qmesh synoptic manual

Release 0.5

Alexandros Avdis

Aug 30, 2020

Contents

Welcome to the qmesh synoptic manual.

qmesh is an unstructured triangular mesh generator for geophysical domains. It is suited to mesh generation over topologically two-dimensional domains, typically used in coastal and ocean modelling. Typical examples can be seen in the *Academic papers* section.

qmesh was developed by Alexandros Avdis and Jon Hill, primarily to meet our meshing needs. We developed qmesh with an open-source philosophy in mind, so you are welcome to view the qmesh code and use qmesh, under the GNU General Public License, version 3. As the project grew and matured, the code was reorganised into various modules and packages. Following the traditional Python practice, each package is self-contained and includes documentation. However, the documentation of the packages is best described as a technical reference. Therefore, we have developed this synoptic manual where we aim to provide a brief but complete description of the general architecture of qmesh, development practices, installation procedures, an overview of further documentation and literature as well as a few tutorials that will enable the novice to make a quick start.

In particular, this manual is organised as follows:

- The *Introduction* is a broad descriptions of the aims and objectives behind qmesh development.
- The *Architecture* is discussed in the second chapter, describing the user interfaces, the packages and the qmesh modules.
- The third chapter outlines the development and distribution methods, including a description of the testing framework and release practices. The chapter includes verbatim copies of the licences attached to qmesh code and documentation.
- Chapter four includes instructions on qmesh *Installation*.
- Chapter five outlines sources of *Further documentation* and *Tutorials*. Other literature sources, such as *Academic papers*, are also listed.

CHAPTER 1

Introduction

Computational models are increasingly used to study complex geophysical fluid dynamics and its interaction with biological and geochemical processes [24]. The potential insight from simulations has made such models a valuable tool in scientific research as well as engineering. Simulations are used to assess the impact of anthropogenic changes, the vulnerability of infrastructure to natural hazards [25][18] and in hydrocarbon exploration and sequestration research [16][17][22]. qmesh was built to facilitate one of the first stages in computational geophysical modelling: the specification of the simulation domain and its tessellation into discrete elements, commonly referred to as mesh generation. The predictive accuracy of simulations can be significantly affected by the mesh resolution, gradation and shape of mesh elements, collectively identified as "mesh quality". Therefore, generation of high-quality meshes is fundamental to ocean and coastal modelling.

The paradigm of mesh generation in large-scale geophysical modelling can be broadly described as a two–step procedure. In the first step, the domain is defined in a topologically two-dimensional space, bounded by geophysical contours such as shorelines and open boundaries [1]. A finite, two-dimensional reference surface is thus defined, often on a geodetic datum, and a mesh is generated over this reference surface. If a two-dimensional approximation, such as the depth-averaged shallow water equations is sufficient, mesh-generation is complete. When a three-dimensional approximation is required, the second step is the projection of the surface mesh vertices at successive levels towards the domain floor, thus creating three-dimensional elements [1]. The second step requires little user intervention, so the first step has become a synonym for mesh generation, in the large–scale geophysical modelling context.

Despite the reduction of the relevant dimensions to just two, the production of quality meshes for geophysical domains can be an elaborate procedure [1][10][7][26][8]. A significant number of data sources must be combined to compose a geometrically complex domain, and the geometry of geophysical domains is one of the most widely known examples of fractal geometry in nature [27]. Geometrical length–scales across four orders of magnitude is typical in ocean modelling: For example in ocean modelling, the simulation domain size can span hundreds of kilometres, while the smallest bays can be tens of metres long. An even smaller scale may be relevant when the domain must accurately represent coastal infrastructure such as piers, pylons or embankments [28][20]. In addition to domain geometry, the flow typically exhibits a large range of scales, with many transient flow features, such as internal waves or jets, appearing due to the geometric complexity of the domain [29]. Therefore, a mesh must represent very complex domains with element sizes across a broad range of scales, with smaller elements in areas that require a higher fidelity, while gradation across element sizes must be smooth. Unstructured meshes are increasingly favoured in many models as they tend to satisfy the above requirements with relative ease [30].

High–quality generic unstructured mesh generation packages are available, enabling routine generation of meshes. However, the interface of most mesh generators is based on *Computer Aided Design and Manufacture (CAD–CAM)*.

Such interfaces have been developed for describing geometries produced by manufacturing and construction processes and do not facilitate use or manipulation of geographical data [31]. In particular, the fractal nature of ocean and coastal domains, as well as the various conventions used in geodetic coordinate reference systems are not natively expressed in CAD–CAM systems [31][20]. Combining shoreline or bathymetry data is a typical example; one may have to combine several global, national and regional data sets as well as data from very high–resolution surveys over relatively small regions. Each data–set can also differ at least in terms of extents, resolution, coordinate reference system and vertical zero–datums. Unlike CAD–CAM, *Geographical Information Systems (GIS)* have been developed specifically for storing and analysing geographical information. GIS packages are robust and widely used in research as well as in operational and strategic planning contexts, where resource management, hazard mitigation and infrastructure development are a few examples. Therefore, GIS packages are ideal for supplying mesh generators with the geospatial data they require.

Another important feature of GIS systems is their capability to interact with databases, allowing concurrent data analysis and manipulation. Databases can also be used to record data origin and evolution, termed *data provenance* [5]. However, databases in GIS systems are typically not maintained in a manner pertinent to scientific research, where the primary aim of data provenance is to show reproducibility [5]. The practice of *Research Data Management (RDM)* is aimed at addressing data provenance, attribution and reproducibility. A record of data and software depended upon is usually required as evidence of reproducibility, including scientific data and software used in preparation of simulations. Therefore, the reproducibility of numerical simulations relies, at least in part, on the ability to exactly reproduce the underlying mesh. The integration of RDM and mesh generation was motivated by the increased attention on the reproducibility of scientific computation [32], perhaps as much as open–source software. Also, public research bodies are adopting policies on data and software output from publicly–funded research to be made readily available, and provenance to be clearly identified [33][34][35]. In the industrial sector reproducibility, data archiving and data provenance are viewed as efficient modelling practices. Industry and governing bodies are also bound by regulatory frameworks which require public accesibility to data during the planning phase [36][37][38][39], as well as after commisioning [40][41], especially when data pertains to environmental impact of infrastructure.

