Ocean Dynamics (2008) 58:461–473
DOI 10.1007/s10236-008-0148-3

Multiscale mesh generation on the sphere

Jonathan Lambrechts · Richard Comblen · Vincent Legat · Christophe Geuzaine · Jean-François Remacle

Received: 15 May 2008 / Accepted: 8 September 2008 / Published online: 8 October 2008

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.

Keywords Mesh generation · Sphere · Ocean modeling

Responsible Editor: Pierre Lermusiaux

J. Lambrechts (✉) · R. Comblen · V. Legat · J.-F. Remacle
Institute for Mechanical, Material and Civil Engineering,
Université Catholique de Louvain,
Louvain-la-Neuve, Belgium
e-mail: jonathanlambrechts@gmail.com

C. Geuzaine
Electrical Engineering and Computer Science,
Montefiore Institute, Liège, Belgium

1 Introduction

Finite elements have been used in engineering analysis for several decades. Since the 1990s, geometric domains that are used in finite element analysis and design have been 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 difference 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). On 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 S centered at the origin and of radius R of about 6,370 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 are available, it cannot be envisaged to build a BRep with the maximal available resolution everywhere. For example, the current global shoreline database has a resolution of about 50 m, 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., Beall and Shephard 1997; Haimes 2000). More specifically, some of the authors of this paper have developed Gmsh: a 3D finite element mesh generator with built-in pre- and post-processing facilities. 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 3D model can be defined using its 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 entity* are used: *model vertices* G_i^0 (dimension 0), *model edges* G_i^1 (dimension 1), and *model surfaces* 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 geometries 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:

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

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

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

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

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

As an illustration, let us consider the surface represented in Fig. 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

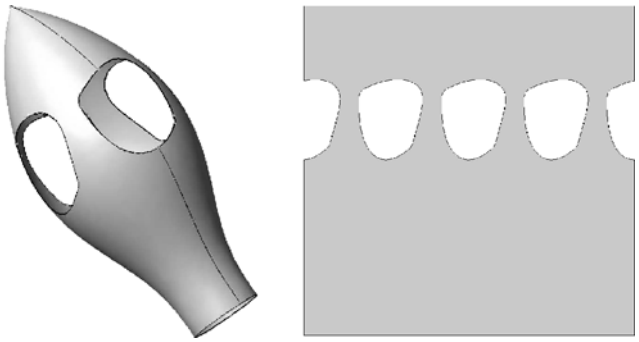


Fig. 1 A model surface in real (*left*) and parametric (*right*) coordinates. The seam of the surface is highlighted in the left plot

sphere—i.e., a surface that is periodic, that is bounded by continents and islands and that has degeneracies at both poles.

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 suffer 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 Fig. 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 with most of the parametrizations, are not conformal. A conformal mapping will conserve the angle at which curves cross each other. Consequently, in order to obtain an isotropic mesh in 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 Fig. 2. Anyway, it is always better to use a conformal mapping, such as the stereographic projection of Fig. 3.

Let us consider a sphere $\mathcal{S}$ centered at the origin and of radius R, and one point $\boldsymbol{s}$. This point lies on the surface of the sphere, does no belong to the oceans, and will be the only singular point of the mapping. A suitable choice for $\boldsymbol{s}$ could be a location in the middle of Kazakhstan, but here, we choose $\boldsymbol{s} = \{0, 0, -R\}$. 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 $\boldsymbol{p}$ of the sphere on the plane $z = R$.

