

# Mesh generation in archipelagos

Arjen Douwe Terwisscha van Scheltinga ·  
Paul Glen Myers · Julie D. Pietrzak

Received: 12 December 2011 / Accepted: 1 June 2012 / Published online: 30 June 2012  
© Springer-Verlag 2012

**Abstract** A new mesh size field is presented that is specifically designed for efficient meshing of highly irregular oceanic domains: archipelagos. The new approach is based on the standard mesh size field that uses the proximity to the nearest coastline. Here, the proximities to the two nearest coastlines are used to calculate the distance between two islands or the width of a strait through an archipelago. The local value of the mesh size field is taken as the width (or distance between two islands) divided by the number of required elements across the strait (or between the islands). This new mesh size fields are illustrated for three examples: (1) the Aegean Sea, (2) the Indonesian Archipelago, and (3) the Canadian Arctic Archipelago.

**Keywords** Unstructured meshes · Mesh generation · Multi-scale modeling · Oceanography

---

Responsible Editor: Mohamed Iskandarani

---

This article is part of the Topical Collection on *Multi-scale modelling of coastal, shelf and global ocean dynamics*

---

A. D. Terwisscha van Scheltinga · P. G. Myers  
Department of Atmospheric and Earth Sciences,  
University of Alberta, Edmonton, Alberta, Canada

J. D. Pietrzak  
Faculteit CTIG, Technische Universiteit Delft,  
Delft, The Netherlands

*Present Address:*

A. D. Terwisscha van Scheltinga (✉)  
School of Earth Sciences, University of Bristol, Bristol, UK  
e-mail: gladtv@s@bristol.ac.uk

## 1 Introduction

The world's oceans are bounded by continents and island coastlines, resulting in complex and irregular geometry on many spatial scales. There are large ocean basins bordered by the continents which have islands in their interior. There are archipelagos, separating the ocean basin and lesser bodies of water, which allow for interbasin transports of water masses. For example, the Indonesian throughflow transports water from the Pacific to the Indian Ocean through the Indonesian Archipelago and is an important component of the global overturning circulation (Godfrey 1996; Lee et al. 2002) and global climate system (Schneider 1998). Another example is the Canadian Arctic Archipelago (CAA). Freshwater is transported from the Arctic Ocean through the CAA to the Labrador Sea and thus to the Atlantic Ocean.

For many applications, well-represented coastlines are an essential component of ocean modeling. Traditionally, ocean models are based on finite difference schemes on structured grids (Griffies et al. 2000). The disadvantage of this approach is that representing coastlines and geometric features like narrow straits and small island is not straightforward. A sufficient resolution in these narrow straits, or near these small islands, can only be achieved by increasing the resolution of the structured grid, greatly increasing the computational overhead. The alternative, sufficient resolution for most of the domain and not representing small islands or narrow straits, is computationally cheaper but does not represent the small-scale dynamics that might be important.

An unstructured mesh approach circumvents this issue, since it has the advantage of accurately repre-

senting complex and irregular coastlines. In addition, it allows for a spatially varying mesh resolution (e.g., Adcroft and Marshall 1998; Legrand et al. 2006). One approach to unstructured meshes is the finite element method, which has been used for several decades in engineering. In the past two decades, geometrical domains that are used for design and finite elements are built using computer-aided design (CAD). Most complex features like machine parts or assemblies can be easily and efficiently be dealt with by CAD. Finite elements and unstructured meshes have been used in ocean modeling (e.g., Danilov et al. 2005; Piggott et al. 2007; White et al. 2008).

In generating the unstructured mesh for ocean modeling, first a suitable boundary has to be defined, i.e., the boundary representation (BRep): shorelines in the horizontal direction and bathymetry in the vertical direction. Inadequate representation of the shoreline may lead to problems (Pain et al. 2005; Adcroft and Marshall 1998), such as spurious stresses in case of staircase shorelines present in structured meshes. Gorman et al. (2007) proposed a method to approximate the shorelines with polygons and polylines. Gorman et al. (2006) described an automatic procedure that builds a BRep of the geometry of the world ocean within prescribed accuracy. Both use the high resolution shoreline data bases and bathymetry data.

Using mesh generation algorithms, a mesh is produced from the BRep. At first, these algorithms were adapted from classical engineering tools. The mesh generation software of Henry and Waters (1993) was used by Le Provost et al. (1994) to generate a mesh of the world ocean used for tidal modeling. The same kind of meshes at a higher resolution was used by Lyard et al. (1994). On regional scales, Legrand et al. (2006) produced high-resolution meshes of the Great Barrier Reef. For generation of meshes in coastal regions, Hagen et al. (2001) developed two algorithms, which were used for tidal modeling in the Gulf of Mexico. Specific algorithms were designed for the world ocean by Legrand et al. (2000), Gorman et al. (2007).

These algorithms use definitions of a local mesh size field to control the size and shape of the elements. Definitions that can be used are proximity to the nearest shore, bathymetry, or various error estimates. Depending on the geometry, several fields can be used simultaneously. Usually, the minimum over all these fields is taken. For example, for each coast (or group of coastlines), a mesh size field can be defined based on the proximity to this coastline (or group of coastlines). For most applications, defining a small number of these fields is sufficient. However, when dealing with archipelagos, this becomes more difficult. Archipelagos

generally consist of a large number of islands with irregular coastlines and with a wide range of sizes. The islands are separated by straits of different lengths and varying widths. To obtain a mesh of sufficient resolution using mesh size fields based on the proximity to the nearest shoreline is difficult since one would need a lot of different mesh size fields in order to obtain a sufficient resolution in each strait. This is computationally expensive and fine-tuning all these fields is labor intensive, even when only the most important straits are refined.

Here, we propose an algorithm that automates the meshing generation of irregular geometries of archipelagos. The algorithm is designed such that it generates a mesh in selected areas in the archipelago with a desired resolution. It only needs a collection of coastlines, and no additional fine-tuning for each shoreline in this collection is necessary, since the mesh size fields based on the proximity to the shorelines in this collection has been replaced by a new single mesh size field. This field only depends on the desired resolution and the proximity to the nearest shoreline in this collection. In addition, this field converges asymptotically to any given background mesh field.

The paper is divided into three sections. First, we outline the procedure for constructing a boundary representation and generating a mesh. In the second section, we describe the new algorithm. In the last section, we show some examples of meshes produced by this algorithm.

## 2 Mesh generation on a sphere

In this section, we will give an overview of the procedures for building geometrical models of the ocean and mesh generation. A more thorough discussion is presented in Lambrechts et al. (2008). The discussion here is limited only to those topics needed for discussion in later sections.

### 2.1 Geometric model

Before a mesh can be generated, a geometric model of the ocean must be defined. This geometric model of the ocean, 2D or 3D, can be represented by its boundary representation. A curve is bounded by two points, a surface by curves and a volume by surfaces. Therefore, the model consists of model entities: 2D model surfaces, 1D model edges, and 0D model vertices. Usually, a parametrization of these shapes is available and often is mappings.

In this geometric model, a coastline is usually defined as a periodic curve. Though several approaches are available, like using a piecewise linear interpolation, here we use cubic B-splines. This allows for more flexibility. The control points for the B-splines are taken from the Global Self-consistent Hierarchical High-resolution Shorelines database (GSHHS, Wessel and Smith 1996). This is the most accurate shoreline database, and is guaranteed to be self-consistent, i.e., in this database, coastlines do not intersect. A size field  $\gamma(\mathbf{x})$  can be defined that gives the desired accuracy of the geometrical model at any point  $\mathbf{x}$ . The GSHHS data are coarsened by collapsing every successive point at  $\mathbf{x}$  that is closer than  $\gamma(\mathbf{x})$ .