Here we present the qmesh package, interfacing GIS with a mesh–generator and online data repositories. We link the abstractions offered by mesh generators and GIS packages and build tools that facilitate mesh–generation for coastal flow modelling. Unlike existing integrations of GIS and mesh generators, qmesh was principally developed as an object–oriented software library, accessible through an Application Programming Interface (API). In qmesh GIS capability is implemented through the use of existing and robust GIS implementations as generic libraries, rather than building extensions to a particular GIS implementation, such as is described in [42][10][7][26][8][15]. *qmesh is thus a library with which command line utilities and a graphical interface have also been developed.* The broader aim of qmesh development is the creation of robust, efficient, operational and user–friendly tools for mesh generation over geophysical domains. The qmesh design was centred around providing the following requirements:

- Facilitation of domain geometry and mesh element size definitions;
- An intuitive way of specifying boundary conditions and parameterizations;
- The ability to use various geodetic coordinate reference systems;
- Promotion of the reproducibility of output and citation of data provenance
- Provision of an open–source and tested package.

CHAPTER 2

Architecture

The user perspective of mesh–generation packages is centred around specification of two parts: domain geometry and mesh element size. Encoding domain geometry and mesh element size is a useful paradigm for describing meshes for large–scale geophysical modelling [1], as it organises the necessary information in a conceptually clear way. We here follow the conventional norm in ocean modelling, discussed in the *Introduction* , where meshes are produced in a topologically two–dimensional space [1] and the domain is bound by various contours, typically topographic contours, and arbitrary lines. The two primary data structures of GIS are used to describe linear features and field data. A *vector* data structure can represent points, lines and regions on a reference surface, while a *raster* data structure encapsulates the discrete representation of fields. The analogue to the abstraction used to drive mesh generators is clear: the domain geometry can be described with a vector data structure, while a raster can express the element size metric. Thus, the obvious route to interfacing mesh generators with GIS is to provide a translation of GIS data structures into the corresponding structures native to the mesh generator software. The data structure translation is at the heart of the qmesh package. The translation is done with little user intervention, as the user typically interacts with the GIS package and parts of the qmesh package that facilitate specification of domain geometry and mesh element size. To meet the demands around data archiving, publication and reproducibility a *Research Data Management (RDM)* tool is included in the qmesh package. The RDM tool facilitates the process of publishing all resources, including output such as the mesh, to online, persistent and citable repositories.

The GIS package chosen is QGIS [2], the mesh generator is Gmsh [3] and the PyRDM software library [4] was used to integrate research data management [5]. The main reasons for choosing QGIS, Gmsh and PyRDM, are robustness, extensibility and permissive licences. Specifically, Gmsh is a robust mesh generator featuring a CAD–CAM interface, and has been used for generating meshes in various scientific and engineering domains, including geophysical domains [6][3]. QGIS is a widely used GIS platform, with an active community of users and developers, and has been used as a user interface to mesh generation in past efforts [7][8]. The functionality of QGIS is available to the user as a standard GIS system with a rich graphical interface and as an object–oriented Python module. Therefore, QGIS is a solid framework on which to develop complex applications that require GIS methods. Also, QGIS provides a framework for using such applications as extensions, via the QGIS graphical user interface. Finally, QGIS, Gmsh and PyRDM are released under the GNU General Public Licence, making possible the use of qmesh in an academic or industrial context, free of charge.

Fig. 2.1 presents an overview of the architecture of qmesh and conveys the usual work–flow. As shown, qmesh is composed of four modules, named *vector*, *raster*, *mesh* and *publish*. The purpose of the modules vector and raster is to facilitate the definition of the domain geometry and mesh–size metric and to interface qmesh to QGIS. The translation between GIS and mesh–generator data structures is performed by the mesh module, thus interfacing qmesh to Gmsh.

The RDM functionality is implemented by the publish module, interfacing qmesh to online repositories and enabling identification and publication of data. Apart from just conceptual, the modules shown in Fig. 2.1 are also the *Python modules* of the qmesh implementation and are discussed in detail in section *Module design and implementation*. To allow access to the qmesh modules from a variety of environments three different *User Interfaces* have been developed, discussed in *The qmesh interfaces and packages*. The interfaces are implemented as separate *Python packages*, to facilitate distribution, installation and namespace organisation.

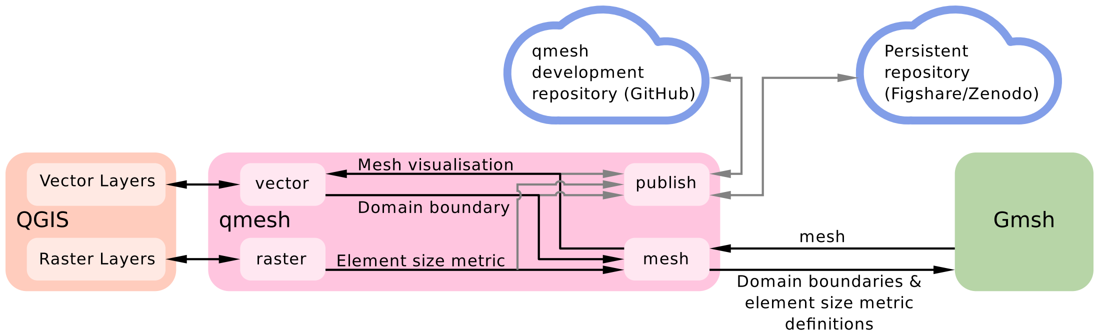

Fig. 2.1: Schematic of qmesh library architecture. The arrows indicate data–flow when qmesh is used.

2.1 The qmesh interfaces and packages

qmesh can be used in a graphical as well as a programmatic environment, through three different user interfaces:

1. A Python–based *Application Programming Interface (API)*.
2. A Linux Terminal *Command Line Interface (CLI)*.
3. A *Graphical User Interface (GUI)*.

Each interface is implemented as a separate Python package. We have avoided the creation of a single monolithic package to minimise dependencies, to orthogonalize the module definition to the User-Interface definition (to the extent possible), but also to more narrowly define the purpose of each package and thus help in more targeted testing. Thus, the following packages compose what collectively can be identified as qmesh:

1. `qmesh`. This is the "core" qmesh package, other APIs/packages require the `qmesh` package as a dependency. According to the Python documentation, Python packages are a way of structuring the Python module namespace by using dotted module names. Thus, the package presents a Python API and contains the implementation of the modules shown in Fig. 2.1, accessible as `qmesh.vector`, `qmesh.raster`, `qmesh.mesh` and `qmesh.publish`.
2. `qmesh-cli`. This package contains the code needed to access the qmesh functionality from a linux terminal.
3. `qmesh-qgis-plugins`. This package contains the code necessary to access the qmesh functionality from within QGIS, using a graphical interface.

