OPENMeshS: an online, open, unstructured mesh generator for OPENCoastS

Ricardo MARTINS[1], André B. FORTUNATO[1], Anabela OLIVEIRA[1], Gonçalo JESUS[1], Soroosh MANI[2], Edward MYERS[2], Saeed MOGHIMI[2]

[1] Laboratório Nacional de Engenharia Civil, Av. Brasil, 101, 1700-066 Lisbon, Portugal
rjmartins@lnec.pt, afortunato@lnec.pt, aoliveira@lnec.pt, gjesus@lnec.pt

[2] Coast Survey Development Laboratory, National Ocean Service, NOAA, Silver Spring, 9 MD 20910, USA
soroosh.mani@noaa.gov, edward.myers@noaa.gov, saeed.moghimi@noaa.gov

ABSTRACT

The flexibility provided by unstructured grids is a major advantage for coastal ocean modeling. The drawback is the difficulty to generate good quality unstructured grids, and the steep learning curve required to use grid generators. In addition, the grid generation procedure requires the availability of bathymetry and coastline data for the system of interest. Finding and processing these data can be a major burden on modelers. This paper describes and illustrates an open, online grid generator web assistant that overcomes these problems, greatly simplifying the process of mesh generation for beginner and intermediate users. The grid generator builds both horizontal and vertical grids, and is integrated in the OPENCoastS service.

1. Introduction

Over the past three decades, hydrodynamic modeling in coastal areas has progressively shifted from traditional structured grids to more versatile unstructured grids. The inherent complexity of coastal systems, with irregular shorelines and bathymetry, and sharp gradients in their dynamic features, calls for grids that can seamlessly adapt to these intricacies. This shift thus reflects a recognition of the advantages that unstructured grids offer in capturing the nuances of coastal dynamics, namely a higher flexibility and adaptability compared to their structured counterparts. The ability of unstructured grids to selectively refine specific regions enhances the accuracy of simulations and reduces grid size, enabling a more precise and efficient representation of coastal phenomena such as tidal flows, storm surges, and sediment transport.

Regardless of the selected numerical model, a fundamental aspect influencing the success of hydrodynamic simulations lies in the quality of the grid. This quality requires a balance between a good representation of the geometry and the flow features, a smooth transition between element sizes, and the minimization of the number of nodes. A good-quality grid is necessary for accuracy, stability, and computational efficiency. Firstly, the accuracy of hydrodynamic simulations depends directly on the ability of the grid to capture the intricate details of the coastal landscape, like headlands, narrow channels, and coastal structures. Unstructured grids, with their ability to refine mesh elements in specific regions, excel in providing accurate representations of complex coastal features. However, poorly constructed element stencils can increase truncation errors and deteriorate accuracy (Hagen et al., 2001). Secondly, overly skewed elements can foster the generation of spurious oscillations, causing stability problems or requiring numerical dissipation to avoid instabilities. Finally, computational cost is a key consideration in large-scale or long-term hydrodynamic simulations. Hence, the acceptable number of nodes in a grid often limits the desired resolution and smooth transition between element sizes.

Striking a balance between high resolution, grid smoothness and computational costs requires grid generation tools. Over the past three decades, increasingly sophisticated codes have been developed to help modelers generate grids for coastal areas (Turner and Baptista, 1993; Geuzaine and Remacle, 2009; Conroy et al., 2012; Roberts et al., 2019; Ye et al., 2023). As the demand for larger and better grids grew, so did the grid generator tools increase in complexity. These tools are an absolute requirement for any coastal modeling application based on unstructured grids, considering the size of grids used presently.

Many available grid generation tools come with steep learning curves, making them inaccessible to researchers and practitioners without extensive computational modeling expertise. This barrier hampers the broader application of advanced hydrodynamic modeling techniques, limiting the community's ability to address coastal challenges effectively. Moreover, existing tools often place a burden on users to provide accurate and up-to-date bathymetric and coastline data. This requirement adds an additional layer of complexity and effort, especially in scenarios where such data are not readily available or where frequent updates are needed.

