

Multiscale mesh generation on the sphere

J. Lambrechts¹, R. Comblen¹, V. Legat¹, C. Geuzaine², J.-F. Remacle¹

¹ Université Catholique de Louvain, Institute for Mechanical, Material and Civil Engineering, Louvain-la-Neuve, Belgium

² Electrical Engineering And Computer Science, Montefiore Institute, Liège, Belgium.

submitted to Ocean Dynamics, 2008

Key words. Mesh Generation, Sphere, Ocean Modeling

Abstract. A method for generating computational meshes for applications in ocean modeling is presented. The method uses a standard engineering approach for describing the geometry of the domain that requires meshing. The underlying sphere is parametrized using stereographic coordinates. Then, coastlines are described with cubic splines drawn in the stereographic parametric space. The mesh generation algorithm builds the mesh in the parametric plane using available techniques. The method enables to import coastlines from different data sets, and consequently to build meshes of domains with highly variable length scales. The results include meshes together with numerical simulations of various kinds.

1 Introduction

Finite elements have been used in engineering analysis for several decades. Since the nineties, geometric domains that are used in finite element analysis and design are built using Computer Aided Design (CAD) programs. Today's CAD systems are highly reliable: they deal with most of the complex geometric features of industrial parts or assemblies.

Traditional ocean models are based on finite differences schemes on Cartesian grids (Griffies et al., 2000). It is only recently that finite elements and unstructured meshes have been used in ocean modeling (e.g. Piggott et al., 2007; White et al., 2008; Danilov et al., 2005). One of the advantages of unstructured grids is their ability to conform to coastlines.

As unstructured grid ocean models began to appear, mesh generation algorithms were either specifically developed or simply adapted from classical engineering tools. Le Provost et al. (1994) use the mesh generation tools of Henry and Walters (1993) on several subdomains to obtain a mesh of the world ocean, aiming at global scale tidal modeling. Further, Lyard et al. (2006) use a higher resolution version of the same kind of meshes with the state of the art FES2004 tidal model. Hagen et al. (2001) give two algorithms to generate meshes

of coastal domains, and use them to model tides in the Gulf of Mexico. Legrand et al. (2006) show high-resolution meshes of the Great Barrier Reef (Australia). At the global scale, Legrand et al. (2000) and Gorman et al. (2007) developed specific algorithms to obtain meshes of the world ocean.

Our domain of interest is the Earth's surface, i.e., within a sufficiently good approximation, a sphere \mathcal{S} centered at the origin and of radius R of about 6370 km. The World Ocean is bounded by continents and islands coastlines. The first aim of the paper is to describe an automatic procedure that enables to build a boundary representation (BRep) of the geometry of the World Ocean within a prescribed accuracy. This procedure takes advantage of various sets of data: high resolution shoreline databases (Wessel and Smith, 1996), global relief data (National Geographic Data Center, 2006), local cartographic data, etc.

Even if accurate data is available, it cannot be envisaged to build a BRep with the maximal available resolution everywhere. For example, the *up to date in 2008* global shoreline database has a resolution of about 50m, which would lead to a huge number of control points (9,451,331) ! Our procedure enables to construct a model with an adaptive geometrical accuracy. Some regions of interest of the globe are discretized with the maximal available geometrical accuracy while other regions are approximated in a coarser way. Our technique also allows to mix various data sets as input.

Numerical analysis procedures utilize meshes, i.e., discretized versions of the domains described by CAD models. In this paper, we have decided *not* to develop a new mesh generation algorithm specifically designed for doing meshes that can be used in finite element marine modeling. Here, we have rather decided to build a CAD model that can serve as input for *any* surface mesher. In the last decade, mesh generation procedures have evolved with the objective of being able to interact directly with CAD models (e.g. Geuzaine and Remacle, 2008; Beall and Shephard, 1997; Haimes, 2000). More specifically, some of the authors of this paper have developed Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing

facilities (Geuzaine and Remacle, 2008). The specific nature of the model—the Earth surface with several thousands of islands, including hundreds of thousands of control points—have led us to greatly improve the meshing procedures implemented in Gmsh. Those specific features are also explained in the paper.

The paper is divided in three sections. The first section deals with the procedure for building CAD models of ocean geometries. The second section describes mesh generation procedures. In the last section, we provide illustrative examples with diverse simulation results.

2 A geometric model for the World Ocean

Any 3-D model can be defined using its Boundary Representation (BRep): a volume (called *region*) is bounded by a set of surfaces, and a surface is bounded by a series of curves; a curve is bounded by two end points. Therefore, three kinds of *model entities* are used: *model vertices* G_i^0 (dimension 0), *model edges* G_i^1 (dimension 1), and *model faces* G_i^2 (dimension 2).

Model entities are topological entities, i.e., they only deal with adjacencies in the model. A geometry has to be associated to each model entity. The geometry of curves and surfaces are their shapes. A parametrization of the shapes, typically a mapping, is usually available.

The geometry of a model edge is its underlying curve defined by the parametrization:

$$t \in \mathcal{R} \mapsto \mathbf{p}(t) \in \mathcal{R}^3.$$

Similarly, the geometry of a model face is its underlying surface defined by the parametrization:

$$(u, v) \in \mathcal{R}^2 \mapsto \mathbf{p}(u, v) \in \mathcal{R}^3.$$

If a curve is included within a surface, it is usually drawn on the parameter plane (u, v) of the surface:

$$t \in \mathcal{R} \mapsto (u(t), v(t)) \in \mathcal{R}^2 \mapsto \mathbf{p}(u(t), v(t)) \in \mathcal{R}^3.$$

As an illustration, let us consider the surface represented in Figure 1. Most important features of model entities are highlighted in this example:

- The surface is periodic. A seam curve has been introduced in the list of boundary edges of the surface to define its closure properly.
- The surface is trimmed: it contains four holes and one of those holes is crossed by the seam.
- One of the model edges on the closure of the model face is degenerated. Degenerated edges are used to take into account singularities of the mapping. Such degeneracy is present in many surface geometries: spheres, cones and other surfaces of revolution.