2.1.1 Application Programming Interface

The Python-based API is an integral part of the qmesh implementation. It can be used to build scripts that automate series of operations, but its primary purpose is to allow the use of qmesh as a software library. It is the most powerfull and flexible of the qmesh interfaces, due to the power and flexibility of Python, its interactive shells (e.g. [9]) and numerous extensions.

2.1.2 Command Line Interface

The command line interface consists of a set of utilities each with well–defined input and output, making each utility a separate program. However, their purpose is to be used as diagnostic tools or, to automate operations which do not require a graphical interface.

2.1.3 Graphical User Interface

The graphical user interface is a QGIS extension. This way qmesh can be used via the QGIS application and in combination with other QGIS functionality. The GUI has been designed to allow access to the qmesh package with ease and little knowledge of the qmesh design.

2.2 Module design and implementation

The qmesh implementation follows the object-oriented programming paradigm. Thus, each of the modules shown in Fig. 2.1 contain class definitions that facilitate the aim of the module. In this section, we describe the modules in more detail and summarise the classes typically used. A complete listing of all definitions in each module can be found in the *API Rerefence*.

2.2.1 The vector module

The vector module is used to construct a complete definition of the domain geometry in terms of domain boundaries (lines) and domain surfaces (areas). Surfaces are defined in terms of lines, so the definition of surfaces is automated by methods in the vector module such that only the boundary lines need to be supplied. Other methods allow for essential geometric operations such as checking for erroneous geometries (i.e. intersecting shorelines) or the removal of small islands and lakes, based on a threshold surface area specified by the user. The necessity of shoreline processing in ocean modelling is discussed in [10], where the Terreno project used GMT [11] to affect shoreline processing. In qmesh however, geometry processing is done primarily through the QGIS software library, also allowing use of extensive functionality built-into the GIS platform. In addition to geometry definition, methods for identifying separate parts of the domain geometry are necessary. For example, open boundaries are associated with different boundary conditions to shorelines. The qmesh user can assign numerical identifiers to separate lines and apply different boundary conditions to separate boundaries. Numerical IDs can also be assigned to surfaces, allowing the identification of areas where different numerical treatments or parameterizations must be applied, as shown in the *Tutorials*. The QGIS library is used to store and retrieve the digital IDs as standardised feature attributes. The output of the vector module uses the *ESRI shapefile* [12] vector data–structure, which also supports storage of the ID feature attributes. This way the module output as well as IDs can be visualised, assigned and edited with any GIS platform.

2.2.2 The raster module

The aim of the raster module is to facilitate construction of raster fields that describe the desired element edge length distribution over the domain. For example, the element size might be chosen to be smaller in areas of shallow water, steep bathymetry and areas of significant variation in bathymetry slope. Therefore an optimal element size distribution is typically expressed as a function of bathymetry, its gradient and Hessian and the distance to boundaries [13][14][6]. The raster module facilitates application of various mathematical operators to be applied to raster data such as derivatives, methods for combining raster fields such as pointwise minimum and maximum operators, but also methods for calculating the *distance function* raster from any given vector feature. The latter is useful when specifying a mesh size gradation towards specific features in the domain: for example, the element size gradually becoming smaller as a coastline or a tidal turbine is approached. A generic method has been implemented, aimed towards the construction of element size raster fields based on the distance from a given vector feature (lines, polygons or points). This kind of operation is expressed by the arrows between the raster and vector modules inside qmesh, in Fig. 2.1. As with the vector module, the output of the raster module uses GIS raster data structures enabling visualisation and editing of the output via the GIS system.

2.2.3 The mesh module

The mesh module is used to translate the domain and mesh element size definitions into Gmsh data structures and, as suggested in figure Fig. 2.1 can be used to convert the mesh into a vector data–structure. Such functionality enables mesh visualisation using QGIS, and in particular to over–lay the mesh on other data. Various qualities of the mesh can thus be assessed and the work–flow can be restarted towards improving the mesh. The meshing module also allows the user to specify the coordinate reference system of the output mesh, which need not be the same as that of the domain geometry and mesh–metric raster. Coordinates are reprojected to the target coordinate reference system before the data is passed to the mesh generator. The reprojection procedure uses the QGIS library; this way meshes can be obtained in all cartographic projections that QGIS supports and identifies via an *EPSG code*. The output mesh is two–dimensional and the EPSG specification describes the dimensions, including their units. As a particular case, the output mesh can be constructed in a three–dimensional space, where the mesh vertices lie on a sphere, using specific Gmsh functionality described in [6][15]. The vertex coordinates are specified in terms of a Cartesian reference system whose origin lies at the sphere centre, the z–axis is the axis of rotation and the x–axis intersects the surface of the sphere at 0^o longitude and 0^o latitude. Meshes thus constructed can be used to perform global simulations or simulations over large areas [1][6][16][17][18][15].

2.2.4 The publish module

The aim of the publish module is to facilitate provenance description and reproducibility of qmesh output. Broadly, the specific version of qmesh used to produce the mesh and all of the input data sources are stored in an online repository. In general data provenance may seem intractable since data and software are often stored in a non–persistent way and are not easily accessible. However, given the increasing importance of data provenance, online data repositories with efficient storage and access controls such as Zenodo and figshare, are becoming popular means of archiving and dissemination. Also, such services incorporate meta–data as means of describing hosted data and *minting* a unique *Digital Object Identifier (DOI)*. The DOI is a standardised [19] citable identifier and is aimed to be assigned to digital objects, stored in a persistent way in open repositories. Therefore, DOI is a widely–adopted identifier for digitally stored data, be that a scientific publication, the output of scientific computations or records from experiments and observations. Given the wide range of data sources that can be combined during mesh generation for realistic geophysical domains, the task of manually maintaining the provenance information of all the relevant data files can be time–consuming and error–prone. As shown in figure Fig. 2.1 the publish module interfaces with the qmesh development repository, via PyRDM [4][5], to identify the exact version of qmesh used. A query is then made with the repository hosting service to establish if this version of qmesh has already been uploaded and assigned a DOI. A similar query is performed for each input data source. Each unpublished item is then uploaded and a new DOI is minted and assigned to the entire dataset. The dataset also includes citations, in the form of meta–data, of the DOI markers of already published items. The various DOI markers can be thought of as nodes of a tree, and the citations are the tree connections, a similar concept to scientific publications. Also, the output can be archived in a private repository, without a DOI, to facilitate archival of commercially sensitive information.

