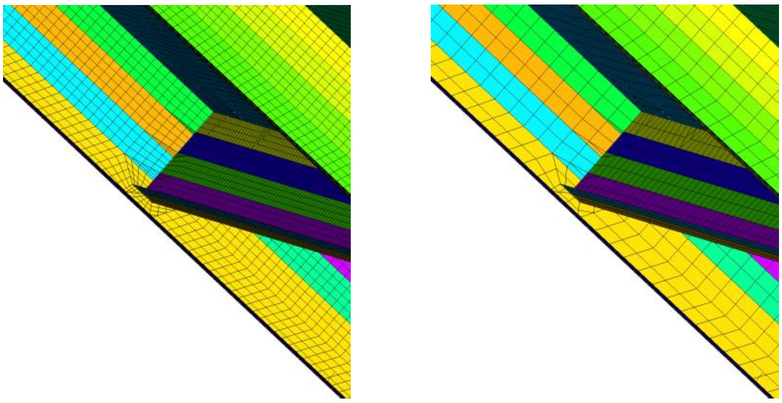

Fig 7. Refining around a contact of two girders.

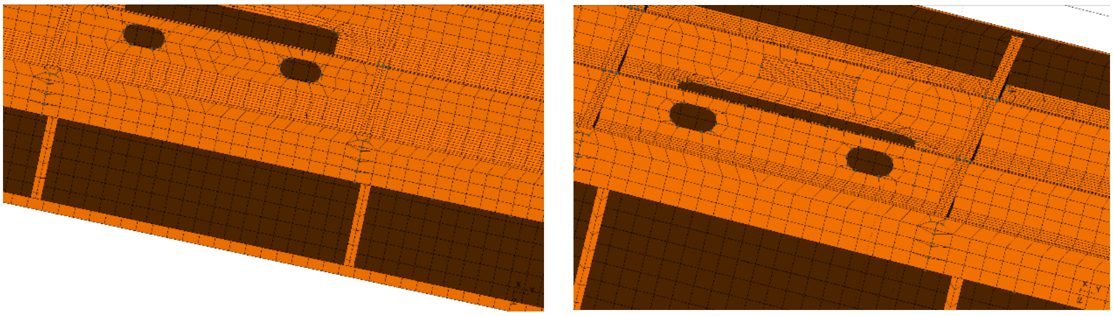

Fig 8. Controlling the meshing around a hole in a girder using virtual stiffeners.

An alternative is a geometry preconditioning approach based on virtual stiffeners. The virtual stiffeners enforce the line restriction on the cross-filed and allow for at most a couple of sub-optimal elements on the boundary. Rather than to follow the principles of idealization and delete these elements from the model, we leave them in the model and argue that their presence will not affect the solution more than would their removal.

3.3. Grading of a mesh

In this final example we would like to draw the readers attention to the gradual transition from the fine mesh to the standard regular four element discretization of a girder. The refinement was incurred by the contact of two incommensurate girders (Figure 9). In Figure 7 we see a consequence of a uniform refinement around the girders in contact. This indicates that placing a fine region around this point in the geometry could solve the problem of degenerate triangles. On the other hand, modeling this as a fine region and investing much more computational time to solve the global model does not seem to be justified just based on the motivation to avoid a couple of suboptimal triangles.

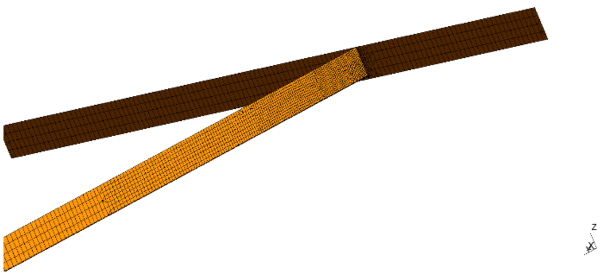

Fig. 9. Transition from a fine mesh to a coarse mesh (four elements per girder).

4. **Discussion**

We assess the quality of the mesh based on the statistics of the key performance indicators. On one hand this the value of the τ and η parameters and on the other this is the statistics for the number of elements. We see that there are 3% triangles in the mesh and that 84% of elements are regular quadrilaterals. Those are quadrilaterals with the internal angles between 80° and 100°. Further analysis shows that the percentage of the surface area covered by regular elements is 90% (see Figure 10 representing statistics for the finer mesh).