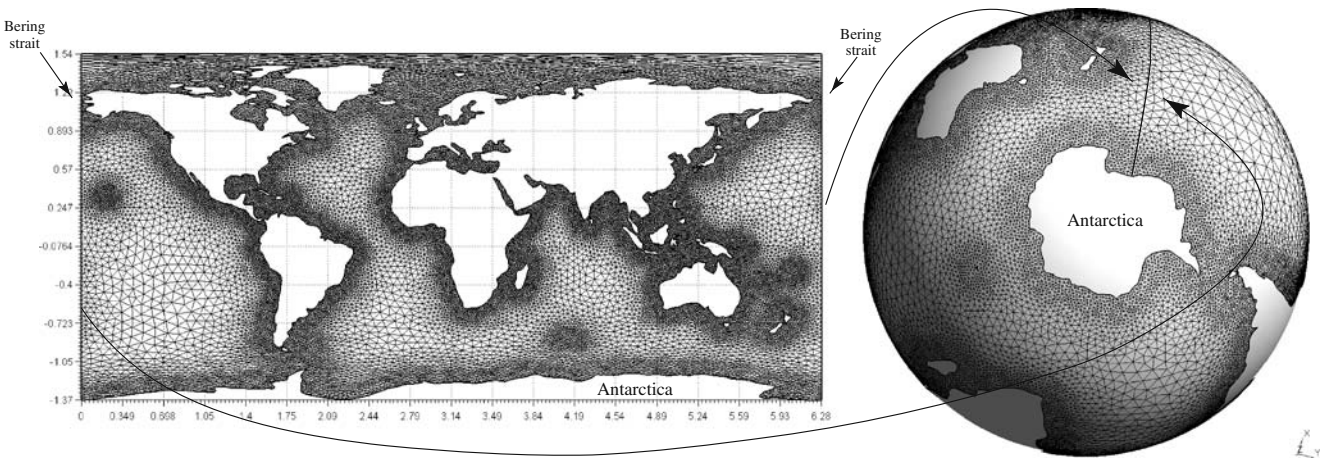


Fig. 2 Mesh of the World Ocean using the spherical coordinate system. The seam edge is visible on the right plot

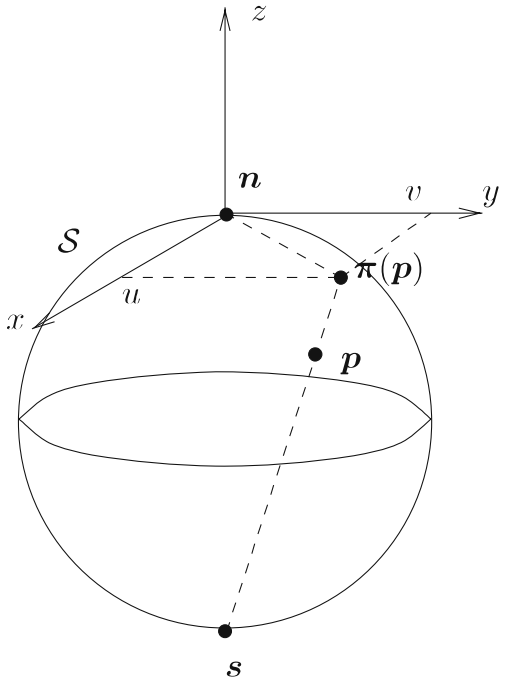


Fig. 3 Stereographic projection

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

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

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

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 the Global Self-consistent Hierarchical High-resolution Shorelines (GSHHS, (Wessel and Smith 1996)). This data set describes 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 $\gamma(\boldsymbol{x})$ that expresses the geometrical requested accuracy of the model at any point $\boldsymbol{x}$. The GSHHS data set is coarsened with respect to this size field and every successive point at $\boldsymbol{x}$ that is closer than $\gamma(\boldsymbol{x})$ is 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