CHAPTER 3

Development, Distribution and Licence

3.1 Development and distribution repositories

3.1.1 Code development repositories

3.1.2 DOI repositories

3.1.3 Docker Cloud repositories

3.1.4 qmesh in the Python Package Index

The qmesh packages can be obtained through the Python Package Index (PyPI), also known as "The Cheeseshop". The python *pip* package manager can be used to install the packages with all dependencies, details are given in the Installation section.

3.2 Testing

3.2.1 Unit testing

3.2.2 Regression testing

3.2.3 Integration testing

3.3 Version numbering

(Section in progress) The PEP 440 is followed to implement Semantic Versioning 2.0.0.

3.4 Licences

The qmesh code and documentation are open-source and are released under licences published by the Free Software Foundation. Verbatim copies of the licences are placed in the following sections. In specific:

- The source code of all qmesh packages is released under the GNU General Public Licence, version 3 or (at your option) any later version. A copy of the full licence text is available in the section titled "GNU GENERAL PUBLIC LICENSE".
- All qmesh documentation, including this manual, is released under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation A copy of the full licence text is available in the section entitled "GNU Free Documentation License".

3.4.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d. A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR

IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

3.4.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was

published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

''' Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

'''

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

''' with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

'''

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

3.5 How to cite qmesh

It is good practice to cite sources, as it promotes provenance, attribution and reproducibility. The reason the qmesh code, as well as manuals, are available under Digital Object Identifiers (DOI) is to facilitate citation, but also to provide persistent sources where the code, containers and documentation can be accessed. A number of papers have been published in academic journals using or describing qmesh [18][5][20][21][22][23].

Given the independent peer-review process typical of academic journals, a paper is a source you can confidently cite as a robustly reviewed description of qmesh. Indeed, the availability of code and data through persistent repositories with a minted DOI is motivated by greater transparency, reproducibility and attribution of scientific research. Also, qmesh is an open source project. The source code and documentation are available free of charge, so appropriate attribution is the only way to support the project's longevity.

Therefore, it is recommended to:

- Include the qmesh version number in the citation.

- Cite the qmesh paper [23].
- Cite this manual.
- Cite the code or container, using the DOI.
- State the gmsh version, with appropriate references [3].
- State the qgis version, with appropriate references [2].

For example, a possible way to cite all of the above, when the Python API package was used is:

In this study the meshes were generated using the qmesh Python package (version 0.9) [23], which interfaces gmsh (version 2.8) [3] and qgis (version) [2]

If you want to use qmesh as a library in your code, and in particular redistribute qmesh code as part of another project, pay attention to the code and documentation licences.

CHAPTER 4

Installation

4.1 Dependencies

qmesh relies on the following packages:

- QGIS
- gmsh
- GFD_basisChange_tools
- PyRDM
- setuptools-qmesh

4.2 Installing from PyPI

The easiest way to install qmesh and qmesh-cli is through the Python Package Index. To install the qmesh python API type the following at the command line:

```
pip install qmesh
```

If your system has `pip` installed, the above command will install all dependencies themselves distributed through PyPI. The install procedure will also check for gmsh and qgis and will halt if those are not installed. In your system is missing qgis, it is best to install it through your distributions package manager as documented in the qmesh web-page. If your system is missing gmsh, the best option is to obtain a compiled binary from the gmsh web-page.

To install the qmesh cli :

```
pip install qmesh-cli
```

An advantage to installing via PyPI, is the ease to upgrade and remove. Should you wish to upgrade, type the following at a terminal:

```
pip install --upgrade SomePackage <qmesh-package>
```

To remove type the following:

```
pip uninstall <qmesh-package>
```

4.3 Docker containers

Docker containers can be thought of as virtual machines, and provide isolation and control of a working environment. It is easy to construct *docker images* where the operating system has all the necessary dependencies installed. The images are typically made accessible by uploading to online *registries*. Docker hub is a popular docker registry, where developers upload images, organised into *repositories*. For example, the qmesh repository houses all qmesh images, where each image is identified by a unique image *tag*. Some images are intended for testing and development, and the environment is set-up with a minimal set of dependencies to provide a clean "sand-box". Other images provide a full qmesh installation and are aimed at facilitating use of qmesh when a user's environment makes installation hard or impossible. You can see all qmesh images in the qmesh repository tags page. Note that you must have docker installed on your system, preferably thought a package manager to use docker images.

However, since docker provides a user with a bare-bone system, it puts the task of system administration onto the container user. Administering a Linux system can be demanding for users who just want to run qmesh, rather than spend time administering a system. The package `qmeshcontainers` provides the command-line utility `qmeshcontainer`, the latter is meant to facilitate creating and running qmesh containers. The package `qmeshcontainers` is available from the **'Python Package Index https://pypi.org/project/qmeshcontainers/'_**.

A complete description of the `qmeshcontainer` utility is available in the `qmeshcontainers` manual, a brief outline of installation and use is presented here.

To install `qmeshcontainers` issue the following at the command line:

```
sudo pip install qmeshcontainers
```

The primary usage scenario of qmesh containers, is that the user will have prepared the input files in some way and want to run qmesh. In the commands below, it is assumed that the input files are located in the directory `mesh`.

From the command line of your host (computer) navigate to the `mesh` directory and enter the directory. To create and run a qmesh container issue the following at the command line:

```
qmeshcontainer -mwd
```

The above will place you inside the qmesh container, giving you a Bash prompt. If you see any warning messages about user or group details mismatch, pay attention to them, but you do not have to take any action. As the `qmeshcontainers` utility is preparing the container, it tries to match your user set-up (including user and group IDs) so it can access and write files that match your profile into the `mesh` directory. The warning messages are produced in case the utility cannot fully replicate your user set-up. The above `qmeshcontainer` command will also make the directory `mesh` available inside the container. Any changes or deletions you make to the `mesh` directory are propagated to the host, so pay attention to any file deletions or changes you might make. Navigate into the `mesh` directory in the container bash prompt:

```
cd mesh
```

Assuming that the qmesh mesh generation script is called `mesh_generation` the following command at the container bash prompt will execute the script:

```
python mesh_generation
```

Review the qmesh script output to troubleshoot any problems and rerun the script to obtain the mesh you want. You can leave the container running, edit the input files on the host machine, and return to the terminal window with the container bash prompt to re-run the meshing script. Similarly, you can use gmsh or qgis on the host machine to visualise the mesh you have produced. Once you have obtained the mesh you want you can exit the container by typing `exit` or with the keyboard shortcut `Ctrl + d`.