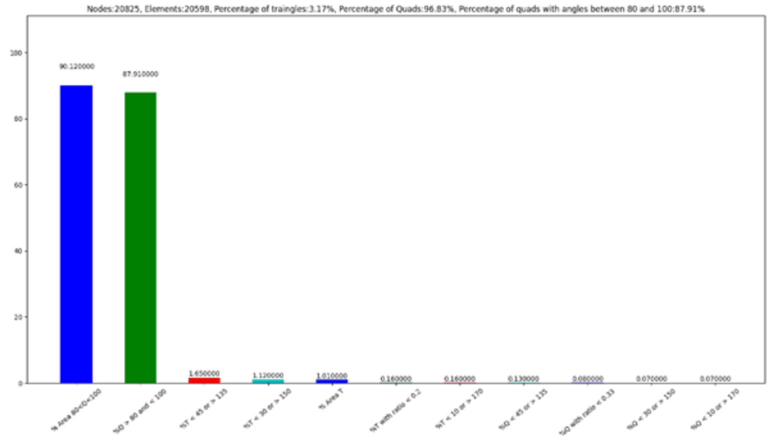

Fig. 10. Statistics for the refined mesh of the structure. The number of regular quadrilaterals has increased, but not for much. This indicates that the marching front algorithm correctly detected the challenging parts of the geometry, even at the level of a coarser mesh.

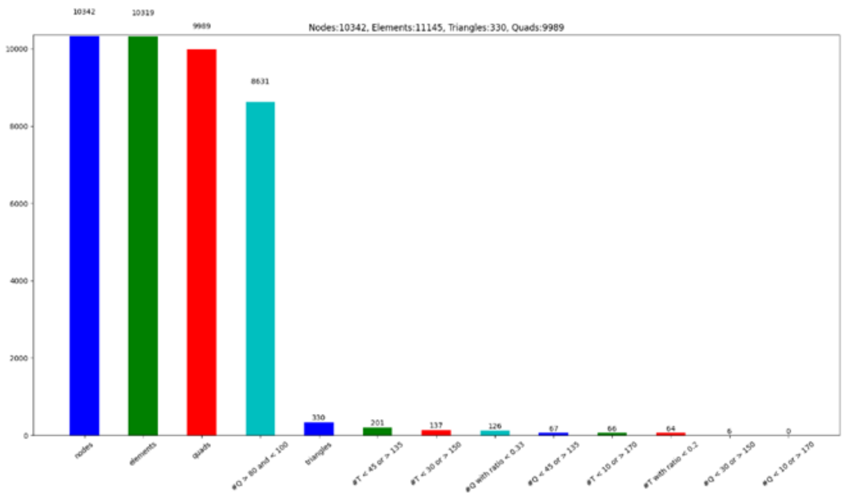

Fig. 11. Statistics for the coarse mesh

A more informative consideration than an assessment of the statistical report is to consider where the problematic elements have appeared. The marching front quadrilateral Delaunay algorithm has pushed suboptimal elements (mostly triangles) to the boundary edges. There thy could have been removed from the geometry in a modification of a geometric idealization procedure. This would incur the problems of keeping the mesh connected while changing the model (possibly contrary to the wishes of a modeler). We claim, on the other hand that it is better to leave the suboptimal elements on the boundaries of the model in the mesh. The perturbation which such elements cause to the stiffness matrix are such that they can be controlled by the modern linear system solvers. The potentially cause bad conditioning of the stiffness matrix, but their influence on the overall quality of the solution is controlled by a successful pivoting algorithm.

Finally, let us point out that a mesh generated by hand had 63% more degrees of freedom (at 18000 elements). Also, while generating the mesh by hand geometry is idealized by a modelers' intervention and so a general comparison of the quality of mesh is hard. Should we compare the original or idealized geometry? We claim that it is not possible to generate a mesh by hand which will mesh the original geometry and have less than 12000 elements. Note that the in the automatically generated mesh in Figure 7 we see clearly a line of the gradual transition between advancing fronts of the quadrilateral elements. A gradual transition is achieved by a continuous transformation of a quadrilateral from a square to a slightly distorted trapeze (having angles in the span 80° to 100°). In comparison the hand generated mesh in Figure 12 has sharper transitions (a couple of more degenerate quadrilaterals or triangles) which accumulate the change needed to compensate for the consistent use of squares wherever possible. The meshing algorithm completed the automatic meshing task in under 10 seconds, whereas generating the mesh by hand required hours of expert work. This is our main rational for proposing the current solution.