From an engineering point of view, dealing with the geometry of the ocean is dealing with a trimmed sphere—i.e., a surface that is periodic, that is bounded by continents and islands and that has degeneracies at both poles.

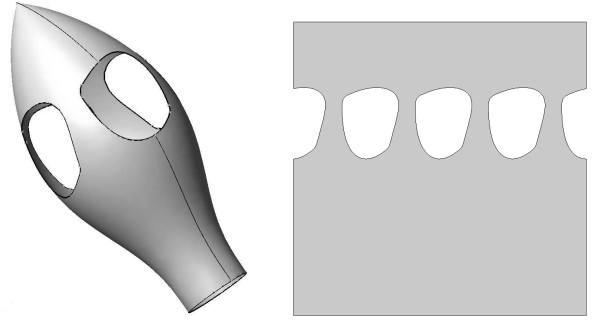


Fig. 1. Figures show a model face in both real and parametric coordinates. The seam of the surface is highlighted in the right Figure.

2.1 Parametrization of the sphere

Several parametrizations exist for the sphere. CAD systems use spherical coordinates. In geosciences, most of the available data are expressed in the geographic coordinate system, which has the same properties as the spherical coordinate system. Spherical coordinates suffers from all the problems that we have just mentioned before: there exist two singular points in the mapping, leading to the definition of two degenerated edges in the model; one of the coordinate directions is periodic, leading to the introduction of one seam edge; shorelines may cross the seam edge, leading to complexity in the definition of the geometry. It is indeed impossible to choose the seam edge so that it does not cross any shoreline. In Figure 2, a mesh of the World Ocean built using the spherical coordinate system is shown. The seam passes through the Bering strait and crosses the Pacific ocean, ending somewhere in the coastline of Antarctica.

Moreover, the spherical coordinates, as most of the parametrizations, are not conformal. A conformal mapping will conserve angle at which curves cross each other. Consequently, in order to obtain an isotropic mesh in the real space, one has to build an anisotropic mesh in the parametric plane. In the case of spherical coordinates, the mapping is highly distorted near the singularities, i.e., near the poles. Robust surface meshers might be able to deal with that issue, as illustrated in Figure 2. Anyway, it is always better to use a conformal mapping, such as the stereographic projection of Figure 3.

Let us consider a sphere \mathcal{S} centered at the origin and of radius R , and one point s . This point lies on the surface of the sphere, does not belong to the oceans, and will be the only singular point of the mapping. A suitable choice for s could be a location in the middle of Kazakhstan but here, we choose $s = \{0, 0, -R\}$. It corresponds to the South Pole. Antarctica being a continent, this choice makes sense for ocean modeling applications. The stereographic projection consists in projecting points \mathbf{p} of the sphere on the plane $z = R$.

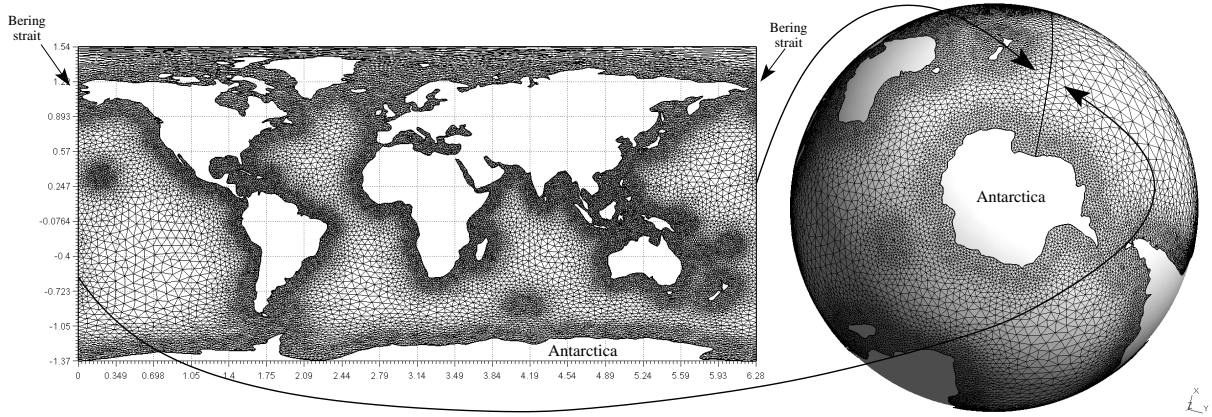


Fig. 2. Mesh of the World Ocean using the spherical coordinate system. The seam edge is visible on the right plot.

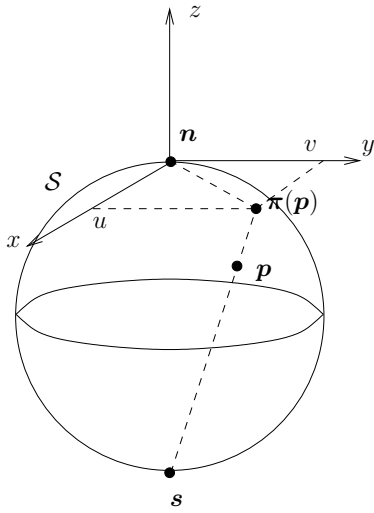


Fig. 3. Stereographic projection.

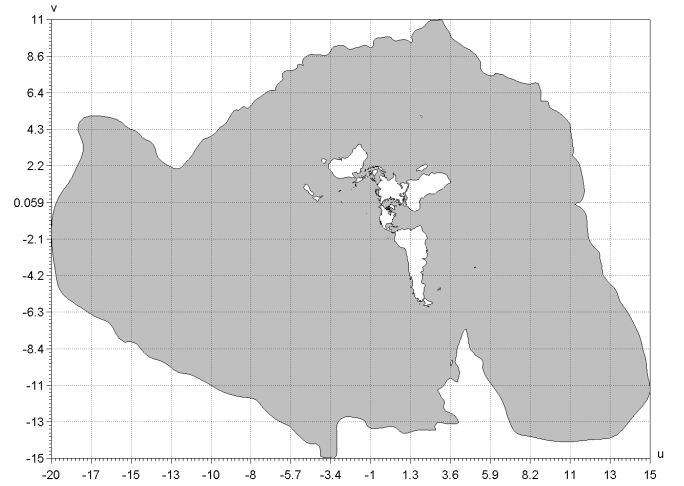


Fig. 4. The World Ocean in stereographic coordinates.