There are several choices for parameterizing a sphere. In oceanography, most of the time geographic or spherical coordinates are used. The mapping of the coordinate system might have some particular problems: (1) a singular point might exist, (2) one of the coordinates might be periodic, (3) shorelines might cross a periodic edge, and (4) the mapping does not conserve angles, i.e., the mapping is not conformal. As a result, this leads to complexity in the definition of the geometric model. Though a conformal mapping is preferred since it conserves angles, there might be applications where a parametrization is not conformal and anisotropic mesh has to be constructed in the parametric plane to obtain an isotropic mesh in real space.

### 2.2 Local mesh size

After the boundaries have been represented by B-splines, finite element meshes can be generated on the surfaces. Usually, the mesh generator builds elements, by adapting the mesh size field to control their size and shape. This mesh size field is a scalar function  $\delta(\mathbf{x})$  that defines the optimal size of an edge at position  $\mathbf{x}$ . For the world oceans, some choices for the  $\delta$  exist (Lambrechts et al. 2008).

The bathymetry  $H(\mathbf{x})$  can be taken into account by forcing the mesh to capture its variations. Since the bathymetry is usually interpolated piecewise linearly at the mesh vertices, the first term in its error is dependent on the absolute maximum eigenvalue  $\lambda_{\max}$  of the Hessian

$$\mathcal{H}(\mathbf{x}) = \nabla^2 \left( \frac{H(\mathbf{x})}{H_{\text{ref}}} \right), \tag{1}$$

where  $H_{\text{ref}}$  is a reference bathymetry. This leads to a criterion field  $f_1(\mathbf{x}) = 1/\sqrt{\lambda_{\max}}$ .

In addition, the mesh can be forced to capture for a reference depth the wavelength of the gravity waves, which travel at speed  $\sqrt{gH}$ , with  $g$  being acceleration

due to gravity. The length scale of a gravity waves  $\lambda$  is proportional to  $\mathcal{O}(1/\sqrt{H})$ . If one wave with a minimum wavelength  $\lambda_{\min}$  is to be captured by  $N$  mesh sizes for a given reference bathymetry  $H_{\text{ref}}$ , a criterion function  $f_2(\mathbf{x})$  can be designed as

$$f_2(\mathbf{x}) = \frac{\lambda_{\min}}{N} \sqrt{\frac{H_{\text{ref}}}{H(\mathbf{x})}}. \tag{2}$$

Coastlines need to be resolved well, in order to capture small-scale phenomena. The mesh size should decrease towards the coastline. As such, the function  $f_3(\mathbf{x})$  can be defined that measures the proximity to the nearest coastline:

$$f_3(\mathbf{x}) = d(\mathbf{x}), \tag{3}$$

where  $d(\mathbf{x})$  is the distance to the nearest shoreline.

It is possible to add other size fields, such as error estimates dependent on the finite element solution. Also, to produce a mesh in parametric space with right sizing in real space, suitable mesh size fields should be defined in parametric space.

The mesh size field  $\delta_i(\mathbf{x})$  is now computed from the criterion field  $f_i(\mathbf{x})$  by assuming  $\delta_i(\mathbf{x})$  to be a linear function of  $f_i(\mathbf{x})$  within a refinement zone. Define the desired small and large mesh sizes  $\delta_i^{\text{small}}$  and  $\delta_i^{\text{large}}$  and the zone of refinement defined by the criterion field values  $f_i^{\text{min}}$  and  $f_i^{\text{max}}$ . The linear function  $\alpha_i(\mathbf{x})$  in the zone of refinement is given by

$$\alpha_i(\mathbf{x}) = \begin{cases} 0 & \text{if } f_i(\mathbf{x}) \leq f_i^{\text{min}} \\ \frac{f_i(\mathbf{x}) - f_i^{\text{min}}}{f_i^{\text{max}} - f_i^{\text{min}}} & \text{if } f_i^{\text{min}} < f_i(\mathbf{x}) < f_i^{\text{max}} \\ 1 & \text{if } f_i(\mathbf{x}) \geq f_i^{\text{max}} \end{cases} \tag{4}$$

The mesh size field  $\delta_i(\mathbf{x})$  follows from

$$\delta_i(\mathbf{x}) = \delta_i^{\text{small}} + \alpha_i(\mathbf{x}) \left( \delta_i^{\text{large}} - \delta_i^{\text{small}} \right). \tag{5}$$

The final mesh size field is taken to be the minimum of all size fields:  $\delta(\mathbf{x}) = \min(\delta_1(\mathbf{x}), \delta_2(\mathbf{x}), \dots)$ . In this paper, we will focus on defining a criterion function  $f$  and mesh size function  $\delta$  that refines the mesh in narrow straits. As such, we will not take into account bathymetry, though it can be taken into account by including a mesh size field based on the bathymetry to the minimum operator. Before discussing this refinement, a few words are needed on generating a meshing using the definition of the local mesh size.

### 2.3 1D and 2D mesh generation

For the meshes produced in this paper, we used Gmsh (Geuzaine and Remacle 2009), which includes the

definitions of criterion and mesh size functions above, as well as built-in pre- and postprocessing capabilities. Since the code is an open source, its source is available and can be extended and adapted.

For meshing a curve  $\mathbf{u} = [u_1, u_2]$ , it is parametrized by a parameter  $t : \mathbf{u}(t) : [0, 1] \rightarrow \mathcal{R}^2$ . Define the number of subdivisions  $N$  in the 1D mesh as

$$N = \int_0^1 \frac{\sqrt{(\partial_t u_1)^2 + (\partial_t u_2)^2}}{\delta(\mathbf{u}(t))} dt. \quad (6)$$

The coordinates of the  $N + 1$  mesh points are  $t_0, t_1, \dots$ , where  $t_i$  is given by

$$i = \int_{t_0}^{t_i} \frac{\sqrt{(\partial_t u_1)^2 + (\partial_t u_2)^2}}{\delta(\mathbf{u}(t))} dt. \quad (7)$$

Since the coastlines are represented by piecewise B-splines, an adaptive trapeze rule must be applied to evaluate these integrals. Even when the geometrical representation of the coastlines is nonintersecting, this algorithm does not guarantee that the 1D mesh is likewise nonintersecting. For example, two islands close together might result in a 1D mesh intersecting with itself. Given a large number of islands, resolving this issue by hand is not an option. A procedure to recover intersecting 1D meshes is available (see Lambrechts et al. 2008), which uses a divide-and-conquer algorithm to construct a Delaunay mesh. Missing edges are recovered using edge swaps and intersecting edges are split in two, after which a new Delaunay mesh is constructed. If necessary, this is repeated.

For 2D mesh generation, Gmsh provides several alternatives: (1) the del2d algorithm (George and Frey 2000), (2) the frontal algorithm (Rebay 1993), and (3) the meshadapt algorithm (Li et al. 2005). These methods start by forming an initial 1D Delaunay mesh, and vertices are iteratively inserted inside the domain. Though the del2d algorithm is the fastest, the frontal algorithm results in a high-quality mesh. Here, we have chosen to use the latter.

### 3 Refinement in narrow straits

#### 3.1 Sketch of the problem

Archipelagos are groups of islands of different sizes, and usually, their coastlines are highly irregular. As a result, they have many straits of varying length, and most straits have varying width. In such areas, meshing is not a straightforward exercise. If small-scale processes are to be studied, the mesh should be of sufficient resolution to resolve these processes. When the archipelago is embedded in a large region, at a

minimum, the straits that provided a large part of the through flow should have a resolution that captures any flux accurately.