Over the past few years, significant efforts have been undertaken to develop tools to generate on-demand coastal forecast systems (Oliveira et al., 2019, 2021; Trotta et al., 2021). These tools greatly simplify the setup and the launching of numerical simulations, drastically reducing the skills and the effort required from the modelers. In this context, grid generation appears increasingly as the major bottleneck in the modeling procedure. Hence, these developments increase the need for straightforward grid generation tools.

This paper introduces OPENMeshS, an online, open grid generation assistant integrated in the OPENCoastS service (Oliveira et al., 2019, 2021). OPENMeshS seeks to overcome the drawbacks associated with existing grid generation tools by providing a user-friendly platform that integrates seamlessly with the OPENCoastS modeling framework. OPENMeshS takes advantage of several pieces of the OPENCoastS frontend and relies heavily on other pre-existing codes, namely OCSMesh (Mani et al., 2021), JIGSAW (Engwirda, 2014), NICEGRID (Fortunato et al., 2011). While designed for the SCHISM model (Zhang et al., 2016), horizontal grids generated by OPENMeshS can be used in other popular coastal models based on unstructured grids. In the following sections, we delve into the design, features, and demonstration of OPENMeshS, illustrating how it addresses the challenges posed by learning curves and data provision while harnessing the benefits of unstructured grids for coastal hydrodynamic modeling.

2. The mesh generator

OPENMeshS is an assistant integrated in OPENCoastS that helps users to build and generate a mesh in a workflow, which is further detailed in section 3. This assistant can be accessed remotely through any web browser from the OPENCoastS site and does not require installing any tools on the user side. Also, OPENMeshS is fully developed with open source components. Like the OPENCoastS configuration assistant, the grid generation procedure follows several steps where the user selects or provides the information required for the grid generation.

OPENMeshS backend uses Django (web-framework) and Celery (for asynchronous task execution using the RabbitMQ broker). For the mesh generation, OPENMeshS uses OCSMesh (developed by NOAA by processing topo-bathymetric data into georeferenced unstructured meshes, and generate simulation domain and element-size function, using the JIGSAW meshing engine library), NICEGRID and other scripts in Fortran and Python (for mesh analysis and improvement). In the frontend, the user interaction is done by HTML, Javascript, jQuery and Leaflet (map to show the mesh).

Some procedures of the workflow described in section 3 can be very computing intensive depending on the domain size. For instance, the generation of the mesh resolution function, and the node placement and triangularization can take several minutes, locking up the assistant. The assistant relies on Celery to send requests to be processed asynchronously, to avoid the lockup and allowing multiple requests for mesh generation to be submitted and run at same time. Figure 1 describes the handling of these requests. As an example, when the user clicks on the button to generate the mesh in the frontend, a request is sent to Django, then an asynchronous task is sent to Celery to run the OCSMesh mesh generator. Meanwhile, the frontend notifies the user to wait (with the ability to cancel at any time) and checks the task state every 10 seconds. When the task finishes processing, Celery returns the task output (geojson of the generated mesh) to Django, which is then loaded to the frontend map. Computational resources for these tasks are available from the INCD infrastructure.