The stereographic projection $\mathbf{u}(\mathbf{x}) = \{u, v\}$ of a point $\mathbf{x} = \{x, y, z\}$ is the intersection of vector $\mathbf{q} - \mathbf{p}$ with $z = R$:

$$\mathbf{u} = \{u, v\} = \left\{ \frac{2R}{R+z}x, \frac{2R}{R+z}y \right\},$$

$$\mathbf{x} = \{x, y, z\} = \frac{4R^2}{u^2 + v^2 + 4R^2} \{u, v, R(4R^2 - u^2 - v^2)\}.$$

Figure 4 shows the World Ocean in stereographic coordinates $\{u, v\}$. The outside loop surrounding the domain is the stereographic projection of the Antarctica. The radius of the Earth is chosen arbitrarily to $R = 1$. No seam is required to define the overall domain and no singular point exists in the domain of interest.

2.2 Coastlines definition

Today, the most accurate shoreline database is, to date, the Global Self-consistent Hierarchical High-resolution Shorelines (GSHHS, (Wessel and Smith, 1996)). This data set de-

scribes the Earth's coastlines with a global resolution of about 50 m. GSHHS data are guaranteed to be self-consistent, i.e., coastlines in the database do not intersect themselves. In our approach, we define a size field $\gamma(\mathbf{x})$ that expresses the geometrical requested accuracy of the model at any point \mathbf{x} . The GSHHS data set is coarsened with respect to this size field and every successive points at \mathbf{x} that are closer than $\gamma(\mathbf{x})$ are collapsed.

A coastline is defined in practice as a periodic curve. A first approach could be to use a piecewise linear definition for defining such a curve. A first naive approach would consist in defining such a curve by a piecewise linear interpolation. However, in order to enjoy more flexibility, we use here cubic B-splines with control points taken in the GSHHS database. As B-splines remain inside the convex hull defined by the control points, it can be derived that if the piecewise linear representation is convex and consistent, then the curvilinear B-splines representation is also consistent.

In Figure 5, we generate the coastlines of Great Britain and Ireland with different resolutions. With a resolution $\gamma =$



Fig. 5. Great Britain and Ireland with a resolution of 100 km (top/left), 15 km (top/right), 1.5 km (bottom/left) and 150 m (bottom/right). Splines control points are depicted on the geometry with the two lowest resolutions.

100 km, we only consider Ireland and Great Britain, and we neglect all smaller islands. With a resolution of $\gamma = 15$ km, 15 contours appear in the domain. Typically, the Isle of Wight is now included. With a resolution of $\gamma = 1.5$ km, the domain contains 152 islands. With a resolution of $\gamma = 150$ m, the domain contains 2176 islands and a total of 83,277 control points.

3 Mesh generation

As an accurate representation of the boundaries is now available, the next task involve the generation of finite element meshes on curved surface. Two major approaches are available:

- Techniques for which the surface mesh is generated directly in the real 3D space.
- Techniques for which the surface mesh is generated in the parametric plane of the surface.

When a smooth parametrization of the surface exists, building the mesh in the parametric plane appears to be the most robust choice.

3.1 Definition of a local mesh size field

The aim of the mesh generation process is to build elements of controlled shape and size. Mesh generators are usually able to adapt to a so called mesh size field. An isotropic mesh size field is a scalar function $\delta(\mathbf{x})$ that defines the optimal length of an edge at position \mathbf{x} of the real space. In the domain of ocean modeling, there exist some heuristics on the way mesh sizes should be distributed in the World Ocean.

The mesh should take into account the bathymetry. The bathymetry $H(\mathbf{x})$ is taken into account in two ways, leading to two fields f_1 and f_2 . Gravity waves move at speed \sqrt{gH} with g being the acceleration of gravity. The lengthscale λ of a gravity wave is therefore proportional to $\lambda = \mathcal{O}(1/\sqrt{H})$. If we consider that N mesh sizes are necessary to capture one wavelength, and if λ_{\min} is the smallest wavelength that has to be captured for a reference bathymetry H_{ref} , we define f_1 as

$$f_1(\mathbf{x}) = \frac{\lambda_{\min}}{N} \sqrt{\frac{H_{ref}}{H(\mathbf{x})}}.$$

Another way of taking into account the bathymetry is to force the mesh to capture its variations with a given accuracy (Gorman et al., 2006). Bathymetry can be seen as a scalar field defined at mesh vertices and interpolated piecewise linearly. As the first term of error in its interpolation is supposed to depend on λ_{\max} the greatest (in absolute value) eigenvalue of the Hessian $\mathcal{H}(x)$,

$$\mathcal{H}(x) = \nabla \nabla \left(\frac{H(x)}{H_{ref}} \right),$$

we define the second field :

$$f_2(x) = \frac{1}{\sqrt{\lambda_{\max}}}.$$

In order to well represent to coastlines and to capture the small scales phenomenons generated by the friction on the coasts, mesh size should be even smaller near coastlines. This criterion has already been used in the literature (e.g. Legrand et al., 2006). We define a first field $f_3(x)$ as the distance to the closest shoreline:

$$f_3(x) = d(x).$$

This field f_3 is also called a *shore proximity function*.

This distance can be computed *in place* using the Approximated Nearest Neighbor Algorithm (ANN, (Arya et al., 1998)).