How to obtain a mesh with sufficient resolution that takes in account these criteria? A straightforward approach is using the shore proximity function to refine the mesh where necessary. This has one limitation: one has to manually adjust the values  $\delta^{\text{small}}$ ,  $\delta^{\text{large}}$ ,  $f^{\text{min}}$ , and  $f^{\text{max}}$  for each coastline (island) in the strait. Given the possible large numbers of straits and their different widths, this might prove to be an inefficient and labor-intensive approach. Adding to the difficulty is that refinement in a strait might lead to refinement for the entire coastline, not just the relevant coastlines, of the islands in this strait if the coastline is represented by B-splines. This can be circumvented by splitting the B-spline in segments, each with different refinement. In addition, if an island is separating two straits and different resolutions are required, likewise, the B-spline needs to be split.

Rather, one would like a more systematic and efficient approach, where no manual editing of the coastline representation is needed. Here, we have chosen to use as our definition for the mesh size field the minimum of two functions: (1)  $f_0$ , based on the proximity of the coastline and (2)  $f_1$ , based on the width of the a strait. The former is a “background” mesh that is defined over the domain, with the latter refining the former where needed.

#### 3.2 Proximity to coastline

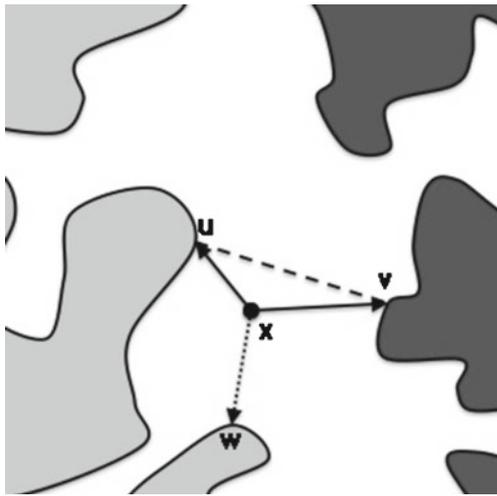
Let  $\Omega = \{\mathbf{u}_i | i = 1, \dots, L\}$  be the collection of  $L$  coastlines, where  $\mathbf{u}_i(t)$  is the  $i$ th parameterized coastline. In this context,  $\Omega$ , or any subset, is a collection of points belonging to the member coastlines. In the remainder, coastlines are assumed to be discretized coastlines, with a finite number of points. The function  $f_0$  is defined as the shortest distance to any shoreline:

$$f_0(\mathbf{x}) = d(\mathbf{x}) = \min_{\mathbf{u} \in \Omega} \|\mathbf{x} - \mathbf{u}\|. \quad (8)$$

The mesh size field  $\delta(\mathbf{x})$  is evaluated using Eqs. 4 and 5. Figure 1 shows a simple example of a strait. In this figure, three points are shown that are closest to  $\mathbf{x}$ . Using this criterion,  $\mathbf{u}$  would have the shortest distance to  $\mathbf{x}$ .

#### 3.3 Refinement in straits

To refine the above criterion in selected narrow straits, we need to define what a strait is in geometrical terms. The definition used here is based on the fact that



**Fig. 1** A sketch of a simple strait geometry. A strait runs from south to north through the middle of the figure. The islands shaded in light gray belong to  $\Omega^P$ , while the islands shaded in dark grey belong to  $\Omega^S$ . Point  $\mathbf{u}$  is the closest point to  $\mathbf{x}$  on any of these shorelines. Point  $\mathbf{v}$  is the next nearest point to  $\mathbf{x}$ , not in the same shoreline collection. Note that point  $\mathbf{y}$  is located closer to  $\mathbf{x}$  than  $\mathbf{v}$  but is not used for determining the width of a channel since it belongs to the same set of shorelines as  $\mathbf{u}$ . The width of the coastline is defined as the sum of the length of the two arrows. An alternative definition is indicated by the dashed line

most straits have clearly defined sides, i.e., when sailing through a strait one can distinguish between coastlines located to port and to starboard. Define

$$\Omega^P = \bigcup_{i \in P} \Omega_i \tag{9}$$

as a subset of  $\Omega$  of those coastlines  $P$  located on the port side. Likewise, define

$$\Omega^S = \bigcup_{i \in S} \Omega_i \tag{10}$$

as a subset of  $\Omega$  of those coastlines  $S$  located on the starboard side. Naturally, one must have

$$\Omega^P \cap \Omega^S = \emptyset. \tag{11}$$

An example is given in Fig. 1, where both subsets are shaded in light and dark gray. The width of the channel  $w(\mathbf{x})$  at a location  $\mathbf{x}$  is now found by adding the closest distance to  $\Omega^P$  and the closest distance to  $\Omega^S$  or

$$w(\mathbf{x}) = \min_{\mathbf{u} \in \Omega^P} \|\mathbf{x} - \mathbf{u}\| + \min_{\mathbf{v} \in \Omega^S} \|\mathbf{x} - \mathbf{v}\|. \tag{12}$$

If the strait has to be captured by at least  $N$  mesh sizes, then the criterion function becomes

$$f_1(\mathbf{x}) = \frac{w(\mathbf{x})}{N}. \tag{13}$$

The mesh size function  $\delta_1(\mathbf{x})$  is evaluated using Eqs. 4 and 5. Of the three points shown in Fig. 1, the point closest to  $\mathbf{x}$  not belonging to the same subset as  $\mathbf{u}$  is  $\mathbf{v}$ . Note that the third point,  $\mathbf{y}$ , is closer than  $\mathbf{v}$ . The resulting width of the strait is the sum of the two black arrows. In the remainder, this criterion will be referred to as “strait” criterion.

Note that an alternative formulation for  $w(\mathbf{x})$  is indicated by the dashed line in Fig. 1, i.e., the strait line from  $\mathbf{u}$  to  $\mathbf{v}$ . That is,  $w(\mathbf{x})$  is the distance between the two points located on the nearest shorelines that are closest to  $\mathbf{x}$ . More formally,

$$w(\mathbf{x}) = \|\tilde{\mathbf{v}} - \tilde{\mathbf{u}}\|, \tag{14}$$

where

$$\tilde{\mathbf{u}} : \min_{\mathbf{u} \in \Omega^P} \|\mathbf{x} - \mathbf{u}\|, \tag{15}$$

$$\tilde{\mathbf{v}} : \min_{\mathbf{v} \in \Omega^S} \|\mathbf{x} - \mathbf{v}\|. \tag{16}$$

Though the alternative is closer to the width of the strait when  $\mathbf{x}$  is near the narrow sections of the straits, as is the case in Fig. 1, the former is used here since it was found during testing to be more numerically stable and  $\delta_1$  converges more smoothly towards  $\delta_0$  for large distances away from  $\Omega^P$  and  $\Omega^S$ . In the remainder, this will be called the “strait criterion.”

