



Fast and robust mesh generation on the sphere—Application to coastal domains[☆]

Jean-François Remacle^{*}, Jonathan Lambrechts

Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Bâtiment Euler, Avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Keywords:

Delaunay triangulation on the sphere
Geophysical flows
Parallel meshing

ABSTRACT

This paper presents a fast and robust mesh generation procedure that is able to generate meshes of the earth system (ocean and continent) in matters of seconds. Our algorithm takes as input a standard shape-file i.e. geospatial vector data format for geographic information system (GIS) software. The input is initially coarsened in order to automatically remove unwanted channels that are under a desired resolution. A valid non-overlapping 1D mesh is then created on the sphere using the Euclidean coordinates system x, y, z . A modified Delaunay kernel is then proposed that enables generation of meshes on the sphere in a straightforward manner without parametrization. One of the main difficulty in dealing with geographical data is the over-sampled nature of coastline representations. We propose here an algorithm that automatically unrefines coastline data. Small features are automatically removed while always keeping a valid (non-overlapping) geometrical representation of the domain. A Delaunay refinement procedure is subsequently applied to the domain. The refinement scheme is also multi-threaded at a fine grain level, allowing to generate about a million points per second on 8 threads. Examples of meshes of the Baltic sea as well as of the global ocean are presented.

© 2018 Published by Elsevier Ltd.

1. Introduction

Traditional ocean models are based on finite differences schemes on Cartesian grids [1]. It is only recently that unstructured meshes have been used in ocean modeling [2–4], essentially using finite elements. 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. [5] uses the mesh generation tools of [6] on several sub-domains to obtain a mesh of the world ocean, aiming at global scale tidal modeling. Further, [7] uses a higher resolution version of the same kind of meshes with the state of the art FES2004 tidal model. [8] gives two algorithms to generate meshes of coastal domains, and uses them to model tides in the Gulf of Mexico. [9] shows high-resolution meshes of the Great Barrier Reef (Australia). At the global scale, [10,11] developed specific algorithms to obtain meshes of the world ocean. More recently, we have developed a proper CAD model of ocean geometries [12]. This model relies on the stereographic projection of the sphere which is conformal i.e. it conserves angles. This

approach has been quite successful up to now: we and other teams have applied it to numerous coastal domains [13,14].

Our CAD approach has two major drawbacks. First, at least two maps are required to cover the whole sphere, making it awkward for atmosphere simulations for example. Then, using splines is maybe not the most robust/natural manner for describing coastlines: geographical information systems provide description of coastlines as series of non-overlapping closed polygons and using splines may lead to intersections.

Here, a new approach that addresses both issues is proposed. It has essentially the following features:

- It starts from the finest available representation of the coastlines that is watertight i.e. a series of polylines on the sphere.
- The fine representation is subsequently coarsened in order to remove all features larger than a desired size h that depends on the scales that are aimed to be modeled.

A modified Delaunay kernel is first presented that allows generation of meshes on the unit sphere. Based on our recent paper [15], a multi-threaded version of this new kernel has been implemented that allows triangulation of over one million points per second on the sphere on a standard quad-core laptop. This new approach does not rely on any parametrization and has all the proof structure of the usual Delaunay kernel (proof of termination, angle-optimality, polynomial complexity).

[☆] This paper has been recommended for acceptance by S. Canann.

^{*} Corresponding author.

E-mail address: jean-francois.remacle@uclouvain.be (J.-F. Remacle).

In this new approach, the most refined representation of coastlines available in the geographical system is used as input. A constrained Delaunay mesh of the whole data set is created using the new Delaunay procedure. This first mesh allows to automatically and robustly removing from the domain any water channel that has a width that is smaller than a given threshold (this threshold being possibly variable in space). This step leads to a coarsened version of the boundary mesh where locally small features have been removed, producing a valid (non-overlapping) boundary description of the domain. Finally, a multi-threaded version of the edge-based Delaunay refinement procedure of [16] has been used to saturate the domain with points and triangles.

The developments that are presented here have been released as a self consistent open source code that can be used as a standalone program or that can be plugged in other software's such as Gmsh [17] or QGIS [18].

2. Delaunay triangulation on the sphere

Here we consider the unit 3D sphere S centered at the origin $\mathbf{o}(0, 0, 0)$: $S = \{\mathbf{x}(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$. Any section of a sphere by a plane is a circle. We distinguish *great circles* that are sections of a sphere that diameter is equal to the diameter of the sphere and *small circles* that are any other section.

Consider two points \mathbf{p}_1 and \mathbf{p}_2 on the sphere. Geodesics are the shortest path between points on the sphere. It is well known that geodesics of the sphere are segments of a great circle. The *geodesic distance* between \mathbf{p}_1 and \mathbf{p}_2 is the length of the great circular arc joining \mathbf{p}_1 and \mathbf{p}_2 . We call it $d(\mathbf{p}_1, \mathbf{p}_2)$.

A spherical triangle $\mathcal{T}_1(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ (see Fig. 1) is a figure formed on the surface of a sphere by three great circular arcs intersecting pairwise in three vertices \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 . A spherical triangle is sometimes called an Euler triangle. Spherical triangles have an orientation that is computed as the sign of the volume $\|\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{o}\|$ of tetrahedron $t(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{o})$ is positive, with \mathbf{o} the center of S .