For more information on the `qmeshcontainer` utility, including various user options, please consult the qmesh containers manual.

CHAPTER 5

Tutorial

5.1 A Crash course in GIS

If you are an expert in GIS, you can skim this section. However, do not completely skip it, as we have included here a number of tips and tricks with GIS, that will save you a lot or time and effort when creating new meshes.

The GIS software we have used in this section is QGIS, because it is open-source and free, thus it is accesible to everyone. However, the concepts are applicable to any GIS suite. Indeed, the two most fundamental datatypes in GIS are *vector* and *raster* data, and their correspondence to mesh-generator data-structures was discussed in the *Introduction* and in [23].

Lets examine an example of each data-structure. Create a folder for the purpose of this example, named `GIS_crash_course` and inside that place the following two files:

- The filtered and subsampled GEBCO global bathymetry, which you can download from **‘figshare <>‘_**.

This is an example of a raster file, where the global bathymetry and topography is represented as data over a regular grid: a number of points along the two geodetic coordinates, longitude and latitude define “pixels” and values attached to each of these points give the value of the elevation at that point.

- The global 0-meter elevation contour, which you can download from **‘figshare <>‘_**.

This is an example of a vector file, where a sequence of points defines a line.

Place both files inside the `GIS_crash_course` folder, and open QGIS. You should have a window similar to Fig. 5.1, below.

Lets examine the major areas and tools highlighted in figure Fig. 5.1, starting with the *canvas*, and *layers* windows. Each vector or raster file is represented in GIS as a separate *layer*. The purpose of the *canvas* is to overlay all layers, and facilitate contextual comparison and editing of the layers. Lets add the files we downloaded to see how the canvas and layers windows change. You can add the bathymetry via the `Add Raster Layer` button, also highlighted in Fig. 5.1.

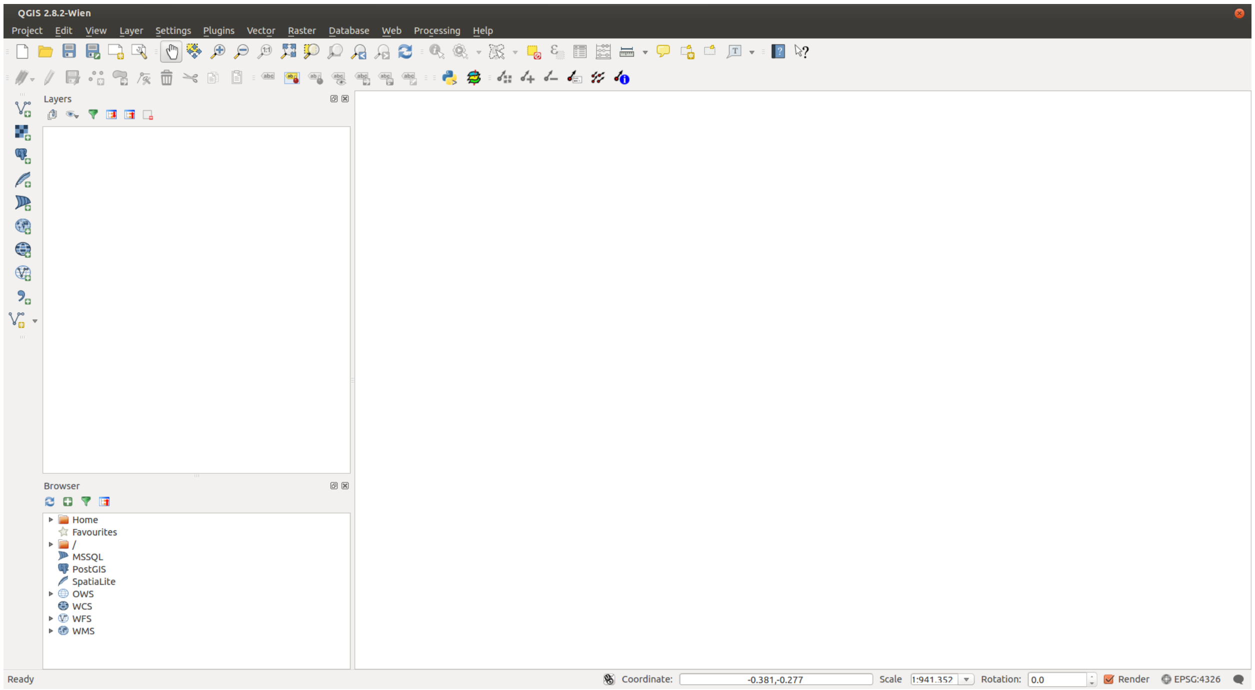

Fig. 5.1: The Qgis Window.

5.2 Global meshes

5.3 Meshing the Mediterranean Sea

5.4 Meshing the North Sea

5.5 Meshing the Severn Estuary

5.6 Meshing the Seas around the Orkney and Shetland Islands

CHAPTER 6

Further documentation

6.1 API Rerefence

6.2 CLI Rerefence

6.3 GUI Rerefence

6.4 Developer Documentation

6.5 Academic papers

6.5.1 Papers on qmesh

- Avdis, A. & Candy A. S. & Hill J. & Kramer C. S. & Piggott M. D., *"Efficient unstructured mesh generation for marine renewable energy applications"*, Renewable Energy, Volume 116, Part A, February 2018, Pages 842-856, https://doi.org/10.1016/j.renene.2017.09.058
- Jacobs C. T. & Avdis A. & Mouradian S. L. & Piggott M. D., *"Integrating Research Data Management into Geographical Information Systems"*, 5th International Workshop on Semantic Digital Archives (SDA), 2015, http://hdl.handle.net/10044/1/28557, http://ceur-ws.org/Vol-1529/

6.5.2 Papers using qmesh

- Collins D. S. & Avdis A. & Allison P. A. & Johnson H. D. & Hill J. & Piggott M. D., Hassan M. H. A. & Damit A. R., *"Tidal dynamics and mangrove carbon sequestration during the Oligo–Miocene in the South China Sea"*, Nature Communications, Volume 8, Article number: 15698, 2017, https://doi.org/10.1038/ncomms15698

- Pérez-Ortiz A. & Borthwick G. L. A. & McNaughton J. & Avdis A., *"Characterization of the tidal resource in Rathlin Sound"*, Renewable Energy, Volume 114, Part A, December 2017, Pages 229-243, https://doi.org/10.1016/j.renene.2017.04.026