For each criterion field f_i , a mesh size field δ_i is computed as follow:

$$\delta_i(x) = \delta_i^{\text{small}} + \alpha_i(x)(\delta_i^{\text{large}} - \delta_i^{\text{small}}),$$

where

$$\alpha_i(x) = \begin{cases} 0 & \text{if } f_i(x) \leq f_i^{\min} \\ \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} & \text{if } f_i^{\min} < f_i(x) < f_i^{\max} \\ 1 & \text{if } f_i(x) \geq f_i^{\max} \end{cases}$$

with δ_i^{large} and δ_i^{small} a large and a small desired mesh size, f_i^{\max} and f_i^{\min} two field values that define the zone of refinement. The final size field is simply computed as the minimum of all size fields:

$$\delta(x) = \min(\delta_1(x), \delta_2(x), \dots).$$

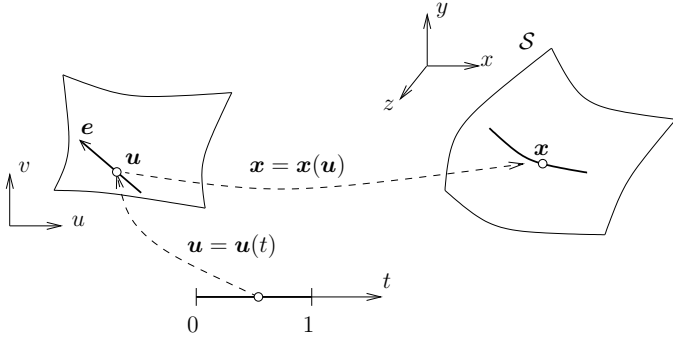


Fig. 6. Three parametrizations of a straight edge.

Finally, it is always possible to add other size fields, as error estimators that may depend on the finite element solution.

In Figure 6, we consider a straight edge e described by its vector e in the parametric plane, where the mesh generation process is performed. Its length L is computed as follows:

$$L = \int_e \sqrt{\|d\mathbf{x}\|^2} = \int_e \sqrt{d\mathbf{u}^T \mathbf{J}^T \mathbf{J} d\mathbf{u}} = \int_0^1 \sqrt{\mathbf{e}^T \mathbf{M} \mathbf{e}} dt$$

where $\mathbf{J} = \partial\mathbf{x}/\partial\mathbf{u}$ is the Jacobian of the mapping and $\mathbf{M} = \mathbf{J}^T \mathbf{J}$ is the metric tensor. In the case of a stereographic projection, both eigenvalues of \mathbf{M} are positive, and equal:

$$\lambda(\mathbf{u}) = \left(\frac{4R^2}{u^2 + v^2 + 4R^2} \right).$$

To obtain the mesh in the parametric space but with the right sizing in the real space, a suitable mesh size field $\delta_u(\mathbf{u})$ has to be defined in this parametric plane. As the stereographic projection is a conforming mapping, it can be defined with a simple scaling:

$$\delta_u(\mathbf{u}) = \delta(\mathbf{x}(\mathbf{u})) \frac{1}{\lambda(\mathbf{u})}.$$

3.2 Coastlines mesh generation

Let us consider a curve in the parametric plane $\mathbf{u}(t) : [0, 1] \rightarrow \mathcal{R}^2$. It is easy to show that the number of subdivisions N of the curve is the following function of the size field

$$N = \int_0^1 \frac{1}{\delta_u(\mathbf{u}(t))} \|d_t \mathbf{u}\| dt,$$

where $\|d_t \mathbf{u}\| = \sqrt{(\partial_t u)^2 + (\partial_t v)^2}$. The $N + 1$ mesh points on the curve are located at coordinates $\{t_0, \dots, t_N\}$ where t_i is computed using the following rule:

$$i = \int_{t_0}^{t_i} \frac{1}{\delta_u(\mathbf{u}(t))} \|d_t \mathbf{u}\| dt.$$

Integration of those expressions must be performed with and adaptive trapeze rule, as coastlines are discretized with

cubic splines that contain a large number of control points. Typically, Europe and Asia are discretized by only one spline with more than twenty thousand control points, in Figure 4.

However, this algorithm does not guarantee that, even if the model edges G_j^1 that constitute the boundaries of the domain are non intersecting, the corresponding 1D meshes do not self intersect. Figure 7 shows two islands very close to each other. Yet, even the geometry is itself not self-intersecting, the first 1D generated mesh intersects itself. This can be considered as a critical issue: modifying the mesh size field *by hand* locally cannot be considered when several thousand of islands are to be involved. It is therefore mandatory to define a systematic recovery procedure. Such an algorithm illustrated in Figure 7 works as follow:

1. A Delaunay mesh that contains all points of the 1D mesh is initially constructed using a divide and conquer algorithm (Dwyer, 1986).
2. Missing edges are recovered using edge swaps (Weatherill, 1990). If a mesh edge e_i that belongs to the 1D mesh is to be swapped for recovering edge e_j , then the mesh edges e_i and e_j that both belong to the 1D mesh intersect.
3. All intersecting edges e_k are split in 2 segments and the new point is snapped onto the geometry. Then, we go back to the first step until the list of intersecting edges is empty.

If an intersecting edge is smaller than the geometrical tolerance, then an error message is thrown claiming that the geometry is self intersecting. Note that when a unique mesh edge connects two different islands, those islands are numerically merged if a non slip boundary condition is applied along their coastlines.

3.3 Surface mesh generation

To generate a mesh on the sphere, three approaches are available in Gmsh software. All of them start with an initial Delaunay mesh that contains all the mesh vertices of the contours. Then, every mesh edge of the 1D mesh is recovered using edge swaps. Then, internal vertices are iteratively inserted inside the domain. The way points are inserted differ in the three algorithms:

- The *del2d* algorithm is inspired from the work of the GAMMA team at INRIA (George and Frey, 2000). New points are inserted sequentially at the circumcenter of the element that has the largest adimensional circumradius. The mesh is then reconnected using an anisotropic Delaunay criterion.
- In the *frontal* algorithm (Rebay, 1993) new points are inserted optimally on Voronoï edges. The mesh is then reconnected using the same anisotropic Delaunay criterion as the one in the *del2d* algorithm. Note that this algorithm's implementation only differs slightly from the one of algorithm *del2d*.