### 3.4 Refinement in local island groups

In addition to clearly defined straits, archipelagos can have local groups of islands that do not neatly define straits but might be important. The above approach can be adapted for these cases. In this case, a distinction between starboard and port is not longer possible. Instead, define

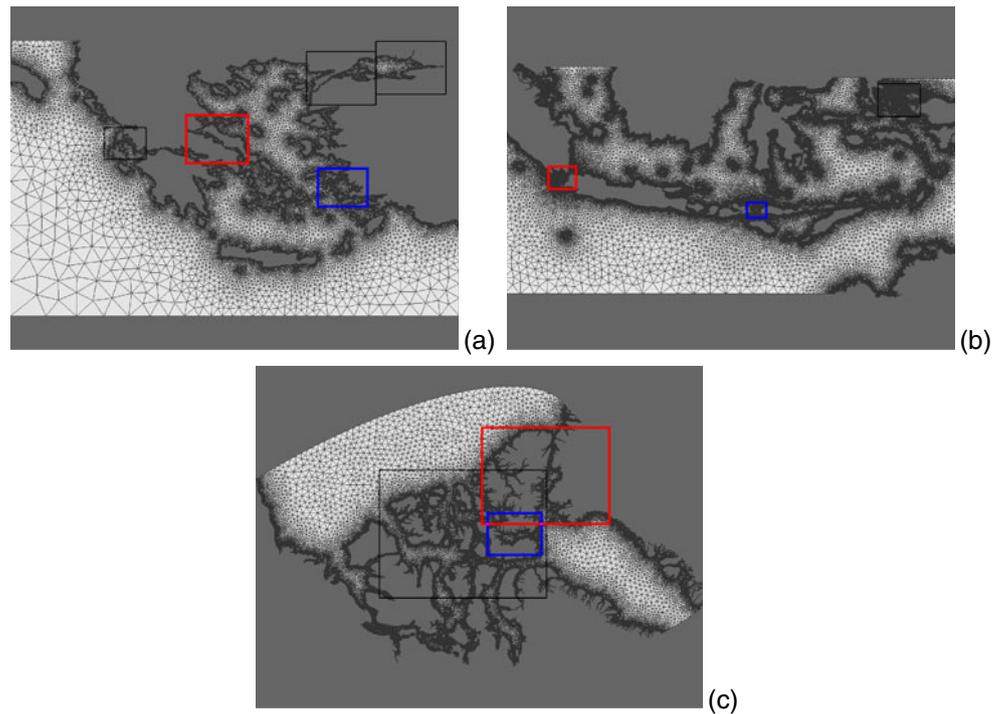
$$\Omega^L = \bigcup_{i \in L} \Omega_i \tag{17}$$

as a subset of  $\Omega$  of those coastlines in  $L$  in this local island group. The definition of the width of the strait is now (see Fig. 1)

$$w(\mathbf{x}) = \min_{\mathbf{u} \in \Omega^L} \|\mathbf{x} - \mathbf{u}\| + \min_{\mathbf{v} \in \Omega^L} \|\mathbf{x} - \mathbf{v}\| \quad \text{subject to } \mathbf{u} \neq \mathbf{v}. \tag{18}$$

The width is based on the proximity to the two nearest, but distinct, coastlines. For the example, in Fig. 1, the distances to points  $\mathbf{v}$  and  $\mathbf{y}$  would be used. The criterion function  $f_1$  is again defined by Eq. 13, and the mesh size field  $\delta_1$  is given by Eqs. 4 and 5. In the remainder, this criterion will be referred to as “local group criterion.”

**Fig. 2** The domains of the three examples: **a** Aegean Sea, **b** the Indonesian Archipelago, and **c** the Canadian Arctic Archipelago. In each figure, the mesh shown is that of the background mesh, without refinement in selected straits. Boxes indicate which regions are to be refined. Red and blue boxes indicate the regions for which the results will be shown in detail



### 3.5 Implementation

For finding the point  $\mathbf{v}$  in a set  $\Omega$  of coastlines with the closest distance to  $\mathbf{x}$ , Gmsh uses the approximated nearest neighbor algorithm (ANN, Aray et al. 1998). When using this algorithm for finding the distances to calculate the width of the strait, using ANN is less straightforward. This is mainly due to the fact that the *two* nearest points on *different* coastlines (local group criterion) or in *different sets of coastlines* (strait criterion) have to be found, which is more complicated. Some care has to be taken to assure this is done efficiently.

First, define a maximum width  $w^{\max}$  as

$$w^{\max} = N\delta_1^{\text{large}} = Nf_1^{\max}. \quad (19)$$

In other words,  $w^{\max}$  is defined as the number of mesh sizes to be captured  $N$  times the upper limit of the mesh size  $\delta_1^{\text{large}}$ .

Before evaluating  $f_1(\mathbf{x})$  using the strait criterion, for each shoreline  $k$  in  $\Omega^P$ , all the shorelines in  $\Omega^S$  with at least one point located within a distance  $w^{\max}$  are located and stored. This to minimize to space to be searched. Note that  $\Omega^P$  and  $\Omega^S$  can be swapped with each other. When using the local group criterion, the same is done, with the exception that the search is limited to  $\Omega^L$  excluding the shoreline where the nearest point is found. In both cases, denote the subset to be searched as  $\Omega^k$ . Depending on the size of  $w^{\max}$ , the size

of  $\Omega$  is much smaller than  $\Omega^P$  or  $\Omega^L$  leading to a far more efficient and faster algorithm.

When evaluating  $f_1(\mathbf{x})$  using the strait criterion, the procedure now is as follows (see also Fig. 1):

1. Find the nearest point  $\mathbf{u}$  to  $\mathbf{x}$ .
2. If  $\mathbf{u} \notin \Omega^P \cup \Omega^S$  or  $d(\mathbf{x}) > w^{\max}$  then  $f_1(\mathbf{x}) = f_1^{\max}$ . This holds that if (1) the nearest point is not part of the strait or (2) farther away than  $w^{\max}$ ,  $f_1(\mathbf{x})$  assumes the maximum value.
3. Find the nearest point  $\mathbf{v} \in \Omega^k$  to  $\mathbf{x}$ .
4. Evaluate  $f^1(\mathbf{x})$ .
5. Evaluate  $\delta_1(\mathbf{x})$  according to Eqs. 4 and 5.

Using the local group criterion, the procedure is nearly the same, apart from the rule that  $\mathbf{u} \in \Omega^L$ .

### 4 Examples

In this section, the new mesh size field described in the previous section is illustrated in three archipelagos: (1) the Aegean Sea,<sup>1</sup> (2) the Indonesian Archipelago, and (3) the Canadian Arctic Archipelago.

<sup>1</sup>The Aegean Sea was traditionally known as Archipelago (*Αρχιπελαγος* in Greek). The general sense of which has since changed to refer to the Aegean Islands and, generally, to any island group because the Aegean Sea is remarkable for its large number of islands.

**Table 1** Parameters used for producing the example meshes

Parameter	Aegean Sea	Indonesian Archipelago	Canadian Arctic Archipelago
$f_0^{\text{small}}$	0.025	0.025	5.0
$f_0^{\text{large}}$	25.000	25.000	250.0
$\delta_0^{\text{min}}$	0.025	0.025	5.0
$\delta_0^{\text{max}}$	5.000	5.000	50.0
$f_1^{\text{small}}$	0.000	0.000	0.0
$f_1^{\text{large}}$	25.000	25.000	250.0
$\delta_1^{\text{min}}$	0.000	0.000	0.0
$\delta_1^{\text{max}}$	5.000	5.000	50.0

Subscript 0 refers to the criterion function based on the proximity to the coastline (see Eq. 8) and subscript 1 refers to the refinement in narrow straits and island groups (see Eq. 13). The parameters refer to those used in Eq. 4 to evaluate the corresponding mesh size functions

Figure 2 shows the spatial extent of each domain. In addition, the background mesh, without refinement, is shown. This mesh is produced based only on the proximity to the coastline (Eq. 8), using the top four (those with subscript 0) in Table 1. Since this is the unrefined mesh, in the remainder of the text,  $N = 0$  for this mesh. Note that depending on the width of the strait, the background mesh might have at some locations only one element across.

The background mesh is further refined by using the mesh size field (Eq. 13) above for several values of the desired resolution  $N$ . The parameters used for meshing the domains are shown in Table 1.