6.6 Tutorials

6.6.1 Basic invocation and initialisation

initialisation, version numbers git sha keys in all APIs

6.6.2 Mesh generation for simulations of tidal flow with a turbine array in the Inner Sound

Here we demonstrate how qmesh can be used for high-resolution simulations in the context of renewable energy generation with tidal turbine farms. .. A mesh is constructed in a domain encompassing Northern Scotland, the Orkney and Shetland islands, as shown in

A dataset containing the complete definition of the domain geometry can be downloaded from figshare. A python script that uses this data can be found in the qmesh (API core) code repository and is a good example of the qmesh API: *qmesh/examples/OrkneyShetland_UTM30/OrkneyShetland_UTM30.py* run this file with *python OrkneyShetland_UTM30.py*. You will hopefully obtain a mesh similar to the one in figshare, but your mesh will not be identical to it. Differences in platforms, dependecy versions as well as differences in qmesh versions will prevent identical meshes.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Bibliography

[1] G J Gorman, M D Piggott, C C Pain, C R E de Oliveira, A P Umpleby, and A J H Goddard. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Model.*, 12(3–4):436–452, 2006.

[2] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009. URL: http://qgis.osgeo.org.

[3] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3–D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal of Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi:10.1002/nme.2579.

[4] Christian T Jacobs, Alexandros Avdis, Gerard J Gorman, and Matthew D Piggott. PyRDM: A Python-based library for automating the management and online publication of scientific software and data. *Journal of Open Research Software*, 2(1):e28, 2014. doi:10.5334/jors.bj.

[5] C. T. Jacobs, A. Avdis, S. L. Mouradian, and M. D. Piggott. Integrating Research Data Management into Geographical Information Systems. In *Proceedings of the 5th International Workshop on Semantic Digital Archives*. 2015. URL: http://hdl.handle.net/10044/1/28557.

[6] Jonathan Lambrechts, Richard Comblen, Vincent Legat, Christophe Geuzaine, and Jean-François Remacle. Multiscale mesh generation on the sphere. *Ocean Dynamics*, 58(5-6):461–473, 8 October 2008.

[7] Thomas J Heinzer, M Diane Williams, Emin C Dogrul, Tariq N Kadir, Charles F Brush, and Francis I Chung. Implementation of a feature-constraint mesh generation algorithm within a GIS. *Comput. Geosci.*, 49:46–52, December 2012.

[8] Pat Prodanovic. QGIS as a pre-and post-processor for TELEMAC: mesh generation and output visualization. In *22nd TELEMAC-MASCARET User Conference*. 15 October 2015.

[9] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. URL: http://ipython.org, doi:10.1109/MCSE.2007.53.

[10] G J Gorman, M Piggott, M R Wells, C C Pain, and P A Allison. A systematic approach to unstructured mesh generation for ocean modelling using GMT and terreno. *Computers and Geosciences*, 34(12):1721–1731, December 2008.

[11] Paul Wessel and Walter H F Smith. Free software helps map and display data. *Eos Transactions AGU*, 72(41):441–446, 1991.

[12] ESRI. ESRI shapefile technical description. Technical Report J-7855, Environmental Systems Research Institute, July 1998.

[13] S Legrand, V Legat, and E Deleersnijder. Delaunay mesh generation for an unstructured–grid ocean general circulation model. *Ocean Modelling*, 2(1–2):17–28, 2000.

[14] Sébastien Legrand, Eric Deleersnijder, Eric Delhez, and Vincent Legat. Unstructured, anisotropic mesh generation for the northwestern european continental shelf, the continental slope and the neighbouring ocean. *Continental Shelf Research*, 27(9):1344–1356, 15 May 2007.

[15] Jean-François Remacle and Jonathan Lambrechts. Fast and Robust Mesh Generation on the Sphere–Application to Coastal Domains. *Procedia Engineering*, 163:20–32, 2016.

[16] Matrin R Wells, Peter A Allison, Matthew D Piggott, Gary J Hampson, Christopher C Pain, and Gerard J Gorman. Tidal modelling of an ancient tide-dominated seaway, Part 1: Model validation and application to global Cretaceous (Aptian) tides. *Journal of Sedimentary Research*, 80:393–410, 2010. doi:10.2110/jsr.2010.044.

[17] Andrew J Mitchell, Peter A Allison, Gerald J Gorman, Matthew D Piggott, and Christopher C Pain. Tidal calculation in an ancient epicontinental sea: The Early Jurassic Laurasian Seaway. *Geology*, 39(3):207–210, 2011. doi:10.1130/G31496.1.

[18] Jon Hill, Gareth S Collins, Alexandros Avdis, Stephan C Kramer, and Matthew D Piggott. How does multiscale modelling and inclusion of realistic palaeobathymetry affect numerical simulation of the Storegga Slide tsunami? *Ocean Modelling*, 83:11–25, 2014. doi:10.1016/j.ocemod.2014.08.007.

[19] Technical Committee ISO/TC 46 (Information and documentation), Subcommittee SC 9 (Identification and description). ISO 26324:2012 Information and documentation – Digital object identifier system. Technical Report, International Organisation for Standardization, 2012.

[20] Alexandros Avdis, T Christian Jacobs, Leo Simon Mouradian, Jon Hill, and D Matthew Piggott. Meshing ocean domains for coastal engineering applications. In M Papadrakakis, V Papadopoulos, G Stefanou, and V Plevris, editors, *The European Community on Computational Methods in Applied Sciences and Engineering (ECCOMAS) VII Congress Proceedings, Crete, Greece, June 5–10*. 2016. doi:10.6084/m9.figshare.4524629.

[21] A Pèrez-Ortiz, AGL Borthwick, J McNaughton, and A Avdis. Characterization of the tidal resource in rathlin sound. *Renewable Energy*, 2017. doi:10.1016/j.renene.2017.04.026.

[22] Daniel S Collins, Alexandros Avdis, Peter A Allison, H D Johnson, Jon Hill, Matthew D Piggott, M H A Hassan, and A R Damit. Tidal dynamics and mangrove carbon sequestration during the Oligo–Miocene in the south china sea. *Nature communications*, 2017. doi:10.1038/ncomms15698.

[23] Alexandros Avdis, Adam S Candy, Jon Hill, Stephan C Kramer, and Matthew D Piggott. Efficient unstructured mesh generation for marine renewable energy applications. *Renewable Energy*, 116:842–856, 2018. doi:10.1016/j.renene.2017.09.058.

[24] L Luo, J Wang, DJ Schwab, H Vanderploeg, G Leshkevich, X Bai, H Hu, and D Wang. Simulating the 1998 spring bloom in Lake Michigan using a coupled physical-biological model. *Journal of Geophysical Research C: Oceans*, 2012. doi:10.1029/2012JC008216.