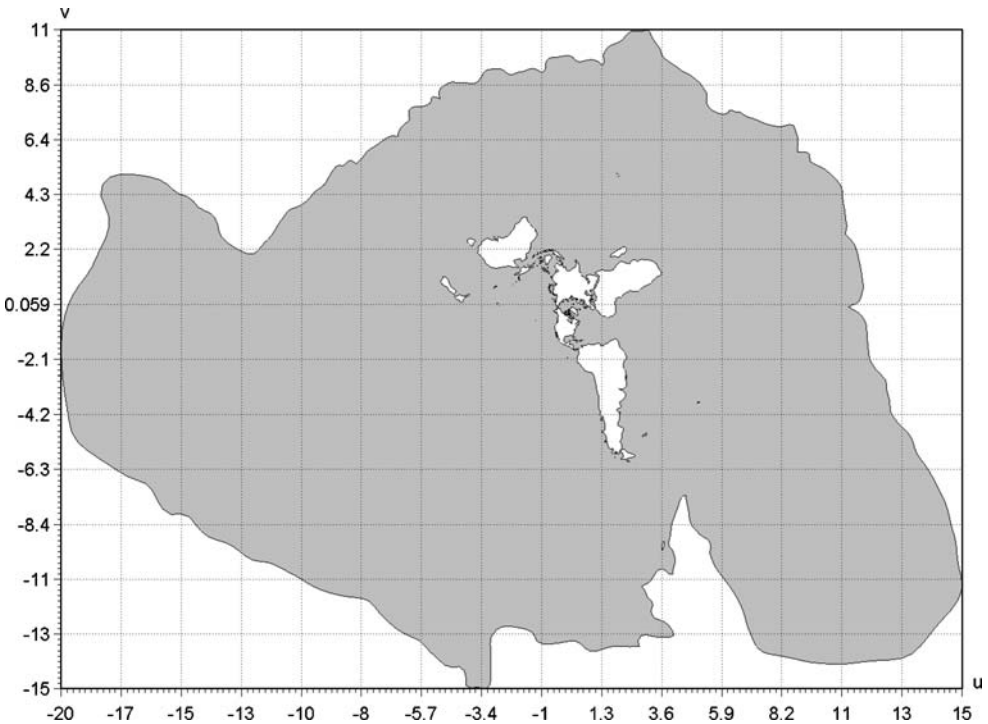


Fig. 4 The World Ocean in stereographic coordinates. The North Pole is the center, Antarctica constitutes the external boundary

Fig. 5 Great Britain and Ireland with resolutions 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

B-splines remain inside the convex hull defined by the control points, it can be shown that, if the piecewise linear representation is convex and consistent, then the curvilinear B-splines representation is also consistent.

In Fig. 5, we generate the coastlines of Great Britain and Ireland with different resolutions. With a resolution $\gamma = 100$ 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 2,176 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 involves 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 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 $\delta(\boldsymbol{x})$ that defines the optimal length of an edge at position $\boldsymbol{x}$ 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 $H(\boldsymbol{x})$ 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 $\lambda_{\min}$ is the smallest wavelength that has to be captured for a reference bathymetry H_{ref}, we define f_1 as

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

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 $\lambda_{\max}$, the greatest (in absolute value) eigenvalue of the Hessian $\mathcal{H}(x)$,

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

we define the second field as:

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

In order to represent coastlines well and to capture the small-scale phenomena 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 $f_3(\boldsymbol{x})$ as the distance to the closest shoreline:

$$f_3(\boldsymbol{x}) = d(\boldsymbol{x}).$$

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

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

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

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

where

$$\alpha_i(\boldsymbol{x}) = \begin{cases} 0 & \text{if} \quad f_i(\boldsymbol{x}) \le f_i^{\min} \\ \dfrac{f_i(\boldsymbol{x}) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} & \text{if} \quad f_i^{\min} < f_i(\boldsymbol{x}) < f_i^{\max} \\ 1 & \text{if} \quad f_i(\boldsymbol{x}) \ge f_i^{\max} \end{cases}$$

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

$$\delta(\boldsymbol{x}) = \min\left(\delta_1(\boldsymbol{x}), \delta_2(\boldsymbol{x}), \dots\right).$$

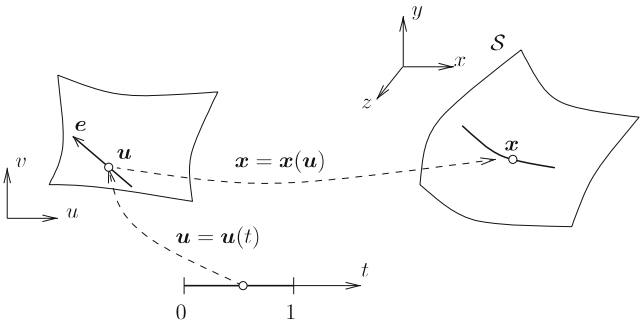


Fig. 6 Three parametrizations of a straight edge

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

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

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

where $\mathbf{J} = \partial\boldsymbol{x}/\partial\boldsymbol{u}$ 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:

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

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

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

3.2 Coastlines mesh generation

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

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

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

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

Integration of those expressions must be performed with an 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 20 thousand control points (Fig. 4).

However, this algorithm does not guarantee that, even if the model edges G_j^1 that constitute the boundaries of the domain are nonintersecting, the corresponding 1D meshes do not self-intersect. Figure 7 shows two islands very close to each other. Yet, even if 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 islands are to be involved. It is therefore mandatory to define a systematic recovery procedure. Such an algorithm, illustrated in Fig. 7, works as follows:

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 two 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 nonslip boundary condition is applied along their coastlines.