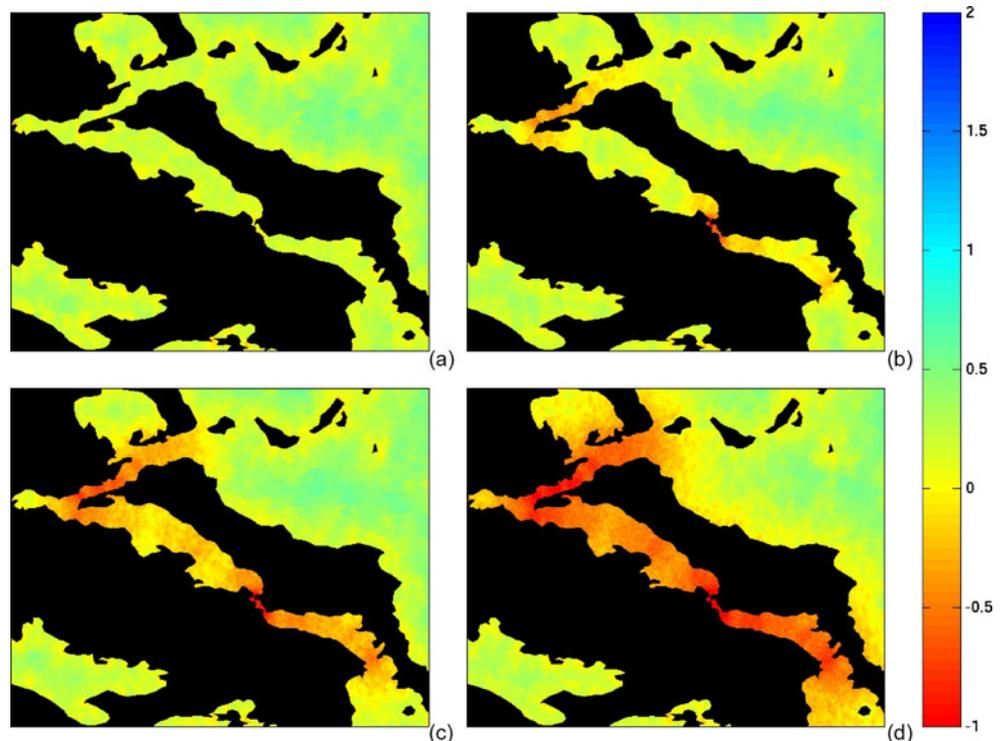
These examples, as well as the areas of refinement, have been chosen to illustrate the mesh size field described above. We do not intend to make statements about whether those resolutions are necessary to resolve the physical processes that might be important in those areas.

### 4.1 Aegean Sea

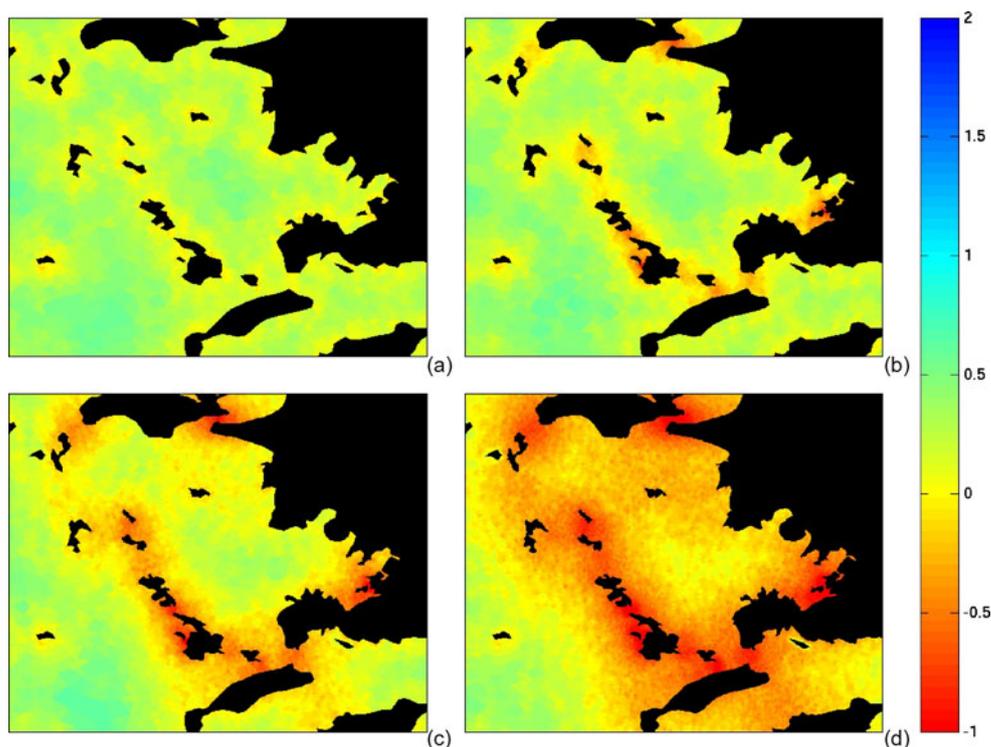
The domain that covers the Aegean Sea and surrounding areas is shown Fig. 2a. It includes the Bosphorus and Sea of Marmara, which connects the Aegean Sea with the Black Sea to the northeast (not shown). Parts of the eastern Mediterranean are also included.

Regions that will be refined are indicated by the boxes in Fig. 2a. The results of the meshing refinement are only shown for the red and blue boxes: (1) the Euboic Gulf between the Greek mainland and the Island of Euboea (located within the red box) and (2) the Icarian Sea (located within the blue box). The results are shown in Figs. 3 and 4, respectively. In these figures, as well as the next few, each triangular element is colored with respect to the  $\log_{10}$  of the radius of the circumcircle of the element

**Fig. 3** The Euboic Gulf and the Euripus Strait. This region is can be found within the red rectangle in Fig. 2a. **a** Background mesh, **b** refinement with  $N = 5$ , **c** refinement with  $N = 10$ , and **d** refinement with  $N = 20$ . The elements are colored with respect to the  $\log_{10}$  of the radius of the circumcircle of the element (in  $\log_{10}$  km)



**Fig. 4** The Icarian Sea. This region can be found within the *blue rectangle* in Fig. 2a. **a** background mesh, **b** refinement with  $N = 5$ , **c** refinement with  $N = 10$ , and **d** refinement with  $N = 20$ . The elements are colored with respect to the  $\log_{10}$  of the radius of the circumcircle of the element (in  $\log_{10}$  km)



the element (in  $\log_{10}$  km). As such, higher resolution shows up as reddish and lower resolution as blue.

Figure 3 shows the refinement of the Euboic Gulf. Here, the strait criterion was used, since one can clearly discriminate between both sides of the gulf: the Greek mainland and the Island of Euboea. Starting from the background mesh (Fig. 3a) that was generated using only the proximity to the coast line, the number of elements enforced across the width of this gulf is doubled from Fig. 3b–d. For the first refinement shown in Fig. 3b, five elements along the width of the gulf and straits were used ( $N = 5$ ). As can be clearly seen from this figure, a higher resolution mesh is generated in the narrow Euripus Strait (slightly off-center in this figure). Doubling the value of  $N$  results in higher resolution mesh while smoothly going to the coarser resolution of the background mesh away from this area.

The refinement for the Icarian Sea is shown in Fig. 4. Here, the local group criterion was used, since no clear strait can be distinguished. Starting from the background mesh, the mesh is refined by doubling  $N$ . In Fig. 4b, where  $N = 5$ , already most of the narrows between the islands are refined, without a significant increase in resolution for the sea. There, the resolution is significantly refined if  $N = 20$ .

#### 4.2 Indonesian Archipelago

The Indonesian Archipelago separates the Pacific Ocean to the north from the Indian Ocean from the

south. The flow is predominantly from the Pacific Ocean to the Indian Ocean and is known as the Indonesian throughflow, which is an important part of the global thermohaline circulation (Godfrey 1996). Three regions are refined (see Fig. 2b), and the two will be discussed here: (1) Sunda Strait (located within the red box) and (2) the area around Komodo Island (located within the blue box).