[25] N M Hunter, P D Bates, S Neelz, G Pender, I Villanueva, N G Wright, D Liang, R A Falconer, B Lin, S Waller, A J Crossley, and D Mason. Benchmarking 2D hydraulic models for urban flood simulations. *Proceedings of the Institution of Civil Engineers: Water Management*, 161(1):13–30, February 2008.

[26] P Berry, S Bonduá, V Bortolotti, C Cormio, and E M Vasini. A GIS-based open source pre-processor for georesources numerical modeling. *Environmental Modelling & Software*, 62:52–64, 2014.

[27] Benoit B Mandelbrot and John A Wheeler. The fractal geometry of nature. *American Journal of Physics*, 51(3):286–287, 1 March 1983.

[28] Junqiang Xia, Roger A Falconer, and Binliang Lin. Impact of different tidal renewable energy projects on the hydrodynamic processes in the Severn Estuary, UK. *Ocean Modelling*, 32(1–2):86–104, 2010.

[29] Olivia Bentley, L, Matthew D Piggott, Alexandros Avdis, and Peter Allison, A. Simulating mixing due to internal wave generation, propagation and breaking with adaptive mesh methods. *Ocean Modelling*, submitted, 2016.

[30] S Danilov. Ocean modeling on unstructured meshes. *Ocean Modelling*, 69:195–210, 2013. doi:10.1016/j.ocemod.2013.05.005.

[31] A Keith Turner. Challenges and trends for geological modelling and visualisation. *Bulletin of Engineering Geology and the Environment*, 65(2):109–127, 7 December 2005. doi:10.1007/s10064-005-0015-0.

[32] R J LeVeque, I M Mitchell, and V Stodden. Reproducible Research for Scientific Computing: Tools and Strategies for Changing the Culture. *Computing in Science and Engineering*, 14(4):13–17, 2012. doi:10.1109/MCSE.2012.38.

[33] European Commission. Commission recommendation on access to and preservation of scientific information. 17 July 2012.

[34] European Commission. Communication towards better access to scientific information. 17 July 2012.

[35] RCUK. Guidance on best practice in the management of research data. 2015.

[36] Directive 2003/35/ec of the european parliament and of the council on providing for public participation in respect of the drawing up of certain plans and programmes relating to the environment and amending with regard to public participation and access to justice council directives 85/337/EEC and 96/61/EC – statement by the commission. 26 May 2003.

[37] Harbours, Docks, Piers and Ferries: The Harbour Works (Environmental Impact Assessment) Regulations 1999 No. 3445. 21 December 1999. URL: https://www.legislation.gov.uk/uksi/1999/3445/pdfs/uksi_19993445_en.pdf.

[38] Environmental protection: the marine works (environmental impact assessment) regulations 2007 no. 1518. 21 May 2007.

[39] Electricity: The Offshore Electricity Development (Environmental Impact Assessment) Regulations (Northern Ireland) 2008 No. 55. 7 February 2008.

[40] Directive 2003/4/ec of the european parliament and of the council on public access to environmental information. 28 January 2003.

[41] Freedom of information, environmental protection: the environmental information regulations 2004 no. 3391. 21 December 2004.

[42] Canadian Hydraulic Centre. *Reference manual for Blue Kenue*. National Research Council, 1998.

[1] G J Gorman, M D Piggott, C C Pain, C R E de Oliveira, A P Umpleby, and A J H Goddard. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Model.*, 12(3–4):436–452, 2006.

[2] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009. URL: http://qgis.osgeo.org.

[3] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3–D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal of Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi:10.1002/nme.2579.

[4] Christian T Jacobs, Alexandros Avdis, Gerard J Gorman, and Matthew D Piggott. PyRDM: A Python-based library for automating the management and online publication of scientific software and data. *Journal of Open Research Software*, 2(1):e28, 2014. doi:10.5334/jors.bj.

[5] C. T. Jacobs, A. Avdis, S. L. Mouradian, and M. D. Piggott. Integrating Research Data Management into Geographical Information Systems. In *Proceedings of the 5th International Workshop on Semantic Digital Archives*. 2015. URL: http://hdl.handle.net/10044/1/28557.

[6] Jonathan Lambrechts, Richard Comblen, Vincent Legat, Christophe Geuzaine, and Jean-François Remacle. Multiscale mesh generation on the sphere. *Ocean Dynamics*, 58(5-6):461–473, 8 October 2008.

[7] Thomas J Heinzer, M Diane Williams, Emin C Dogrul, Tariq N Kadir, Charles F Brush, and Francis I Chung. Implementation of a feature-constraint mesh generation algorithm within a GIS. *Comput. Geosci.*, 49:46–52, December 2012.

[8] Pat Prodanovic. QGIS as a pre-and post-processor for TELEMAC: mesh generation and output visualization. In *22nd TELEMAC-MASCARET User Conference*. 15 October 2015.

[9] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. URL: http://ipython.org, doi:10.1109/MCSE.2007.53.

[10] G J Gorman, M Piggott, M R Wells, C C Pain, and P A Allison. A systematic approach to unstructured mesh generation for ocean modelling using GMT and terreno. *Computers and Geosciences*, 34(12):1721–1731, December 2008.

[11] Paul Wessel and Walter H F Smith. Free software helps map and display data. *Eos Transactions AGU*, 72(41):441–446, 1991.

[12] ESRI. ESRI shapefile technical description. Technical Report J-7855, Environmental Systems Research Institute, July 1998.

[13] S Legrand, V Legat, and E Deleersnijder. Delaunay mesh generation for an unstructured–grid ocean general circulation model. *Ocean Modelling*, 2(1–2):17–28, 2000.

[14] Sébastien Legrand, Eric Deleersnijder, Eric Delhez, and Vincent Legat. Unstructured, anisotropic mesh generation for the northwestern european continental shelf, the continental slope and the neighbouring ocean. *Continental Shelf Research*, 27(9):1344–1356, 15 May 2007.

[15] Jean-François Remacle and Jonathan Lambrechts. Fast and Robust Mesh Generation on the Sphere–Application to Coastal Domains. *Procedia Engineering*, 163:20–32, 2016.

[16] Matrin R Wells, Peter A Allison, Matthew D Piggott, Gary J Hampson, Christopher C Pain, and Gerard J Gorman. Tidal modelling of an ancient tide-dominated seaway, Part 1: Model validation and application to global Cretaceous (Aptian) tides. *Journal of Sedimentary Research*, 80:393–410, 2010. doi:10.2110/jsr.2010.044.