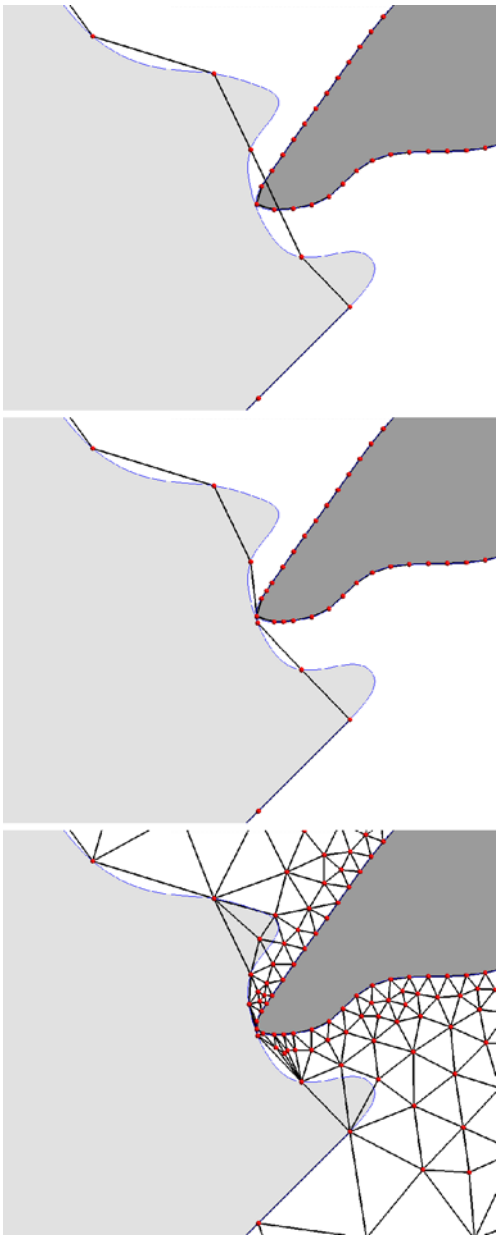


Fig. 7 A geometry with two islands (in *light* and *dark gray*) that are very close to each other. The *top image* shows the initial 1D mesh that respects mesh size field. The *middle image* shows the first iteration of the recovery algorithm. The *bottom image* shows the final mesh that was possible to realize after two recovery iterations

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 differently in the three algorithms:

– The *del2d* algorithm is inspired by 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 that of algorithm *del2d*.
- The *meshadapt* algorithm is very different from the first two ones. It is based on local mesh modification: 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.

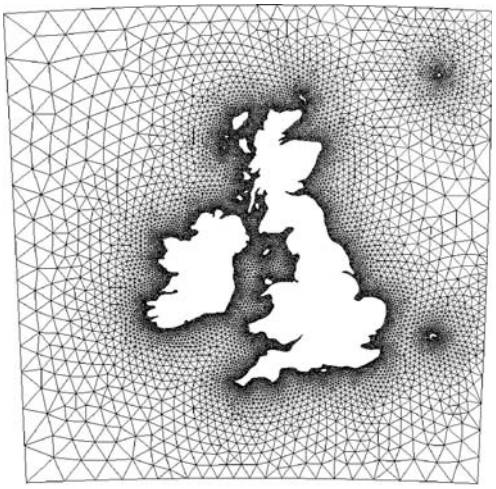
The *frontal* algorithm usually gives the highest-quality meshes while the *del2d* algorithm is the fastest: it produces about five million triangles a minute if the size field δ is not too complex to compute. Figure 8 presents three meshes of one of the models of Fig. 5 for which we have used a shore proximity function as the only size field. Meshes have, respectively, 18,698, 19,514, and 17,154 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, 0.5, and 5.7 s.

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

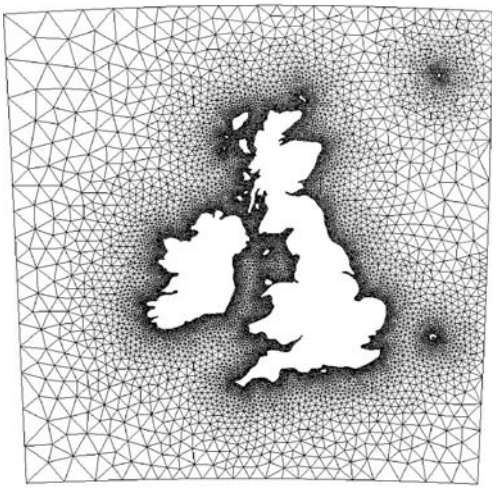
- A shore proximity function f_1 is used with $\delta_1^{\text{small}} = 30$ km, $\delta_1^{\text{large}} = 200$ km, $f_1^{\min} = 0$, and $f_1^{\max} = 500$ km.
- We use f_2 and refine the mesh proportionally to the square root of the ocean depth. The size field δ_2 ranges from 25 to 500 km.
- We use f_3 to capture the bathymetry. The size field δ_3 ranges from 25 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 s on a recent laptop. Those timings compare advantageously