The circumcircle $\mathcal{C}_{\mathcal{T}_1}$ of the spherical triangle \mathcal{T}_1 is the small circle that is formed by the section of S by the plane defined by points \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 (see Fig. 1). The circumcircle $\mathcal{C}_{\mathcal{T}_1}$ divides the sphere into two parts. Consider a point \mathbf{p} of S :

- \mathbf{p} is inside $\mathcal{C}_{\mathcal{T}_1}$ if $\|\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}\| < 0$.
- \mathbf{p} is outside $\mathcal{C}_{\mathcal{T}_1}$ if $\|\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}\| > 0$.
- \mathbf{p} is on $\mathcal{C}_{\mathcal{T}_1}$ if $\|\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}\| = 0$.

There are exactly two antipodal points that are equidistant to \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 . We define the spherical circumcenter of \mathcal{T}_1 as the point $\mathbf{c}_{\mathcal{T}_1}$ that is equidistant to \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 :

$$d(\mathbf{p}_1, \mathbf{c}_{\mathcal{T}_1}) = d(\mathbf{p}_2, \mathbf{c}_{\mathcal{T}_1}) = d(\mathbf{p}_3, \mathbf{c}_{\mathcal{T}_1})$$

and that is inside $\mathcal{C}_{\mathcal{T}_1}$. This corresponds to one of the two antipodal points that is the closest to \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 .

Consider a point set $P = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ of n points of S . A triangulation $T(P)$ of P is a set of $2n - 4$ non overlapping spherical triangles

$$T(P) = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_{2n-4}\}$$

that exactly covers S with all points of P being among the vertices of the triangulation.

A spherical triangle \mathcal{T}_j is Delaunay if its circumcircle is empty i.e. if no point \mathbf{p}_i of P lies inside \mathcal{T}_j . The Delaunay triangulation $DT(P)$ is such that every triangle \mathcal{T}_j of $DT(P)$ is Delaunay. This construction is an actual Delaunay triangulation [19,20]. An interesting

interpretation of this kernel starts with the 3D orientation predicate that consists in computing the sign of the volume of tetrahedron formed by points $\mathbf{p}_j(x_j, y_j, z_j)$, $j = 1, \dots, 4$:

$$\text{sign} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} \quad (1)$$

The 2D 'in-circle' predicate that tells if point \mathbf{p}_4 belongs to the circum-circle of triangle formed by points \mathbf{p}_1 , \mathbf{p}_2 , and \mathbf{p}_3 can be written as

$$\text{sign} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1^2 + y_1^2 & x_2^2 + y_2^2 & x_3^2 + y_3^2 & x_4^2 + y_4^2 \end{vmatrix} \quad (2)$$

Predicate (2) has a form that is close to the one of (1). This is an expression of the standard link between 3D convex hulls and 2D Delaunay triangulations: assume a 2D triangulation and lift it to the paraboloid $z = x^2 + y^2$. Then a 2D triangle is Delaunay if it belongs to the convex hull of the lifted triangulation. In other words, a point $\mathbf{p}(x, y)$ belongs to the circumcircle of a triangle $t(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ if its lifting $\mathbf{p}'(x, y, x^2 + y^2)$ on the paraboloid is below the plane defined by the lifted triangle $t'(\mathbf{p}'_1, \mathbf{p}'_2, \mathbf{p}'_3)$. This is verified by computing the sign of the volume of tetrahedron with vertices $\mathbf{p}'_1, \mathbf{p}'_2, \mathbf{p}'_3, \mathbf{p}'_4$ using Eq. (1). In the case of a triangulation on a unit sphere, predicate (1) becomes

$$\text{sign} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} = \text{sign} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ \sqrt{1 - x_1^2 - y_1^2} & \sqrt{1 - x_2^2 - y_2^2} & \sqrt{1 - x_3^2 - y_3^2} & \sqrt{1 - x_4^2 - y_4^2} \end{vmatrix} \quad (3)$$

The lifting here is on the sphere and not on the paraboloid and the construction that is proposed is a Delaunay triangulation.

3. A parallel Delaunay kernel

A triangulation $T(P)$ of P is a set of non overlapping triangles that exactly covers the convex hull $\Omega(P)$ with all points of P being among the vertices of the triangulation.

Delaunay triangulations are popular in the meshing community because fast algorithms exist that allows generation of $DT(P)$ in $\mathcal{O}(n \log(n))$ complexity.

Let DT_k be the Delaunay triangulation of a point set $P_k = \{\mathbf{p}_1, \dots, \mathbf{p}_k\} \subset \mathbb{R}^d$. The *Delaunay kernel* is a procedure that allows the incremental insertion of a given point $\mathbf{p}_{k+1} \in \Omega(P_k)$ into DT_k and to build the Delaunay triangulation DT_{k+1} of $P_{k+1} = \{\mathbf{p}_1, \dots, \mathbf{p}_k, \mathbf{p}_{k+1}\}$. The *Delaunay kernel* can be written in the following abstract manner:

$$DT_{k+1} = DT_k - \mathcal{C}(DT_k, \mathbf{p}_{k+1}) + \mathcal{B}(DT_k, \mathbf{p}_{k+1}), \quad (4)$$

where the Delaunay cavity $\mathcal{C}(DT_k, \mathbf{p}_{k+1})$ is the set of all triangles whose circumcircles contain the new point \mathbf{p}_{k+1} (see Fig. 2; the triangles of the cavity cannot belong to DT_{k+1}) and the Delaunay ball $\mathcal{B}(DT_k, \mathbf{p}_{k+1})$ is a set of triangles that fill the polyhedral hole that has been left empty while removing the Delaunay cavity $\mathcal{C}(DT_k, \mathbf{p}_{k+1})$ from DT_k .

Download English Version:

<https://daneshyari.com/en/article/6876375>

Download Persian Version:

<https://daneshyari.com/article/6876375>

[Daneshyari.com](https://daneshyari.com)