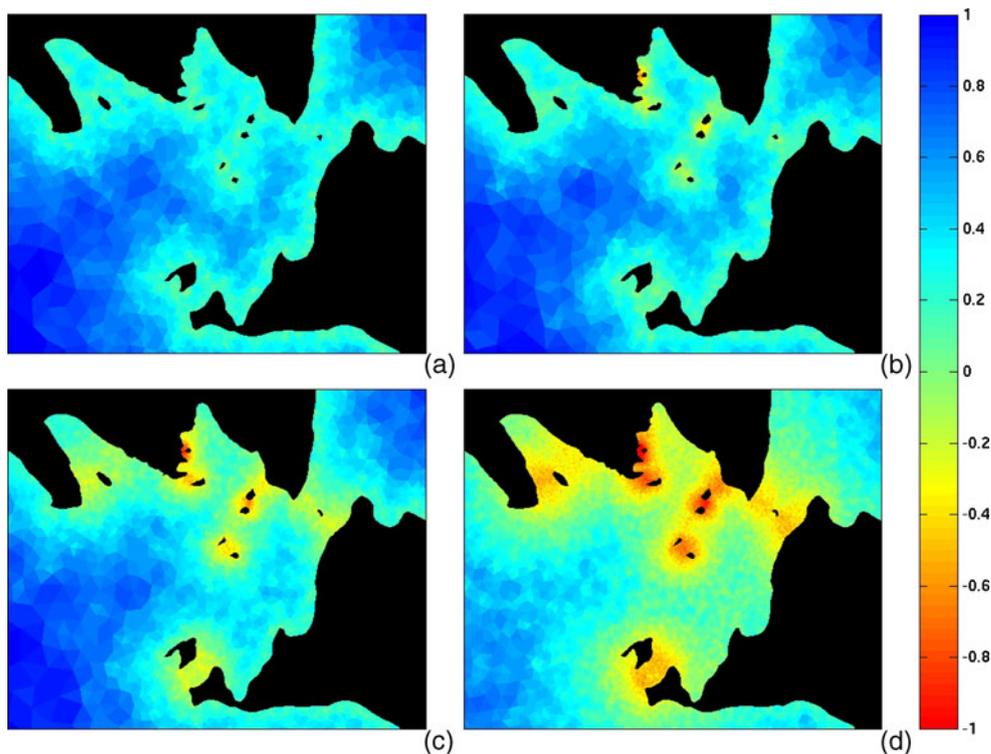
The results of the mesh generation for the Sunda Strait are shown in Fig. 5. The Sunda Strait separates Java (to the east) from Sumatra (to the west). It connects the Java Sea to the Indian Ocean. Within this strait are located a number of volcanic islands including Krakatoa. For the refinement, here we use the local group criterion. Though it is clearly a strait, we found that using the strait criterion resulted in some sharp changes in resolution of the mesh, due to the presence of the islands. The main reason is that it is not possible to smoothly coarsen the resolution to the background mesh near the islands.

The background mesh is shown in Fig. 5a. For  $N = 5$ , some refinement takes place, especially near the islands (see Fig. 5b). Significant refinement happens only when  $N = 10$ , on the Java Sea end of the Sunda Strait, and when  $N = 20$  (see Fig. 5c, d).

Figure 6 shows the generated meshes around Komodo Island (located in the center of these figures). This area is located between the larger islands of Flores (to the west) and Sumbawa (to the east). Be-

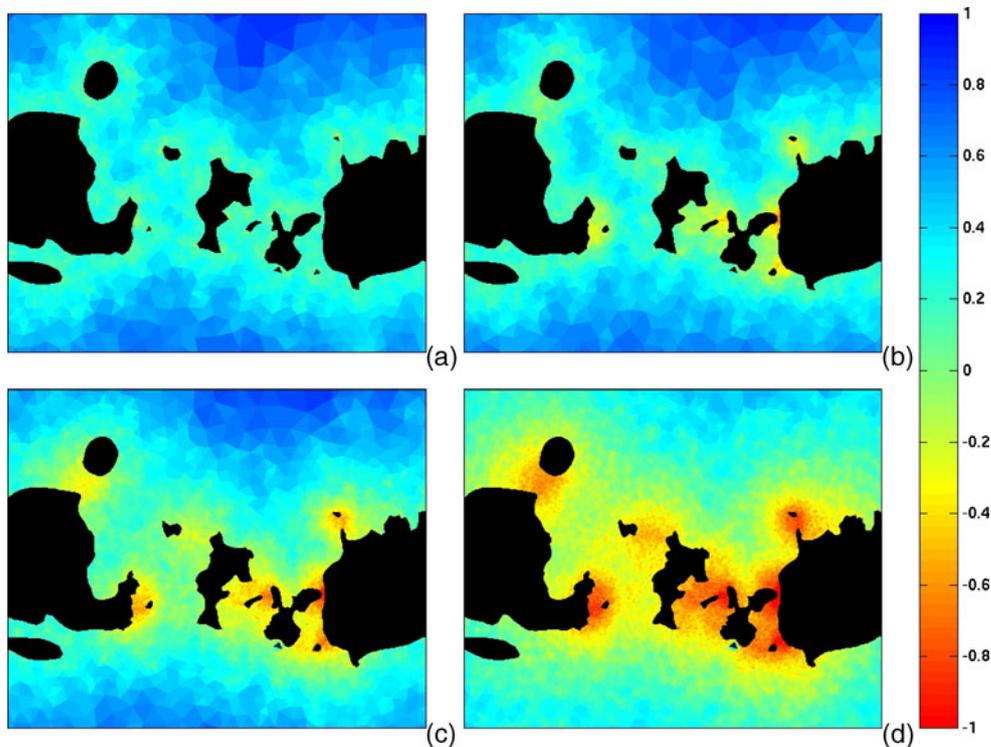
**Fig. 5** The Sunda Strait, between Java to the east and Sumatra to the west, connecting the Java Sea with the Indian Ocean. This region can be found within the *red rectangle* in Fig. 2b.

**a** Background mesh, **b** refinement with  $N = 5$ , **c** refinement with  $N = 10$ , and **d** refinement with  $N = 20$ . The elements are colored with respect to the  $\log_{10}$  of the radius of the circumcircle of the element (in  $\log_{10}$  km)