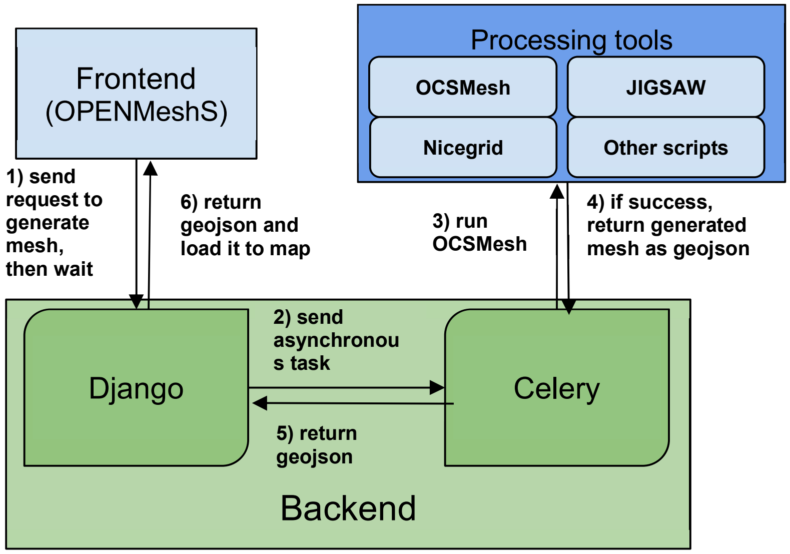

Fig. 1. Diagram how the requests are handled in OPENMeshS

3. Mesh generation procedure

The grid generation procedure follows eight consecutive steps described below (Fig. 2). In Step 1, the user defines a region of interest on the map and provides the bathymetric data or selects from the available bathymetry sources (e.g., GEBCO, EMODNET, SRTM 1-arcsec). Several bathymetric sources can be selected, and the grid bathymetry will be interpolated later from the sources provided here following the chosen order. The grid's land limits can be defined using existing coastlines provided by the grid generator (EMODNET or OpenStreetMaps). As an alternative, the user can provide a maximum elevation, and the grid generator extracts the corresponding isobath from the Digital Elevation Model. The open boundaries are defined within a WebGIS, by defining circular or straight lines on a map (Fig. 3a). These lines can then be adjusted manually, if required. The domain can then be saved as a JSON file, to be reused later.

Fig. 2. Mesh generation workflow

In Step 2, the user defines the spatially-varying mesh resolution by providing the grid's global minimum and maximum resolutions and indicating other optional criteria. At each location, OPENMeshS will use the most stringent criterion to define the grid resolution, within the bounds defined by the user. Several criteria can be applied, such as defining constant resolutions between two isobaths or within a user-defined polygon, or imposing linear increases of resolution with the distance to an isobath. This linear increase is defined through a growth rate, which varies between 0.0001 and 0.8. The result of these criteria can be previewed on a map as isolines of grid resolution (Fig. 3b).

Using the domain and grid resolution defined in the previous steps, a horizontal grid is generated in Step 3 using JIGSAW. JIGSAW generates and triangulates nodes with the resolution defined in Step2. The grid generated can be visualized on the WebGIS (Fig. 3c). An additional script (NICEGRID, Fortunato et al., 2011) can be used to improve the grid. NICEGRID transforms an existing grid such that its internal nodes are connected to about 6 other nodes. Wherever nodes are connected to more or less than 6 other nodes, attempts are made to bring that number closer to 6. This is achieved by adding and deleting nodes, and by switching the connections between them. Nodes are also iteratively moved to the center of the nodes to which they are connected. Nodes located along the boundaries are allowed to slip along the boundary. Because NICEGRID smooths the element size transition, its usefulness depends on the smoothness of the grid initially generated by JIGSAW. Hence, NICEGRID provides a summary of the grid changes, and a recommendation on whether or not the adjusted grid should be adopted.

Once the horizontal grid is defined, the bathymetry is interpolated in Step 4 (Fig. 3d), using the bathymetric sources in the order defined in Step 1.

Boundaries are defined in Step 5. The user can provide a threshold depth, above which the node will be considered part of a closed boundary. As alternative, open boundaries can be defined through consecutive pairs of nodes selected counterclockwise on a map (Fig. 3e). Land boundaries are determined automatically. In this step, users can also impose a minimum depth at elements containing an open boundary node, to prevent drying of those elements during the simulations. At this stage, the final horizontal grid is ready, and can be downloaded in any EPSG coordinate system in Step 6.

Step 7 allows users to generate the vertical grid for SCHISM's 3D simulations. These grids can combine Z-levels in the deeper portion of the domain, with bottom-following S layers at the surface (Fig. 3f). Finally, users can submit the horizontal grid to be stored in OPENCoastS (Step 8).