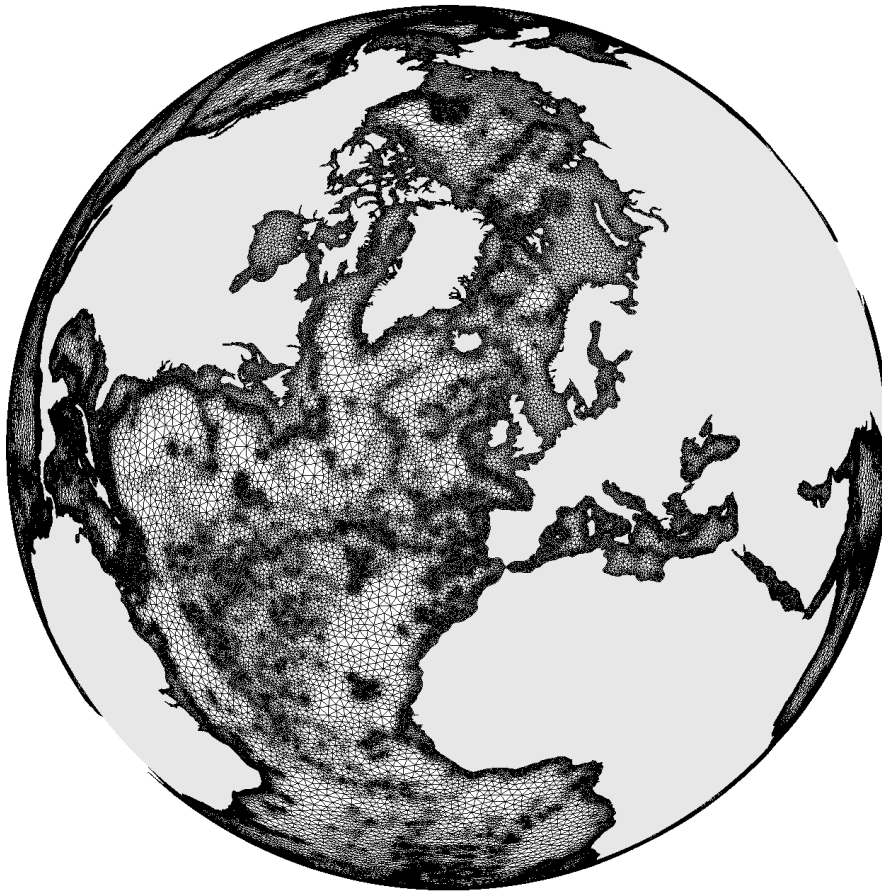


Fig. 9. Mesh of the World Ocean. The mesh size field is defined using a shore proximity function, the bathymetry and its Hessian.

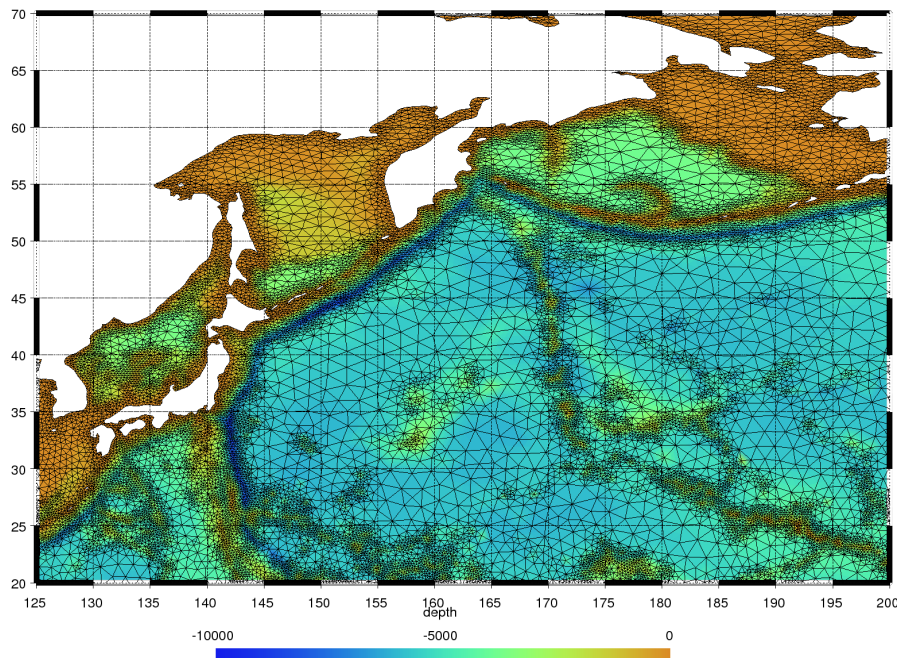


Fig. 10. Close up of the mesh of Figure (9) in the north Pacific ocean, color levels represent the bathymetry (in m.). The effects of the three refinement rules are clearly visible

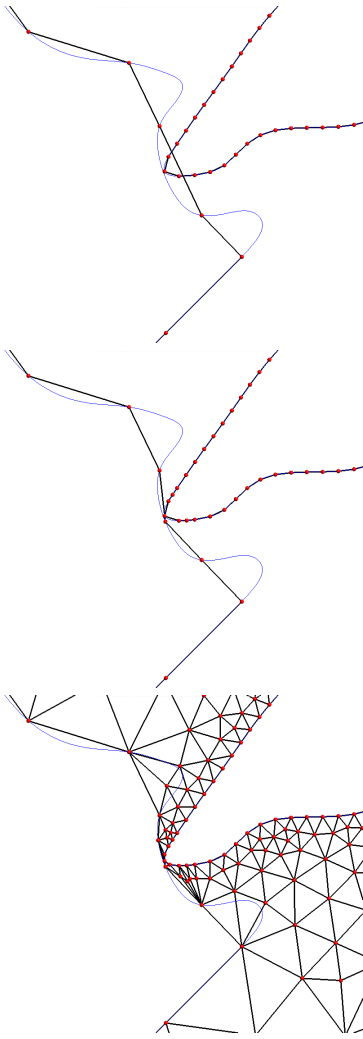
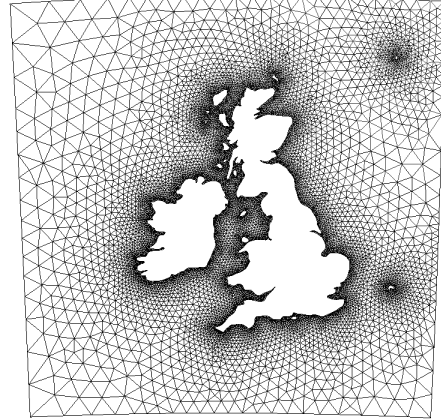