a) Mesh done using the *frontal* algorithm



b) Mesh done using the *del2d* algorithm



c) Mesh done using the *meshadapt* algorithm

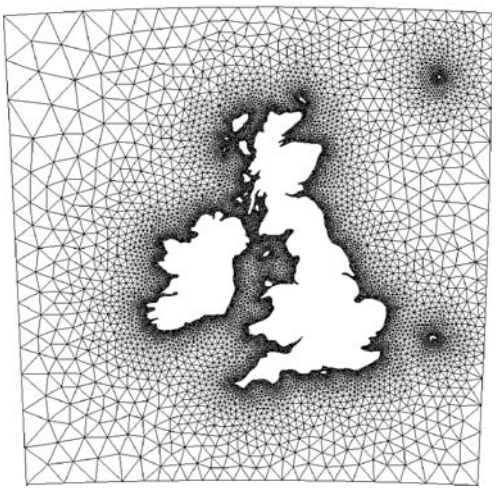


Fig. 8 Meshes of the same domain using three different algorithms (**a**–**c**)

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

with alternative techniques based on mesh decimation (Gorman et al. 2006). The memory footprint of the meshing algorithms is low: about 12 million triangles (six 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 by some simulation results.

4.1 Sea ice modeling

The mesh presented in Fig. 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 constitutes 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 in Fig. 12. A complete description of the model and its validation can be found in Lietaer et al. (2008).

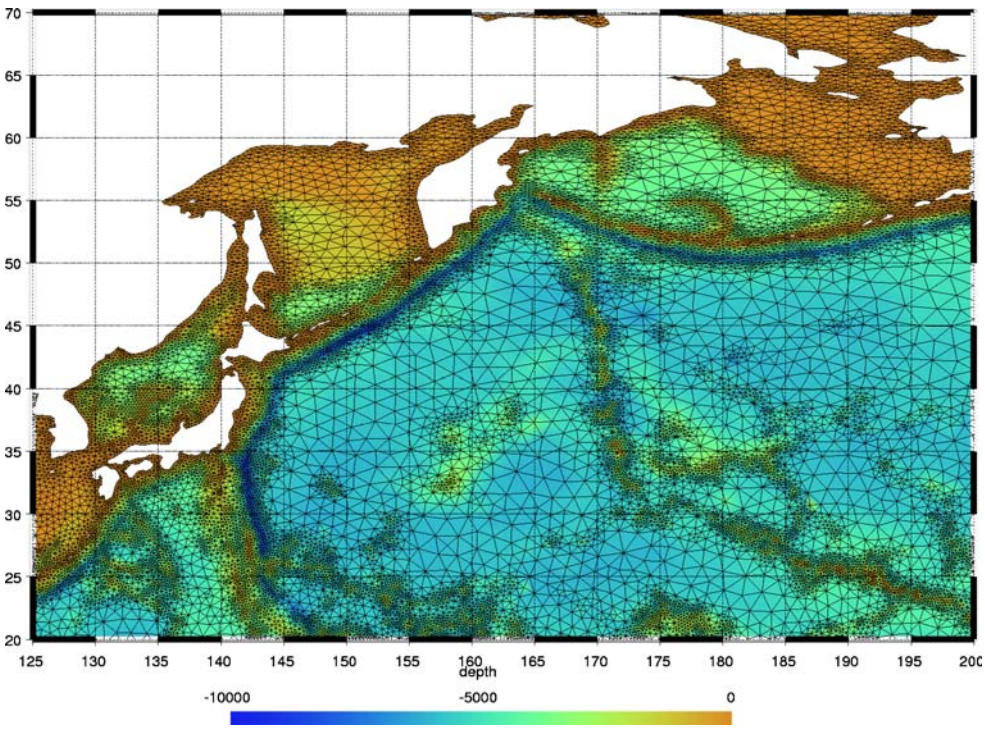


Fig. 10 Close up of the mesh of Fig. 9 in the north Pacific Ocean; *color levels* represent the bathymetry (in meters). The effects of the three refinement rules are clearly visible