If the comparison is to be made on the solution of a finite element study (statical/vibrational) then the influence of the triangles pushed to the boundary of the model on the accuracy of the assembled stiffness matrix is comparable with a similar perturbation of the boundary and should be accepted. This will be the subject of a separate numerical linear algebra analysis and is outside the scope of this paper.

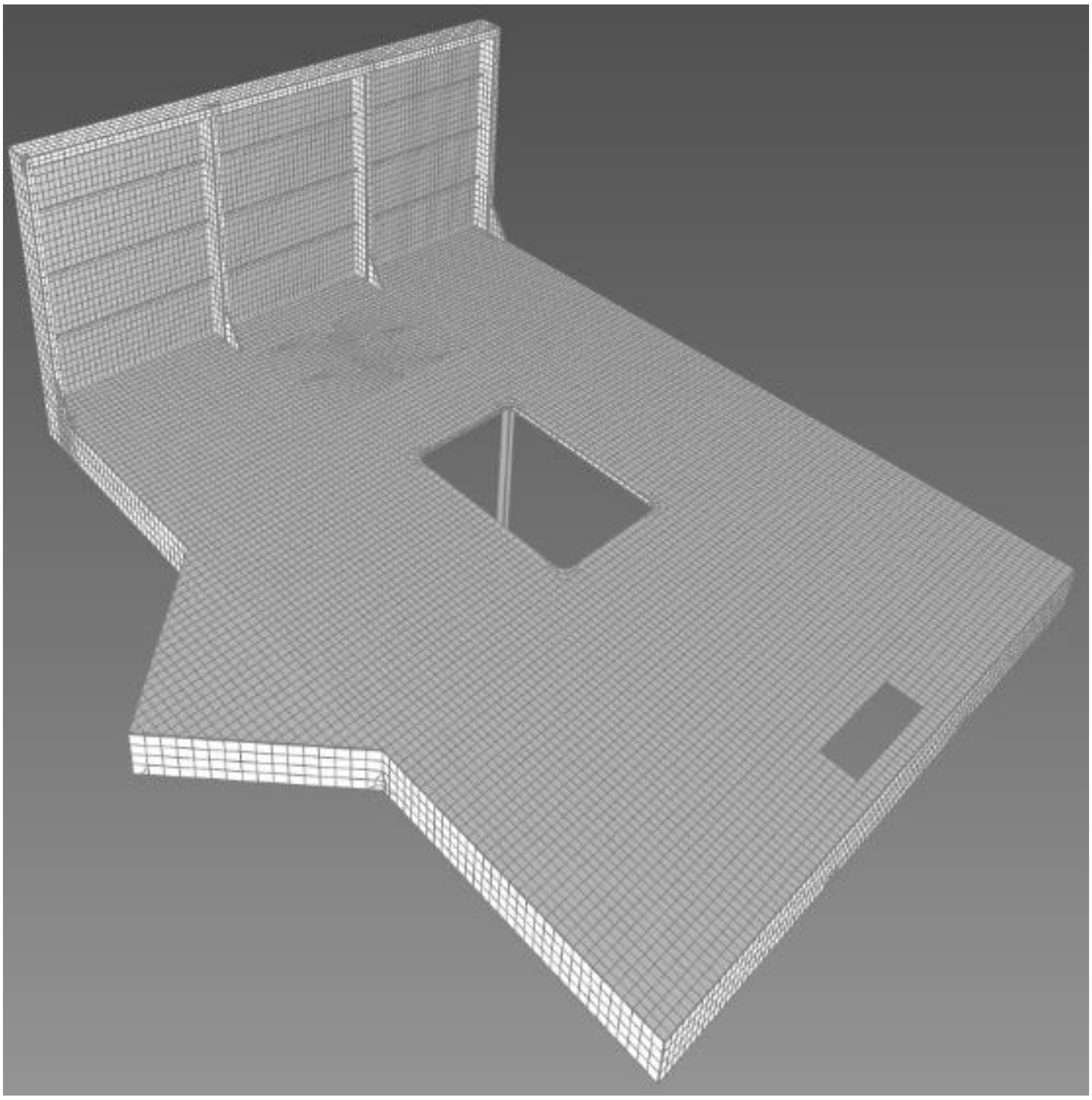

Fig. 12. Hand generated mesh.