Fig. 7. A geometry with two islands that are very close to each other. The first image shows the initial 1D mesh that respects mesh size field. The second image shows the first iteration of the recovery algorithm. The third image shows the final mesh that was possible to realize after 2 recovery iterations

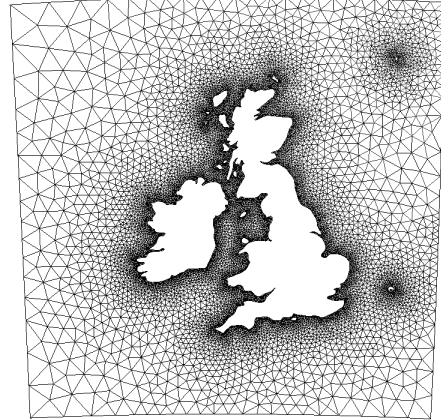
- The *meshadapt* algorithm is very different to the two first ones. It is based on local mesh modification (Geuzaine and Remacle, 2008): this technique makes use of edge swaps, splits and collapses. Long edges are split, short edges are collapsed and edges are swapped if a better geometrical configuration is obtained.

The *frontal* algorithm usually gives the highest quality meshes while the *del2d* algorithm is the fastest: it produces about 5 million triangles a minute, if the size field δ is not too complex to compute. Figure 8 presents three meshes of one of the model of Figure 5 for which we have used a shore proximity function as the only size field. Meshes have respectively 18698, 19514 and 17154 triangles. The percentage of elements that have an aspect ratio $\rho > 0.9$ is respectively 93.2% 88.1% and 84.5%. CPU time for generating meshes was respectively 0.7 seconds, 0.5 seconds and 5.7 seconds.

a) Mesh done using the *frontal* algorithm



b) Mesh done using the *del2d* algorithm



c) Mesh done using the *meshadapt* algorithm

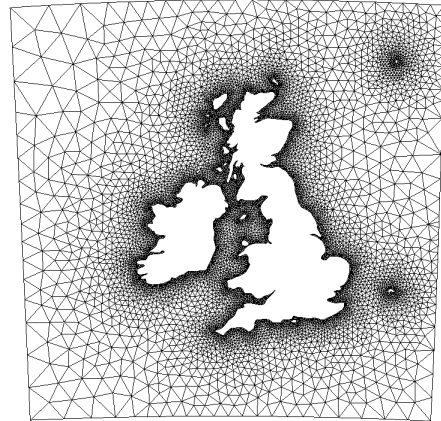


Fig. 8. Meshes of the same domain using three different algorithms.

Figures 9 and 10 present a mesh of the World Ocean that makes use of all size fields defined in Section 3.1:

- A shore proximity function f_1 is used with $\delta_1^{\text{small}} = 30$ km, $\delta_1^{\text{large}} = 200$ km, $f_1^{\text{min}} = 0$ and $f_1^{\text{max}} = 500$ km.
- We use f_2 and refine the mesh proportionally to the square root of the ocean depth. The size field δ_2 ranges from 25 km to 500 km.

- We use f_3 to capture the bathymetry. The size field δ_3 ranges from 25 km to 500 km.

The resulting mesh is generated of 436,409 triangles and the whole mesh generation process (data loading, coastline reduction, 1d mesh generation, 2d mesh generation, output files writing) takes 35 seconds on a recent laptop. Those timings compare advantageously with alternative techniques based on mesh decimation (Gorman et al., 2006). The memory footprint of the meshing algorithms is low: about 12 million triangles (6 million nodes) can be generated per Gigabyte of memory.

4 Examples

In the mesh generation community, it is assumed that a good paper should present nice pictures of meshes. We will not circumvent that prerequisite. Yet, mesh generation is usually considered as a tool, not as an aim. Therefore, the meshes that we present in this section are accompanied with some simulation results.

4.1 Sea ice modeling

The mesh presented on Figure 11 was used to investigate the sensitivity of the Arctic sea ice cover features to the resolution of the narrow straits constituting the Canadian Arctic Archipelago. This mesh is constituted of 17,053 triangles with a resolution of 20 km near the islands in the archipelago and 40 km elsewhere. Far from coasts and islands the resolution decreases up to 300 km. Model results are shown on Figure 12. A complete description of the model and its validation can be found in Lietaer et al. (2008, in press, doi 10.1016/j.ocemod.2008.06.002).

4.2 Multiscale model of the Scheldt river

Within the framework of the multi-disciplinary project TIMOTHY¹, our team is presently involved in the development of a two-dimensional hydrodynamic model of the Scheldt Estuary, in the Netherlands. Our main goal in this project is to take advantage of the finite element method to study very specific ecological problems, such as the dynamics of faecal bacteria or heavy metals. We also intend to study the characteristic time scales determining the physics and the biology in the estuary. In this model, the tide is forced at the shelf break, which is more than 1000 km far from the mouth of the estuary, and the upstream boundary is situated in the areas of Antwerp, where the river width is a few hundreds of meters. The multi-scale character of the problem is then one of its main feature

¹ TIMOTHY, Tracing and Integrated Modeling of Natural and Anthropogenic Effects on Hydrosystems : The Scheldt River basin and adjacent coastal North Sea,
<http://www.climate.be/TIMOTHY>



Fig. 11. Mesh of the Arctic region (north of the parallel 50 degrees North) especially refined along coastlines and in the Canadian Arctic Archipelago.