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

4.2 Multiscale model of the Scheldt River

Within the framework of the multidisciplinary project TIMOTHY,[1] our team is presently involved in the development of a 2D 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 fecal 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 1,000 km away from the mouth of the estuary, and the upstream boundary is situated in the areas of Antwerp, where the river width is a few hundred meters. The multiscale character of the problem is then one of its main feature and the use of an unstructured grid is thus totally appropriate. Figure 13 presents a mesh of 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 sizes range from 150 m in the Scheldt river near Antwerp to about 50 km away 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 northeastern coastline. There are over 2,500 coral reefs in a strip that is about 2,600 km in length and 200 km in width. The simulation that is presented here makes use of most of 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, and special refinement in the domain of interest. A complete simulation is described in Lambrechts et al. (2008). Figure 14 shows a mesh built to run small simulations 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 interreef passages and recirculations around islands. Elsewhere on the shelf, the resolution is very coarse, those regions are only used as boundary conditions. Seventy percent of the 20,384 elements are

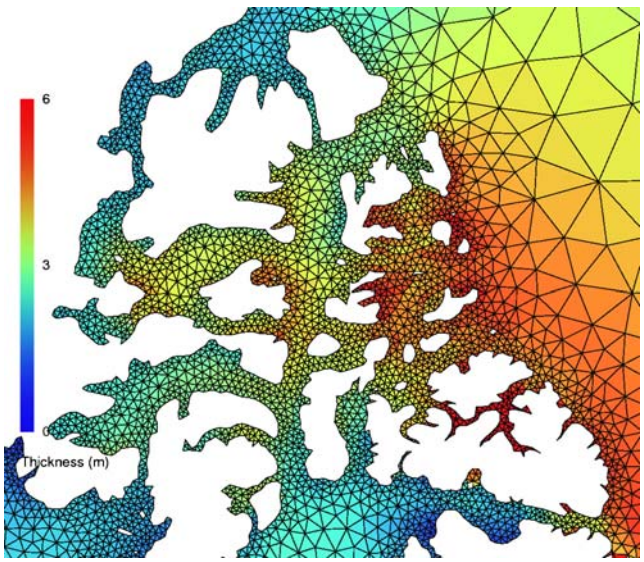


Fig. 12 Detail of the mesh of Fig. 11. Mean March sea ice thickness pattern (in meters) in the western part of the Canadian Arctic Archipelago as computed by the finite element sea ice model (1979–2005)

[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.