Acknowledgements

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. Thanks are due to the project coordinator Bonum d.o.o team for help during preparation of an automatic meshing module.

Appendix A. Requirements on the mesh

The main challenge in the analysis of the marine structures comes from the use, and community's preference, of low degree shell elements. This poses constraints on the topological properties of the mesh which are sometimes contrary to the best practices which would follow from the general design of meshing procedures for the elasticity theory. We now provide the list of requirements, see [12].

A.1. Coarse mesh requirements

First, we will list the types of finite elements - used in linear finite element analysis:

- Rod (truss) element - line element with axial stiffness only and constant cross-sectional area along the length of the element
- Beam element - line element with axial, torsional and bi-directional shear and bending stiffness and with constant properties along the length of the element
- Shell (planar) surface element with constant thickness– with in-plane stiffness (membrane) and out-of-plane bending stiffness (plate)

Further the following guidelines should be met when modeling the structure:

- 2 node line elements and 4 or 3 node shell elements are sufficient for hull structure representation - mesh descriptions given in this CG are based on the assumption that these elements are used in the FE models (however, higher order elements may also be used)
- Quadrilateral shell elements - with inner angles below 45° or above 135° between edges should be avoided
- Quadrilateral shell elements with high aspect ratio as well as distorted elements should be avoided - aspect ratio is to be kept close to 1 but should not exceed 3 for 4 node elements and 5 for 8 node elements
- The use of triangular shell elements is to be kept to a minimum

In the area where high stresses are expected the aspect ratio of shell elements is to be kept close to 1 and the use of triangular elements is to be avoided. The shape functions of an element in such an area must include "incompatible modes" which offer improved bending behavior. Further, elements with "incompatible modes" are required particularly for the modeling of web plates in order to calculate bending stress distribution correctly with a single element over the full web height. For the global, partial ship and fine mesh strength analyses the assessment against stress acceptance criteria is normally based on membrane (in-plane) stresses. Singularities in membrane part of shell elements can be avoided by arranging so-called singularity trusses.

A.2. Fine mesh zone

For the fine mesh zone, the rules are slightly extended. In such a zone the following requirements are placed on the local mesh and its inclusion (transition) to the global coarse mesh.

- A uniform quadratic mesh is to be used with a smooth transition leading up to the fine mesh zone,
- Finite element size it to be limited to 50 mm x 50 mm.
- The extent of the fine mesh zone is to be not less than 10 elements in all directions from the area under investigation.
- The use of extreme aspect ratio (greater than 3) and distorted elements (corner angles below 60° and greater than 120°) are to be avoided.

- The use of triangular elements is to be avoided.
- All structural parts within an extent of at least 500 mm in all directions leading up to the high stress area are to be modeled explicitly with shell elements
- Stiffeners within the zone are to be modeled using shell elements
- Stiffeners outside the zone may be modeled using beam elements
- The transition between shell elements and beam elements is to be modeled so that the overall stiffener deflection is retained.
- Openings - the first two layers of elements around the opening are to be modeled with mesh size not greater than 50 mm x 50 mm
- Face plates - of openings, primary supporting members and associated brackets are to be modeled with at least two elements across their width on either side

References

1. Nersesian, R.; Mahmood, S. International association of classification societies **2009**. pp. 765–774. doi:10.1163/ej.9789004163300.i-1081.675.
2. 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.
3. 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.
4. 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, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620240603]. doi:https://doi.org/10.1002/nme.1620240603.
5. 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.
6. 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.
7. 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, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4458]. doi:https://doi.org/10.1002/nme.4458.
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. Owen, S. A Survey of Unstructured Mesh Generation Technology. *7th International Meshing Roundtable* **2000**, 3.
10. Hughes, O.; Paik, J.K. *Ship Structural Analysis and Design*; The Society of Naval Architects and Marine Engineers: New Jersey, 2010.
11. Lévy, B.; Liu, Y. L^p Centroidal Voronoi Tessellation and Its Applications.ACM Trans. Graph.2010,29.
12. DNVGL-CG-0127 Finite element analysis," 2015.