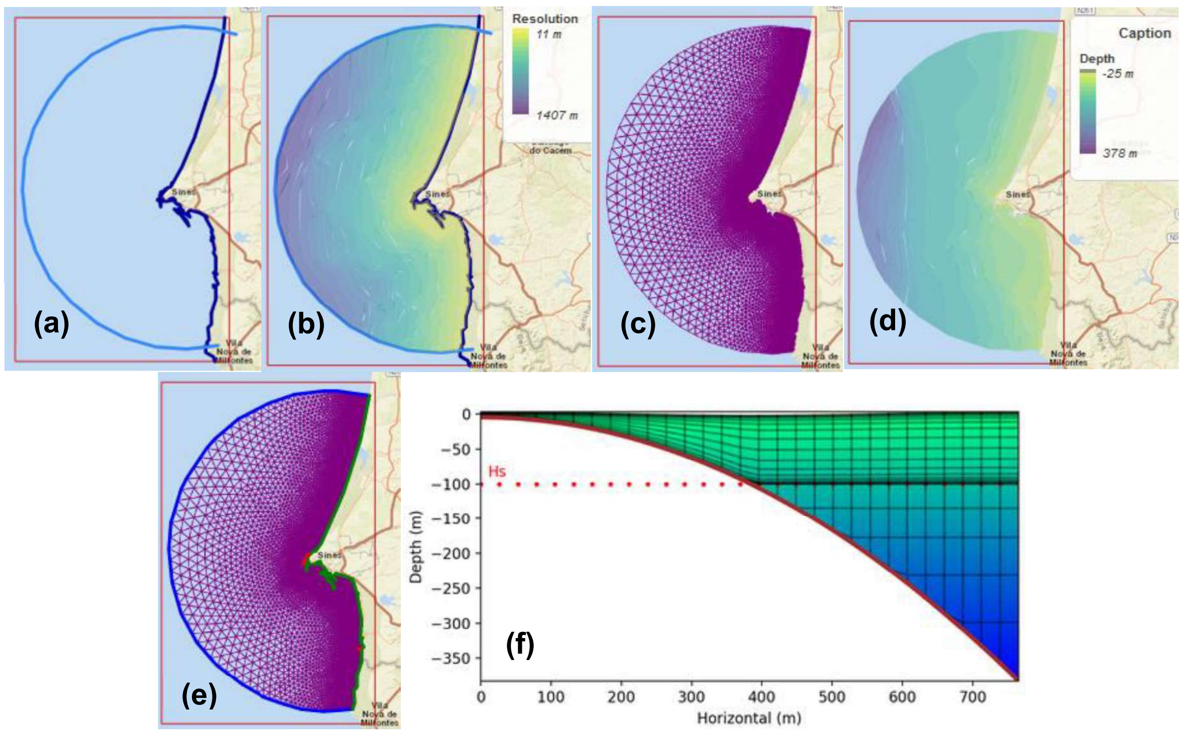

Fig. 3. Results from step 1 (a), step 2 (b), step 3 (c), step 4 (d), step 5 (e) and step 7 (f) of a mesh generation for the Sines Harbor (Portugal)

4. Application examples

4.1. Application to a coastal area (Sines Harbor, Portugal)

The generation of a grid for the Sines Harbor illustrates the use of OPENMeshS in a very simple case (Figure 3). The domain has a semi-circular shape on its ocean limit, with a radius of 25 km, and the land boundary is defined with Open Street Maps (Figure 3a). The topography and the bathymetry are taken from SRTM 1-arcsec and GEBCO, respectively.

Several criteria were used to define the grid resolution, limited to 20-2000 m. First, a resolution of 80 m was imposed above the 10 m isobath, with a growth rate of 0.001. Secondly, a minimum resolution of 50 m was specified in all the harbor areas, with a growth rate of 0.005. Finally, a resolution of 20 m was also imposed along the major harbor structures to allow their representation by the grid (Fig. 3b). The grid generated by OCSMesh was only marginally changed by NICEGRID, but still the maximum number of node connections decreased from 8 to 7 (Figure 3c).

The vertical grid was generated using the default parameters, except the number of Z layers. Because the maximum depth barely exceeds 350 m, only 6 Z layers were used (Fig. 3f).

4.2. Application to an enclosed sea (Sea of Azov, Ukraine and Russia)

The second application illustrates the use of OPENMeshS for an enclosed sea. The Sea of Azov has a surface area of 37,555 km^2. It is bordered by Ukraine and Russia, and connected to the Black Sea by the Kerch Strait.

In Step 1, the Region of Interest (red rectangle in Fig. 4a) is limited by 44.84 and 47.32 latitude North, and 34.8 and 39.34 longitude East. Land borders were defined using the Open Street Maps, and the open boundary in the Black Sea was defined as an arc-circle with a 25 km radius. The topography and the bathymetry are taken from SRTM 1-arcsec and GEBCO, respectively. A few bays with very narrow entrances were omitted to limit the number of nodes.