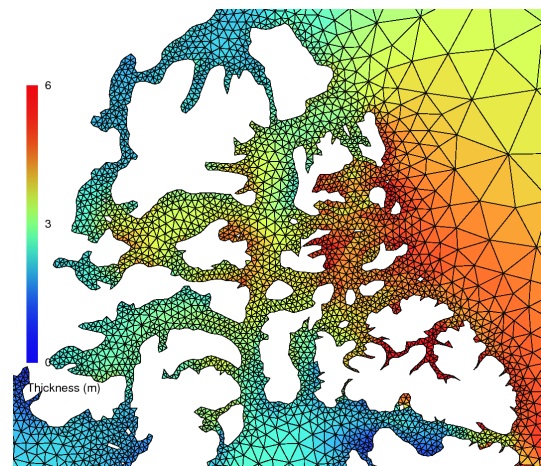


Fig. 12. Detail of the mesh of Figure 11. Mean March sea ice thickness pattern (in m) in the western part of the Canadian Arctic Archipelago as computed by the finite element sea ice model (1979-2005).

and the use of an unstructured grid is thus totally appropriate. Figure 13 presents a mesh 27,472 elements used in our preliminary runs. Various criteria based on the distance from coasts, islands and shelf break are used to define the mesh size fields. The element size range from 150 m in the Scheldt river near Antwerp to about 50 km far from the coastlines.

4.3 The Great Barrier Reef

Our research team has developed the first multiscale hydrodynamic model of the whole Great Barrier Reef. The Great Barrier Reef is on the continental shelf of the Australian north-eastern coastline. There are over 2500 coral reefs in a strip

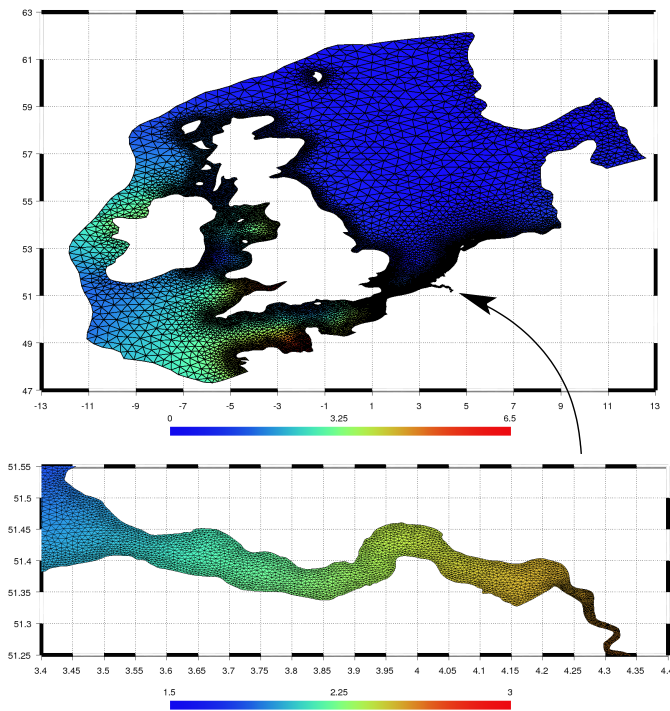


Fig. 13. Multiscale mesh: north sea and Scheldt river estuary. Color levels represent the amplitude of the M2 tidal component (in m).

that is about 2600 km in length and 200 km in width. The simulation that is presented here make use of most the specific mesh generation features that were presented in this paper: a geometric domain with multiple scales, use of a shore proximity function, grid adaptation with respect to the bathymetry, special refinement in the domain of interest. A complete simulation is described in Lambrechts et al. (2008). Figure 15 shows a mesh built to run small simulation on a single CPU to study a specific region while keeping the boundary conditions of the complete simulation. Around a specific island (Lizard Island), the resolution is sufficient to capture some small scale hydrodynamic features like tidal jets in small interreefs passages and recirculations around islands. Elsewhere on the shelf, the resolution is very coarse, those regions are only used as boundary conditions. 70% of the 20,384 elements are located in the refined region. A plot of velocity vectors is also presented. Tidal jets and eddies due to the interaction of the flow with the topography near the open-sea boundary are clearly visible. Those small scale features were captured thanks to the accurate description of the bottom topography.

5 Conclusion

A CAD based mesh generation procedure for ocean modeling has been developed. The new approach has the advantage to rely on existing well known engineering mesh generation procedures. The CAD model, based on a smooth boundary representation of the domain, allows to build a compact,

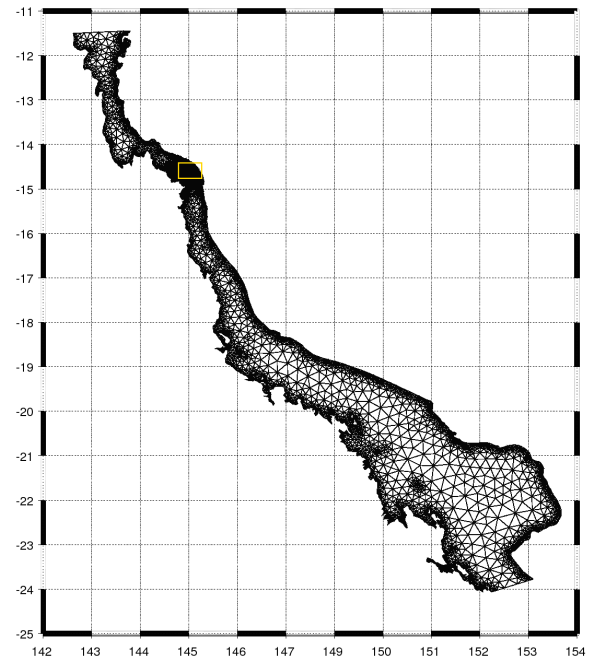


Fig. 14. Coarse mesh of the Great Barrier Reef refined around Lizard Island.