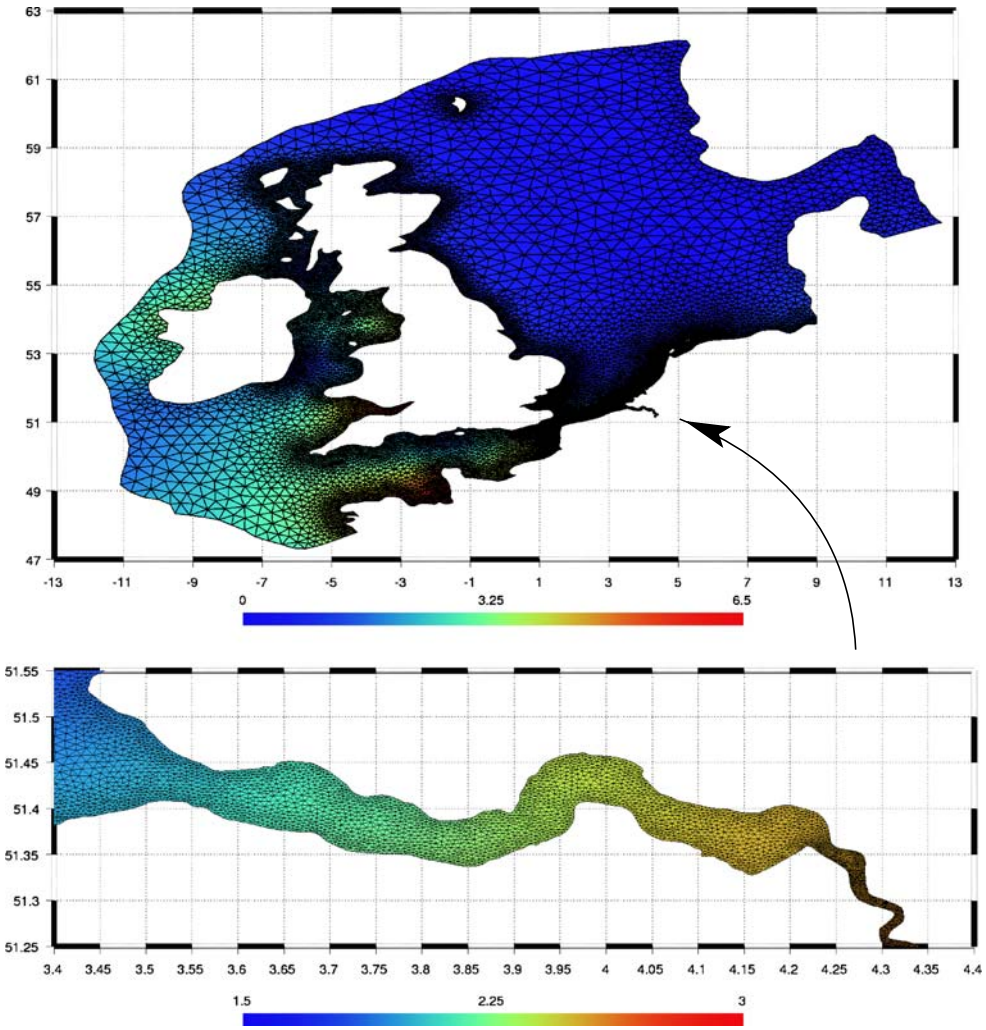


Fig. 13 Multiscale mesh: North Sea and Scheld River Estuary. *Color levels* represent the amplitude of the M2 tidal component (in meters)

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 Summary

A CAD-based mesh generation procedure for ocean modeling has been developed. The new approach has the advantage of relying on existing well-known engineering mesh generation procedures. The CAD model, based on a smooth BRep of the domain, allows to build a compact, 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 3D finite element mesh generator with built-in pre- and postprocessing facilities (http://www.geuz.org/gmsh). Since Gmsh is 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 SLIM[2] project.

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

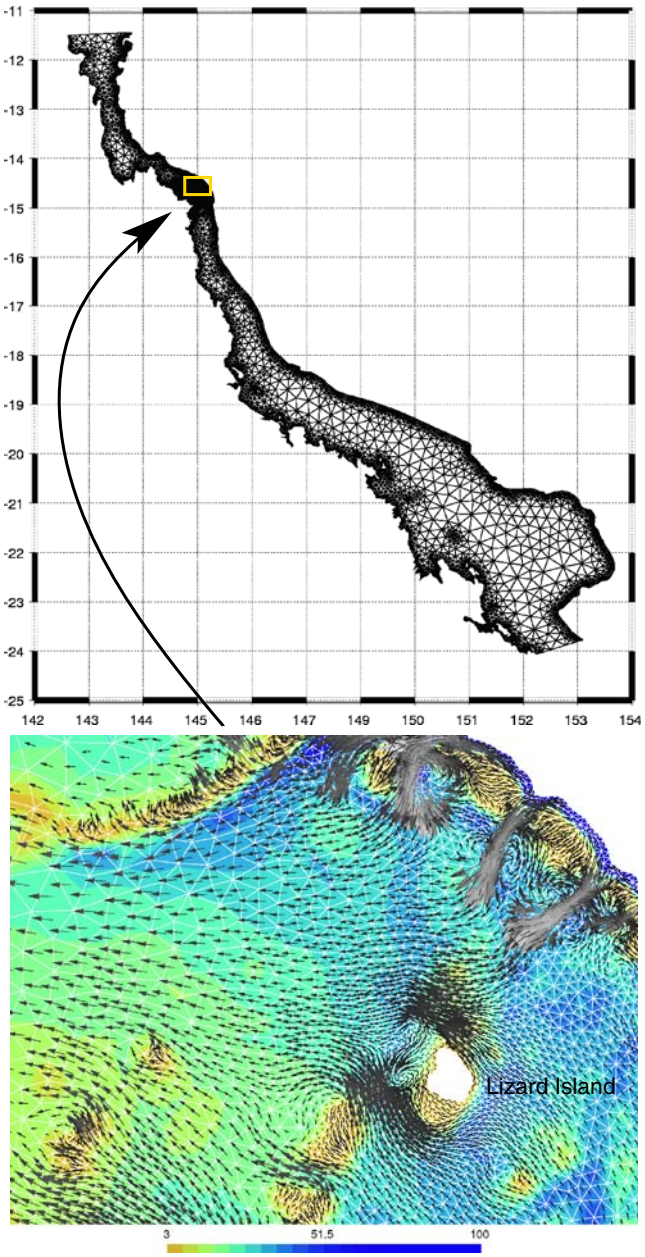


Fig. 14 Coarse mesh of the Great Barrier Reef refined around Lizard Island (*top*). Details of the simulation computed on this mesh in the vicinity of this island (*bottom*). *Color levels* show the depth and the *arrows* indicate the bidimensional velocity field