Given the relatively large size of the domain, the grid resolution (Fig. 4b) varies between 300 m, for depths below 5 m, and 5000 m. In addition, a minimum grid resolution of 1000 m is imposed in the Kerch Strait, where large velocities are expected. The resolution growth rate (0.005) from the constant resolution areas reduces the number of nodes, but can lead to sharp variations in element sizes. NICEGRID is thus used to smooth the element size transitions, increasing the minimum number of node connections from 4 to 5. The final mesh has 73,730 nodes (Fig. 4c).

The domain is very shallow and does not have any sharp bathymetry gradients. Hence the vertical grid only uses S coordinates. The default parameters (θ_B=1, θ_F=8) provide finer resolutions near the bottom and surface (Fig. 4d).

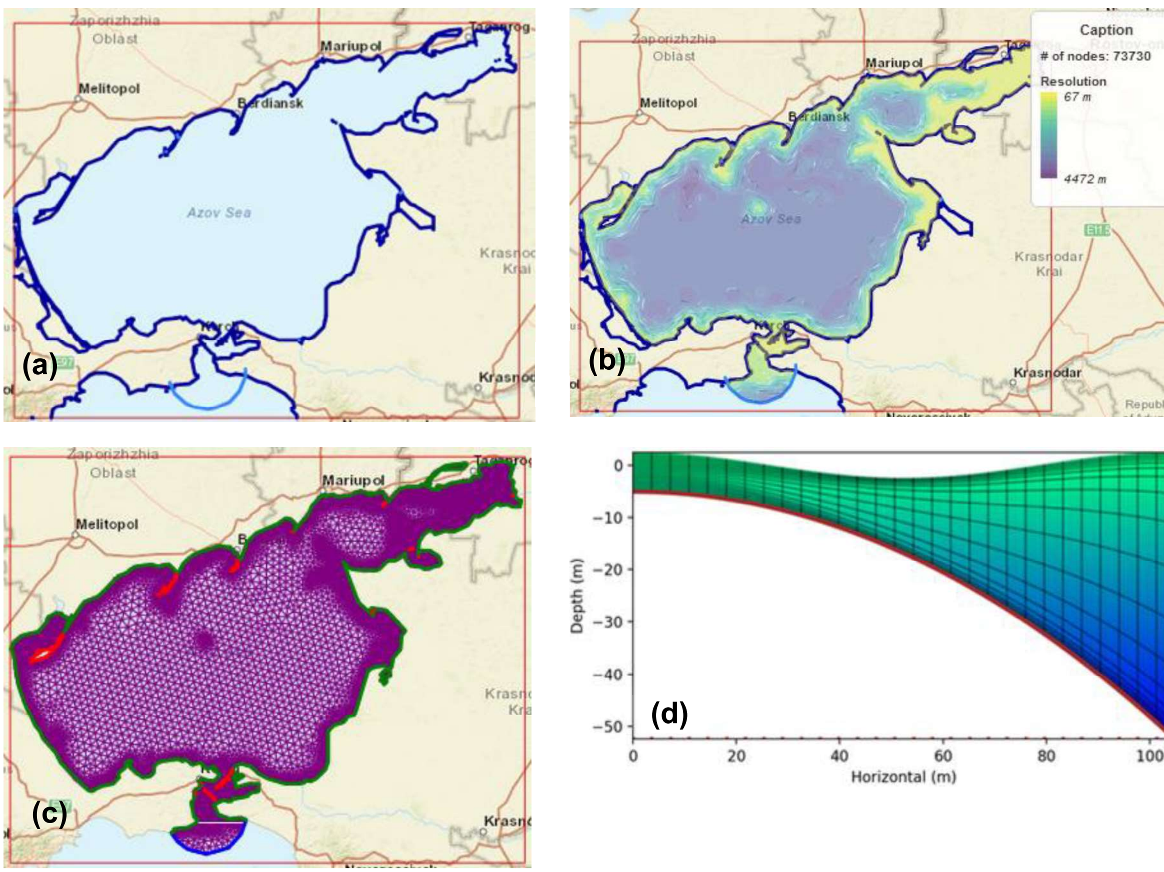

Fig. 4. Results from step 1 (a), step 2 (b), step 5 (c), and step 7 (d) of a mesh generation for the Azov Sea (Ukraine, Russia)

5. Conclusions

In summary, the adoption of unstructured grids in coastal hydrodynamic modeling signifies a paradigm shift towards increased accuracy, stability, and efficiency. As we explore the landscape of existing grid generation tools, their limitations become apparent, emphasizing the need for a user-friendly and accessible solution. OPENMeshS stands poised to bridge this gap, offering a platform that empowers researchers and practitioners to unlock the full potential of unstructured grids in coastal modeling, ultimately contributing to a better understanding of coastal dynamics and improved decision-making processes.