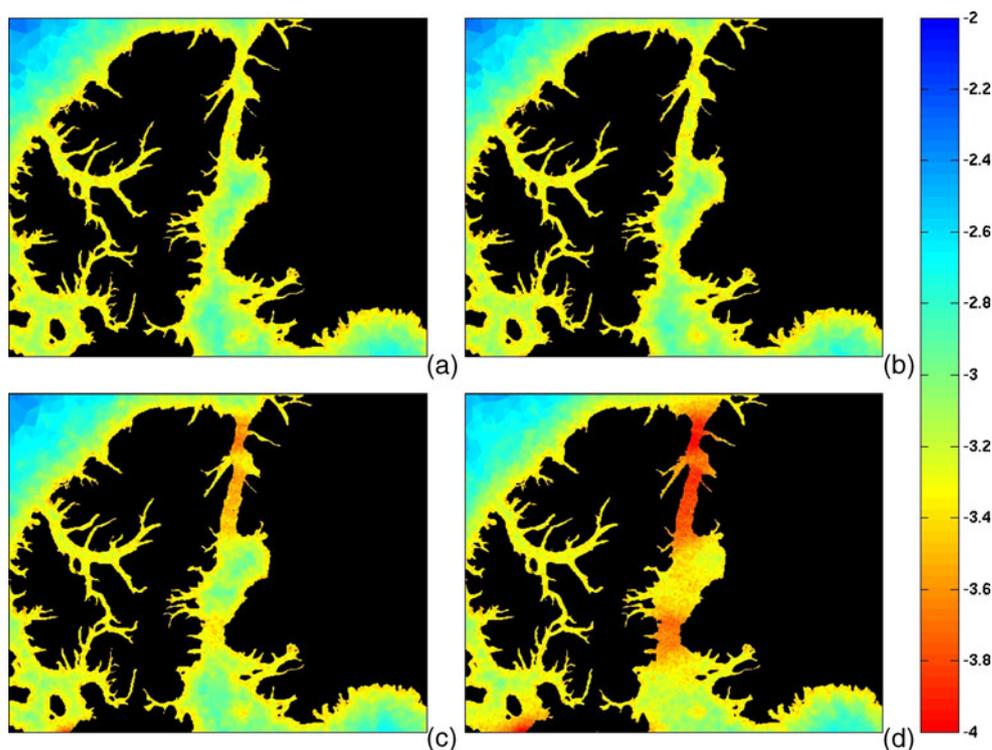
**Fig. 6** The region around Komodo. To the west is Flores and to the east is Sumbawa. Here, the Flores Sea to the north connects with Sumba Strait. This region can be found within the *blue rectangle* in Fig. 2b.

**a** background mesh, **b** refinement with  $N = 5$ , **c** refinement with  $N = 10$ , and **d** refinement with  $N = 20$ . The elements are colored with respect to the  $\log_{10}$  of the radius of the circumcircle of the element (in  $\log_{10}$  km)



**Fig. 7** Nares Strait, connecting the Arctic Ocean to the north with Baffin Bay to the south. This region can be found within the *red rectangle* in Fig. 2d.

**a** Background mesh, **b** refinement with  $N = 5$ , **c** refinement with  $N = 10$ , and **d** refinement with  $N = 20$ . The elements are colored with respect to the  $\log_{10}$  of the radius of the circumcircle of the element (in  $\log_{10}$  km)



tween these two islands are a couple of smaller ones, of which Komodo is located half way between them. Rather than a clear strait, this area resembles more a collection of smaller straits. Instead of defining every strait separately, here we opt to use the local group criterion.

When comparing the background mesh with the first refinement ( $N = 5$ ), it can be seen that the refinement has a direct impact on the smaller straits between the smaller islands and the bigger ones. For example, the areas close to Sumbawa (to the east). Further refinement to  $N = 10$  and  $N = 20$  yields a high-resolution mesh in this area while the coarser resolution is retained in the far field.

#### 4.3 Canadian Arctic Archipelago

The Canadian Archipelago is a complex area of islands and straits in the Arctic, located between the Canadian mainland and Greenland. It is an important pathway of sea ice and water mass transport between the Arctic Ocean and Atlantic Ocean (Kwok 2006; Serreze et al. 2006). Here, the narrow straits have a large impact on ice circulation, due to the formation of ice bridges which can block the flow for several months.

The domain of the Canadian Arctic Archipelago is shown in Fig. 2c. Three areas are refined as indicated by the boxes in this area. The two we will discuss here are (1) Nares Strait (in the red box) and (2) Jones

Sound. For both, the strait criterion was used for the refinement.

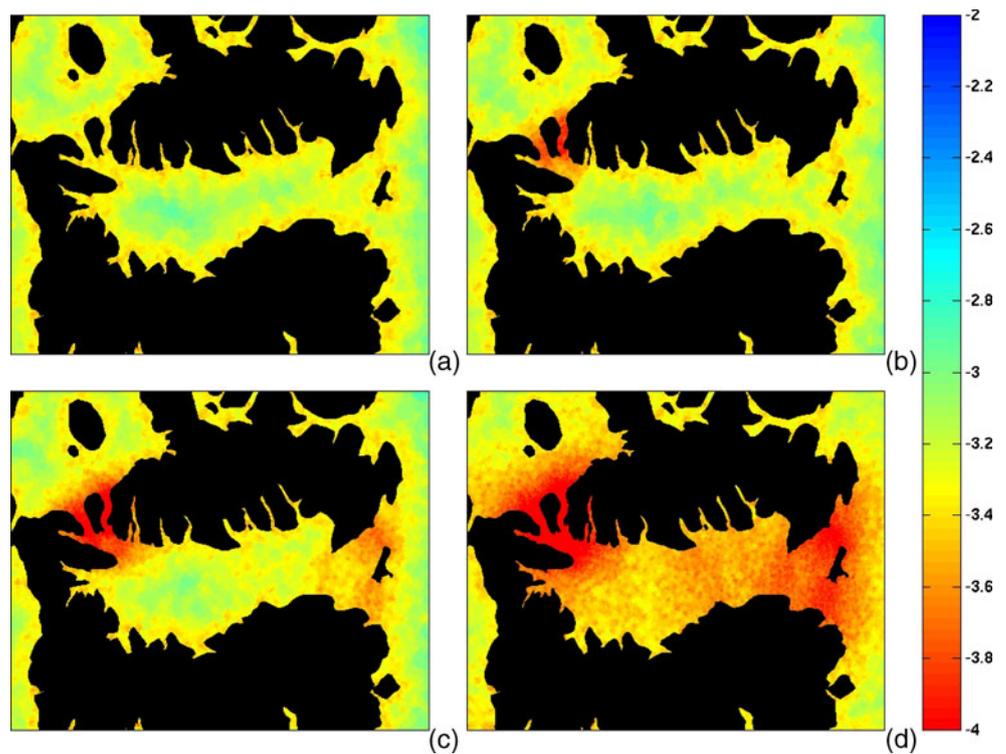
Nares Strait is located between Greenland (to the east) and Victoria Islands (to the west). It is an important gateway of freshwater flux from the Arctic Ocean to the Atlantic Ocean (Melling 2002; Kwok 2005), via Baffin Bay (located to the south) and the Labrador Sea. In this strait, the sea ice shows a complex behavior (Samelson et al. 2006); hence, a high-resolution mesh would be preferable (Lietaer et al. 2008; Terwisscha van Scheltinga et al. 2010). The results of the refinement are shown in Fig. 7. One thing can be easily seen: refinement in Nares Strait does not lead to any refinement in the Arctic Ocean or Baffin Bay. Through increasing the resolution by quadrupling the number of elements across the width of Nares Straits for  $N = 5$  in Fig. 7b to  $N = 20$  in Fig. 7d, you can clearly see in the figures that along Nares Strait the resolution is dependent on the width.

The same can be seen for Jones Sound (Fig. 8). Also here, the refinement is limited to the area, and the refinement decreases away from it. Likewise, the resolution is higher near the narrower ends of Jones Sound.

#### 4.4 Performance

In this section, we will discuss the computational performance of the mesh size field. In Table 2, for each

**Fig. 8** Jones Sound. This region can be found within the *red rectangle* in Fig. 2d. **a** Background mesh, **b** refinement with  $N = 5$ , **c** refinement with  $N = 10$ , and **d** refinement with  $N = 20$ . The elements are colored with respect to the  $\log_{10}$  of the radius of the circumscribed circle of the element (in  $\log_{10}$  km)



of the three domains and the background mesh and refinements, the time taken for the generation and the number of elements of each mesh are shown.

The first observation to be made is that, as expected, increasing the resolution (i.e.,  $N$ ) results in more elements and longer generation times. Not shown in the table is that meshing the 1D geometrical entities only showed a small increase for the three domains, while most of the increase resulted from the 2D meshing.