References

Arya S, Mount DM, Netanyahu NS, Silverman R, Wu AY (1998) An optimal algorithm for approximate nearest neighbor searching. J ACM 45:891–923. http://www.cs.umd.edu/ mount/ANN/

Beall MW, Shephard MS (1997) A general topology-based mesh data structure. Int J Numer Methods Eng 40(9):1573–1596

Danilov S, Kivman G, Schröter J (2005) Evaluation of an eddy-permitting finite-element ocean model in the north atlantic. Ocean Model 10:35–49

Dwyer RA (1986) 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, Yorktown Heights, 2–4 June 1986, pp 276–284

George P-L, Frey P (2000) Mesh generation. Hermes, Lyon

Gorman G, Piggott M, Pain C (2007) Shoreline approximation for unstructured mesh generation. Comput Geosci 33: 666–677

Gorman G, Piggott M, Pain C, de Oliveira R, Umpleby A, Goddard A (2006) Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. Ocean Model 12:436–452

Griffies SM, Böning C, Bryan FO, Chassignet EP, Gerdes R, Hasumi H, Hirst A, Treguier A-M, Webb D (2000) Developments in ocean climate modeling. Ocean Model 2:123–192

Hagen SC, Westerink JJ, Kolar RL, Horstmann O (2001) Two-dimensional, unstructured mesh generation for tidal models. Int J Numer Methods Fluids 35:669–686 (printed version in Richard's office)

Haimes R (2000) CAPRI: computational analysis programming interface (a solid modeling based infra-structure for engineering analysis and design). Tech. rep., Massachusetts Institute of Technology

Henry RF, Walters RA (1993) Geometrically based, automatic generator for irregular triangular networks. Commun Numer Methods Eng 9:555–566

Lambrechts J, Hanert E, Deleersnijder E, Bernard P-E, Legat V, Wolanski J-FRE (2008) A high-resolution model of the whole great barrier reef hydrodynamics. Estuar Coast Shelf Sci 79(1):143–151. doi:10.1016/j.ecss.2008.03.016

Le Provost C, Genco ML, Lyard F (1994) Spectroscopy of the world ocean tides from a finite element hydrodynamic model. J Geophys Res 99:777–797

Legrand S, Deleersnijder E, Hanert E, Legat V, Wolanski E (2006) High-resolution, unstructured meshes for hydrodynamic models of the Great Barrier Reef, Australia. Estuar Coast Shelf Sci 68:36–46

Legrand S, Legat V, Deleersnijder E (2000) Delaunay mesh generation for an unstructured-grid ocean circulation model. Ocean Model 2:17–28

Lietaer O, Fichefet T, Legat V (2008) The effects of resolving the Canadian Arctic Archipelago in a finite element sea ice model. Ocean Model 24:140–152. doi:10.1016/j.ocemod.2008.06.002

Lyard F, Lefevre F, Letellier T, Francis O (2006) Modelling the global ocean tides: modern insights from FES2004. Ocean Dyn 56:394–415

National Geographic Data Center (2006) ETOPO1 global relief model. http://www.ngdc.noaa.gov/mgg/global/global.html.

Piggott M, Gorman G, Pain C (2007) Multi-scale ocean modelling with adaptive unstructured grids. CLIVAR Exch

Ocean Model Dev Assess 12(42):21–23 (http://eprints.soton.ac.uk/47576/)

Rebay S (1993) Efficient unstructured mesh generation by means of delaunay triangulation and Bowyer-Watson algorithm. J Comput Phys 106:25–138

Weatherill NP (1990) The integrity of geometrical boundaries in the two-dimensional delaunay triangulation. Commun Appl Numer Methods 6(2):101–109

Wessel P, Smith WHF (1996) A global self-consistent, hierarchical, high-resolution shoreline database. J Geophys Res 101(B4):8741–8743. http://www.soest.hawaii.edu/wessel/gshhs/gshhs.html

White L, Deleersnijder E, Legat V (2008) 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 Model 22:26–47