From the feedback from beta testers, this tool has proven to be very powerful, helping users to easily build meshes from their domains and speeding up their forecast deployments. Because the mesh assistant can generate very large meshes, the number of grid nodes in OPENCoastS was arbitrarily limited to 175000 to avoid overloading the available cloud resources.

Acknowledgements

We acknowledge Darren Engwirda for the JIGSAW code, and all the team from LNEC for the development of OPENCoastS and the scripts required for this assistant. This work was partially funded by the European Union's Horizon 2020 research and innovation programme under project EGI-ACE, grant agreement No. 101017567 and project ATTRACT-DIH, grant agreement No. 101083770

References

Conroy CJ, Kubatko EJ, West DW (2012) ADMESH: An advanced, automatic unstructured mesh generator for shallow water models, *Ocean Dynamics*, 62/10-12: 1503-1517

Engwirda D (2014) Locally-optimal Delaunay-refinement and optimisation-based mesh generation, Ph.D. Thesis, School of Mathematics and Statistics, The University of Sydney, http://hdl.handle.net/2123/13148, 2014.

Fortunato AB, Bruneau N, Azevedo A, Araújo MAVC, Oliveira A (2011) Automatic improvement of unstructured grids for coastal simulations, *J. Coast. Res. Special Issue*, 64: 1028-1032

Geuzaine C, Remacle JF (2009) Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering, 79/11: 1309-1331

Hagen SC, Westerink JJ, Kolar RL, Horstmann O (2001) Two-dimensional, unstructured mesh generation for tidal models, International Journal for Numerical Methods in Fluids, 35(6): 669-686

Mani S, Calzada JR, Moghimi S, Zhang YJ, Myers E, Pe'eri S (2021) OCSMesh: a data-driven automated unstructured mesh generation software for coastal ocean modeling, NOAA Technical Memorandum NOS CS; 47

Oliveira A, Fortunato AB, Rogeiro J, Teixeira J, Azevedo A, Lavaud L, Bertin X, Gomes J, David M, Pina J, Rodrigues M, Lopes P (2019) OPENCoastS: An open-access service for the automatic generation of coastal forecast systems, Environmental Modelling & Software, 124: 104585

Oliveira A, Fortunato AB, Rodrigues M, Azevedo A, Rogeiro J, Bernardes S, Lavaud L, Bertin X, Nahon A, Jesus G, Rocha M, Lopes P (2021) Forecasting contrasting coastal and estuarine hydrodynamics with OPENCoastS, Environmental Modelling & Software, 143: 105132

Roberts KJ, Pringle WJ, Westerink JJ (2019). OceanMesh2D 1.0: MATLAB-based software for two-dimensional unstructured mesh generation in coastal ocean modeling, Geoscientific Model Development, 12/5: 1847-1868

Trotta F, Federico I, Pinardi N, Coppini G, Causio S, Jansen E, Iovino D, Masina S (2021) A Relocatable Ocean Modeling Platform for Downscaling to Shelf-Coastal Areas to Support Disaster Risk Reduction, Frontiers in Marine Science, 8, DOI: 10.3389/fmars.2021.642815

Turner P, Baptista AM (1993). ACE/gredit User's Manual. Software for Semi-automatic Generation of Two-Dimensional Finite Element Grids, Center for Coastal and Land-Margin Research, Oregon Graduate Institute of Science & Technology

Ye F, Cui LL, Zhang YL, Wang ZG, Moghimi S, Myers E, Seroka G, Zundel A, Mani S, Kelley JGW (2023) A parallel Python-based tool for meshing watershed rivers at continental scale. Environmental Modelling and Software, 166: 105731