For the background mesh ( $N = 0$ ), the time it took to produce a mesh varied for each domain, as did the

number of elements. This is because the domains are different in size, and each domain has a different number of geometric entities. These differences between the domains also result in slightly different increased computational overhead when comparing the additional number of elements. For example, for  $N = 4$ , it takes 143 s more to produce the mesh with an additional  $\pm 41,000$  elements for the Canadian Arctic Archipelago. For the Indonesian Archipelago, these numbers are 74 s and  $\pm 74,000$  elements. This is due to the differences in geometry.

Overall, these meshes are produced quite efficiently in a reasonable amount of generation time. Using this refinement algorithm does require more time than just the background mesh. This is easily offset against the amount of time it would have taken defining and tuning the proximity mesh size fields for each island separately.

### 5 Summary

Archipelagos consist of many islands, straits, and lesser bodies of water at a variety of spatial scales. Ideally, one would like to use unstructured meshes, since they have the advantage of accurately representing complex and irregular coastlines and allow for a spatially varying mesh resolution. Generation of meshes with a sufficient

**Table 2** For each domain and value of  $N$ , the total cpu time ( $t_{cpu}$ ) used for producing the mesh, on a early 2008 MacBook with a 2.4-GHz Intel Processor, and the total number of elements

Domain	Number	$t_{cpu}$	Elements
Aegean Sea	0	27.68	58,437
	5	67.85	62,165
	10	88.52	79,345
	20	175.89	154,989
Indonesian Archipelago	0	80.68	202,724
	5	197.61	204,678
	10	212.08	216,765
Canadian Arctic Archipelago	0	60.0805	94,218
	5	107.6585	94,845
	10	135.3688	99,729
	20	203.8518	125,575

$N = 0$  indicates the background mesh without mesh refinement

resolution in archipelagos is not a straightforward task. To produce a mesh, see Lambrechts et al. (2008) for a procedure on how one would need to define a mesh size field that defines the resolution at every spatial point. A popular choice of such a field is based on the proximity to the nearest coastline. For archipelagos, one would have to define and tune for every coastline such a mesh size field in order to get the required resolution in the relevant straits of the archipelago. This is an inefficient and labor-intensive approach.

Here, we have introduced an extension of the mesh size field that is based on the proximity. We use the proximity to the two nearest coastlines to determine the width of a strait or separation between two islands. This is then used to determine the mesh size field by dividing this length by the number of required elements. This new mesh size field is illustrated for three examples of archipelagos: (1) the Aegean Sea, (2) the Indonesian Archipelago, and (3) the Canadian Arctic Archipelago.

**Acknowledgement** This study received funding support from Arcticonet.

## References

- Adcroft A, Marshall D (1998) How slippery are piecewise-constant coastlines in numerical ocean models? *Tellus* 50A:95–108
- Aray S, Mount D, Netanyahu N, Silverman R, Wu A (1998) An optimal algorithm for approximate nearest neighbor searching. *J ACM* 45:891–923
- Danilov S, Kivman G, Schröter J (2005) Evaluation of an eddy-permitting finite element model in the North Atlantic. *Ocean Model* 10:35–49
- George PL, Frey P (2000) *Mesh generation*. Hermes, Lyon
- Geuzaine C, Remacle JF (2009) Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int J Numer Methods Eng* 79:1309–1331
- Godfrey J (1996) The effect of the Indonesian throughflow on ocean circulation and heat exchange with the atmosphere: a review. *J Geophys Res* 101:C12,217
- Gorman G, Piggott M, Pain C, de Oliveira C, Umpleby A, Goddard A (2006) Optimization based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Model* 12:436–452
- Gorman G, Piggott M, Pain C (2007) Shoreline approximation for unstructured mesh generation. *Comput Geosci* 33:666–667
- Grieffes S, Böning F, Chassignet E, Gerdes R, Hasumi H, Hisrt A, Treguier AM, Webb D (2000) Developments in ocean climate modeling. *Ocean Model* 2:123–192
- Hagen S, Westerink J, Kolar R, Horstmann O (2001) Two-dimensional, unstructured mesh generation for tidal models. *Int J Numer Methods Fluids* 35:669–686
- Henry R, Waters R (1993) Geometrically based, automatic generator for irregular triangular networks. *Commun Numer Methods Eng* 9:555–566
- Kwok R (2005) Variability of Nares Strait ice flux. *Geophys Res Lett* 32:L24,502
- Kwok R (2006) Exchange of sea ice between the Arctic Ocean and the Canadian Arctic Archipelago. *Geophys Res Lett* 33:L16,501
- Lambrechts J, Comblen R, Legat V, Geuzaine C, Remacle JF (2008) Multiscale mesh generation on the sphere. *Ocean Dyn* 58:461–473
- Le Provost C, Genco ML, F Lyard, Vincent P, Cancail P (1994) Spectroscopy of the world ocean tides from a finite element hydrodynamic model. *J Geophys Res* 99:777–797
- Lee T, Fukumori I, Menemenlis D, Xing Z, Fu LL (2002) Effects of the Indonesian throughflow on the Pacific and Indian Oceans. *J Phys Ocean* 32:1404–1429
- Legrand S, Legat V, Deleersnijder E (2000) Delaunay mesh generation for an unstructured-grid ocean circulation model. *Ocean Model* 2:17–28
- Legrand S, Deleersnijder E, Hanert E, Legat V, Wolaski E (2006) High-resolution, unstructured meshes for hydrodynamic models of the Great Barrier Reef, Australia. *Estuar Coast Shelf Sci* 68:36–46
- Li X, Shephard M, Beall M (2005) 3d anisotropic mesh adaptation by mesh modification. *Comput Methods Appl Mech Eng* 194:4915–4950
- Lietar O, Fichefet T, Legat V (2008) The effects of resolving the Canadian Arctic Archipelago in a finite element sea ice model. *Ocean Model* 24:114–152
- Lyard F, Lefevre F, Letellier T, Francis O (1994) Modelling the global ocean tides: modern insights from fes2004. *Ocean Dyn* 56:394–415
- Melling H (2002) Sea ice cover in the northern Canadian Arctic Archipelago. *J Geophys Res* 107:3181
- Pain C, Piggott M, Goddard A, Fang F, Gorman G, Marshall D, Eaton M, Power P, de Oliveira C (2005) Three-dimensional unstructured mesh ocean modelling. *Ocean Model* 10:5–33
- Piggott M, Gorman G, Pain C (2007) Multi-scale ocean modelling with adaptive unstructured grids. *CLIVAR Exch Ocean Model Dev Assess* 12:21–23
- Rebay S (1993) Efficient unstructured mesh generation by means of delaunay triangulation and Bowyer–Watson algorithm. *Comput Phys* 106:25–138
- Samelson R, Agnew T, Melling H, Munchow A (2006) Evidence for atmospheric control of sea-ice motion through nares strait. *Geo Rev Let* 33:L02,506
- Schneider N (1998) The Indonesian throughflow and the global climate system. *J Climate* 11:676–689
- Serreze M, Barrett A, Slater A, Woodgate RA, Aagaard K, Lamers R, Steele M, Moritz R, Meredith M, Lee C (2006) The large-scale freshwater cycle of the Arctic. *J Geophys Res* 111:C11,010
- Terwisscha van Scheltinga A, Myers P, Pietrzak J (2010) A finite element sea ice model of the Canadian Arctic Archipelago. *Ocean Dyn* 60:1539–1558
- Wessel P, Smith W (1996) A global, self-consistent, hierarchical, high-resolution shoreline database. *J Geophys Res* 101:8741–8743
- White L, Deleersnijder E, Legat V (2008) A three-dimensional unstructured mesh finite element method shallow-water model, with application to the flows around an island and in a wind-driven, elongated basin. *Ocean Model* 22:26–47