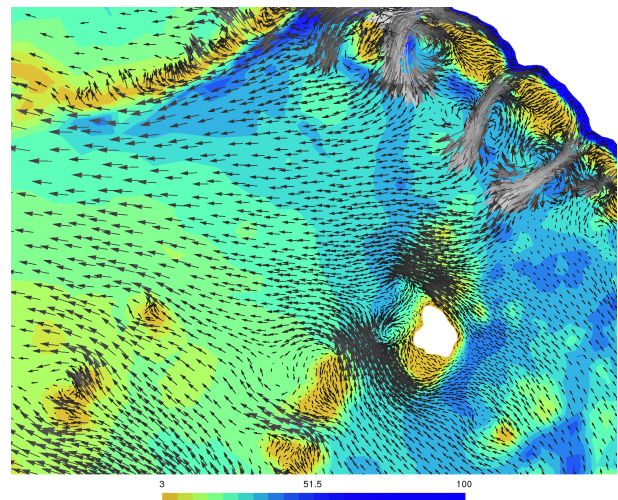


Fig. 15. Details of the simulation computed on the mesh of Figure 14. Color levels show the depth and the arrows indicate the bidimensional velocity field.

self consistent and portable geometric model. Existing robust meshing procedures can be applied to the CAD model. Various meshes can be constructed based on the same CAD definition and various meshing algorithms can be used as well. Last, but not least, everything that has been described in this paper is now part of Gmsh, a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities (<http://www.geuz.org/gmsh>, Geuzaine and Remacle (2008)). Gmsh being open source (under the GNU General Public License), anyone within the finite element marine modeling community has the opportunity to use this freely.

Acknowledgements

The present study was carried out within the scope of the project *A second-generation model of the ocean system*, which is funded by the *Communauté Française de Belgique*, as *Actions de Recherche Concertées*, under contract ARC 04/09-316. This work is a contribution to the SLIM² project.

References

- S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998. <http://www.cs.umd.edu/~mount/ANN/>.
- M. W. Beall and M. S. Shephard. A general topology-based mesh data structure. *International Journal for Numerical Methods in Engineering*, 40(9):1573–1596, 1997.
- Sergey Danilov, Gennady Kivman, and Jens Schröter. Evaluation of an eddy-permitting finite-element ocean model in the north atlantic. *Ocean Modelling*, 10:35–49, 2005.
- R. A. Dwyer. A simple divide-and-conquer algorithm for computing delaunay triangulations in $O(n \log \log n)$ expected time. In *Proceedings of the second annual symposium on Computational geometry*, pages 276–284, 1986.
- P.-L. George and P. Frey. *Mesh Generation*. Hermes, 2000.
- C. Geuzaine and J.-F. Remacle. Gmsh: a finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 2008. submitted, <http://www.geuz.org/gmsh>.
- G.J. Gorman, M.D. Piggott, C.C. Pain, R.E. de Oliveira, A.P. Umpleby, and A.J.H. Goddard. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Modelling*, 12:436–452, 2006.
- G.J. Gorman, M.D. Piggott, and C.C. Pain. Shoreline approximation for unstructured mesh generation. *Computers and Geosciences*, 33:666–677, 2007.
- Stephen M. Griffies, Claus Böning, Frank O. Bryan, Eric P. Chassignet, Rüdiger Gerdes, Hiroyasu Hasumi, Anthony Hirst, Anne-Marie Treguier, and David Webb. Developments in ocean climate modeling. *Ocean Modelling*, 2:123–192, 2000.
- S. C. Hagen, J. J. Westerink, R. L. Kolar, and O. Horstmann. Two-dimensional, unstructured mesh generation for tidal models. *International Journal for Numerical Methods in Fluids*, 35:669–686, 2001. Printed version in Richard’s office.
- R. Haimes. CAPRI: Computational analysis programming interface (a solid modeling based infra-structure for engineering analysis and design). Technical report, Massachusetts Institute of Technology, 2000.
- R. F. Henry and R. A. Walters. Geometrically based, automatic generator for irregular triangular networks. *Communications in numerical methods in engineering*, 9:555–566, 1993.
- J. Lambrechts, E. Hanert, E. Deleersnijder, P.-E. Bernard, V. Legat, and J.-F. Remacle E. Wolanski. A high-resolution model of the whole great barrier reef hydrodynamics. *Estuarine, Coastal and Shelf Science*, 79(1):143–151, 2008. doi 10.1016/j.ecss.2008.03.016.
- C. Le Provost, M. L. Genco, and F. Lyard. Spectroscopy of the world ocean tides from a finite element hydrodynamic model. *Journal of Geophysical Research*, 99:777–797, 1994.
- Sebastien Legrand, Vincent Legat, and Eric Deleersnijder. Delaunay mesh generation for an unstructured-grid ocean circulation model. *Ocean Modelling*, 2:17–28, 2000.
- Sebastien Legrand, Eric Deleersnijder, Emmanuel Hanert, Vincent Legat, and Eric Wolanski. High-resolution, unstructured meshes for hydrodynamic models of the great barrier reef, australia. *Estuarine, Coastal and Shelf Science*, 68:36–46, 2006.
- O. Lietaer, T. Fichefet, and V. Legat. The effects of resolving the Canadian Arctic Archipelago in a finite element sea ice model. *Ocean Modelling*, 2008, in press, doi 10.1016/j.ocemod.2008.06.002.
- Florent Lyard, Fabien Lefevre, Thierry Letellier, and Olivier Francis. Modelling the global ocean tides: modern insights from FES2004. *Ocean Dynamics*, 56:394–415, 2006.
- National Geographic Data Center, 2006. <http://www.ngdc.noaa.gov/mgg/global/global.html>.
- M.D. Piggott, G.J. Gorman, and C.C. Pain. Multi-scale ocean modelling with adaptive unstructured grids. *CLIVAR Exchanges - Ocean model development and assessment*, 12(42), 2007. <http://eprints.soton.ac.uk/47576/>.
- S. Rebay. Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm. *Journal of Computational Physics*, 106:25–138, 1993.
- N. P. Weatherill. The integrity of geometrical boundaries in the two-dimensional delaunay triangulation. *Communications in Applied Numerical Methods*, 6(2):101–109, 1990.
- P. Wessel and W. H. F. Smith. A global self-consistent, hierarchical, high-resolution shoreline database. *Journal of Geophysical Research*, 101, B4:8741–8743, 1996. <http://www.soest.hawaii.edu/wessel/gshhs/gshhs.html>.
- Laurent White, Eric Deleersnijder, and Vincent Legat. A three-dimensional unstructured mesh finite element shallow-water model, with application to the flows around an island and in a wind-driven elongated basin. *Ocean Modelling*, in press, 2008. doi: doi:10.1016/j.ocemod.2008.01.001.

² SLIM, Second-generation Louvain-la-Neuve Ice-ocean Model, <http://www.climate.be/SLIM>