[17] Andrew J Mitchell, Peter A Allison, Gerald J Gorman, Matthew D Piggott, and Christopher C Pain. Tidal calculation in an ancient epicontinental sea: The Early Jurassic Laurasian Seaway. *Geology*, 39(3):207–210, 2011. doi:10.1130/G31496.1.

[18] Jon Hill, Gareth S Collins, Alexandros Avdis, Stephan C Kramer, and Matthew D Piggott. How does multiscale modelling and inclusion of realistic palaeobathymetry affect numerical simulation of the Storegga Slide tsunami? *Ocean Modelling*, 83:11–25, 2014. doi:10.1016/j.ocemod.2014.08.007.

[19] Technical Committee ISO/TC 46 (Information and documentation), Subcommittee SC 9 (Identification and description). ISO 26324:2012 Information and documentation – Digital object identifier system. Technical Report, International Organisation for Standardization, 2012.

[1] G J Gorman, M D Piggott, C C Pain, C R E de Oliveira, A P Umpleby, and A J H Goddard. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Model.*, 12(3–4):436–452, 2006.

[2] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009. URL: http://qgis.osgeo.org.

[3] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3–D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal of Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi:10.1002/nme.2579.

[4] Christian T Jacobs, Alexandros Avdis, Gerard J Gorman, and Matthew D Piggott. PyRDM: A Python-based library for automating the management and online publication of scientific software and data. *Journal of Open Research Software*, 2(1):e28, 2014. doi:10.5334/jors.bj.

[5] C. T. Jacobs, A. Avdis, S. L. Mouradian, and M. D. Piggott. Integrating Research Data Management into Geographical Information Systems. In *Proceedings of the 5th International Workshop on Semantic Digital Archives*. 2015. URL: http://hdl.handle.net/10044/1/28557.

[6] Jonathan Lambrechts, Richard Comblen, Vincent Legat, Christophe Geuzaine, and Jean-François Remacle. Multiscale mesh generation on the sphere. *Ocean Dynamics*, 58(5-6):461–473, 8 October 2008.

[7] Thomas J Heinzer, M Diane Williams, Emin C Dogrul, Tariq N Kadir, Charles F Brush, and Francis I Chung. Implementation of a feature-constraint mesh generation algorithm within a GIS. *Comput. Geosci.*, 49:46–52, December 2012.

[8] Pat Prodanovic. QGIS as a pre-and post-processor for TELEMAC: mesh generation and output visualization. In *22nd TELEMAC-MASCARET User Conference*. 15 October 2015.

[9] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. URL: http://ipython.org, doi:10.1109/MCSE.2007.53.

[10] G J Gorman, M Piggott, M R Wells, C C Pain, and P A Allison. A systematic approach to unstructured mesh generation for ocean modelling using GMT and terreno. *Computers and Geosciences*, 34(12):1721–1731, December 2008.

[11] Paul Wessel and Walter H F Smith. Free software helps map and display data. *Eos Transactions AGU*, 72(41):441–446, 1991.

[12] ESRI. ESRI shapefile technical description. Technical Report J-7855, Environmental Systems Research Institute, July 1998.

[13] S Legrand, V Legat, and E Deleersnijder. Delaunay mesh generation for an unstructured–grid ocean general circulation model. *Ocean Modelling*, 2(1–2):17–28, 2000.

[14] Sébastien Legrand, Eric Deleersnijder, Eric Delhez, and Vincent Legat. Unstructured, anisotropic mesh generation for the northwestern european continental shelf, the continental slope and the neighbouring ocean. *Continental Shelf Research*, 27(9):1344–1356, 15 May 2007.

[15] Jean-François Remacle and Jonathan Lambrechts. Fast and Robust Mesh Generation on the Sphere–Application to Coastal Domains. *Procedia Engineering*, 163:20–32, 2016.

[16] Matrin R Wells, Peter A Allison, Matthew D Piggott, Gary J Hampson, Christopher C Pain, and Gerard J Gorman. Tidal modelling of an ancient tide-dominated seaway, Part 1: Model validation and application to global Cretaceous (Aptian) tides. *Journal of Sedimentary Research*, 80:393–410, 2010. doi:10.2110/jsr.2010.044.

[17] Andrew J Mitchell, Peter A Allison, Gerald J Gorman, Matthew D Piggott, and Christopher C Pain. Tidal calculation in an ancient epicontinental sea: The Early Jurassic Laurasian Seaway. *Geology*, 39(3):207–210, 2011. doi:10.1130/G31496.1.

[18] Jon Hill, Gareth S Collins, Alexandros Avdis, Stephan C Kramer, and Matthew D Piggott. How does multiscale modelling and inclusion of realistic palaeobathymetry affect numerical simulation of the Storegga Slide tsunami? *Ocean Modelling*, 83:11–25, 2014. doi:10.1016/j.ocemod.2014.08.007.

[19] Technical Committee ISO/TC 46 (Information and documentation), Subcommittee SC 9 (Identification and description). ISO 26324:2012 Information and documentation – Digital object identifier system. Technical Report, International Organisation for Standardization, 2012.

[20] Alexandros Avdis, T Christian Jacobs, Leo Simon Mouradian, Jon Hill, and D Matthew Piggott. Meshing ocean domains for coastal engineering applications. In M Papadrakakis, V Papadopoulos, G Stefanou, and V Plevris, editors, *The European Community on Computational Methods in Applied Sciences and Engineering (ECCOMAS) VII Congress Proceedings, Crete, Greece, June 5–10*. 2016. doi:10.6084/m9.figshare.4524629.

[21] A Pèrez-Ortiz, AGL Borthwick, J McNaughton, and A Avdis. Characterization of the tidal resource in rathlin sound. *Renewable Energy*, 2017. doi:10.1016/j.renene.2017.04.026.

[22] Daniel S Collins, Alexandros Avdis, Peter A Allison, H D Johnson, Jon Hill, Matthew D Piggott, M H A Hassan, and A R Damit. Tidal dynamics and mangrove carbon sequestration during the Oligo–Miocene in the south china sea. *Nature communications*, 2017. doi:10.1038/ncomms15698.

[23] Alexandros Avdis, Adam S Candy, Jon Hill, Stephan C Kramer, and Matthew D Piggott. Efficient unstructured mesh generation for marine renewable energy applications. *Renewable Energy*, 116:842–856, 2018. doi:10.1016/j.renene.2017.09.058.