SoftwareX 7 (2018) 41–46

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

FacetModeller: Software for manual creation, manipulation and analysis of 3D surface-based models

Peter G. Lelièvre *, Angela E. Carter-McAuslan, Michael W. Dunham, Drew J. Jones, Mariella Nalepa, Chelsea L. Squires, Cassandra J. Tycholiz, Marc A. Vallée, Colin G. Farquharson

Memorial University of Newfoundland, St. John's, NL, Canada

ARTICLE INFO

Article history:
Received 21 November 2017
Received in revised form 12 February 2018
Accepted 13 February 2018

Keywords:
3D model
Geological model
Meshing
Manual model building
Numerical methods
Java

ABSTRACT

The creation of 3D models is commonplace in many disciplines. Models are often built from a collection of tessellated surfaces. To apply numerical methods to such models it is often necessary to generate a mesh of space-filling elements that conforms to the model surfaces. While there are meshing algorithms that can do so, they place restrictive requirements on the surface-based models that are rarely met by existing 3D model building software. Hence, we have developed a Java application named FacetModeller, designed for efficient manual creation, modification and analysis of 3D surface-based models destined for use in numerical modelling.

Code metadata

Current code version	3.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-17-00091
Legal Code License	GNU General Public License (GPL)
Code versioning system used	git
Software code languages, tools, and services used	Java
Compilation requirements, operating environments & dependencies	The github link above contains a NetBeans project FacetModeller which uses MyLibrary (also supplied at github link above)
If available Link to developer documentation/manual	*will be made available on github*
Support email for questions	plelievre@mun.ca

1. Motivation and significance

The creation of 3D models for visualization or quantitative analysis is commonplace in many disciplines. For example, in applied geophysics, 3D models are often created, modified and queried throughout the life of an exploration project to determine the nature and composition of the Earth volume of interest (VOI). Models are often built from a collection of surfaces that comprise tessellated triangles or other planar polygonal shapes. For example, in geological models, these surfaces define the contacts between different rock units; refer to [1] for a discussion of some of the techniques typically used to generate those surfaces. We refer to the tessellated triangles or polygons as "facets" and these types of models as "surface-based" models.

Computer modelling in many disciplines involves numerical calculation of physical phenomena that are affected by spatial variations in the physical properties in the VOI. Many numerical methods discretize the VOI on a mesh of space-filling elements, often referred to as mesh "cells". For example, a 3D rectilinear mesh comprises rectangular prisms arranged in a structured grid; a 3D unstructured tetrahedral mesh comprises tetrahedra. To apply numerical methods to surface-based models it is therefore often necessary to generate a mesh that conforms to the model surfaces. For example, in applied geophysics, 3D surface-based geological

* Corresponding author.
E-mail address: plelievre@mun.ca (P.G. Lelièvre).

https://doi.org/10.1016/j.softx.2018.02.002
2352-7110/

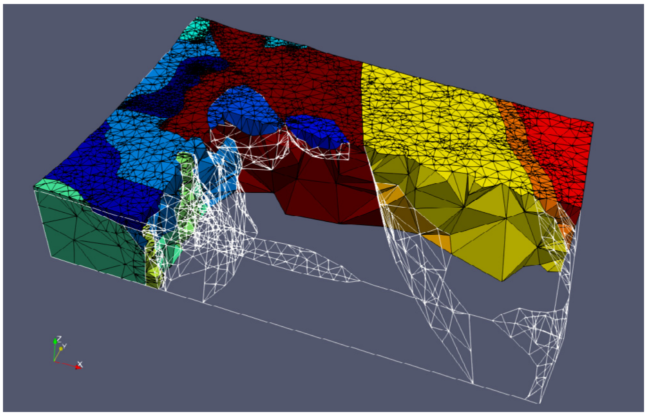

Fig. 1. A mesh of tetrahedral cells that conforms to a surface-based model (a collection of tessellated triangles). The edges of the facets in the surface-based model are white, the mesh tetrahedra are coloured such that each different region in the model has a different colour, and the edges of the mesh tetrahedra are black. Some of the tetrahedral cells have been removed from the southeast of the mesh to expose the surfaces lying within the mesh. This figure is a screenshot from ParaView [14]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

models may be used to help constrain geophysical forward and inverse modelling performed on conforming meshes [e.g. 2–8].

Surface-based models can be fed into meshing algorithms that generate unstructured meshes. For example, the triangular facets in a surface-based model become the faces of the tetrahedra in a mesh: see Fig. 1. There are many mesh generation software packages for doing so [e.g. 9–11] and for improving such meshes [e.g. 12]. For most mesh generation methods, the input must be a piecewise linear complex (PLC), a concept first introduced by [13], representing the 3D domain to be meshed.

Ref. [15] provides the requirements that a PLC must satisfy, which we paraphrase here. A PLC is a set of planar polygonal facets that satisfies the following properties:

- the boundary of every facet (an edge) is a union of facets
- if two distinct facets intersect then their intersection is a union of facets.

These requirements are demonstrated visually in Fig. 2. We use the term "node" to denote the vertices of the facets in a PLC, and we use "edges" to denote line-segments that connect pairs of nodes, that is, the edges of the facets.

The above requirements on a PLC must apply to any surface-based model with which we wish to generate a conforming mesh for performing numerical computations. For such computations, further requirements may exist related to the "quality" of the mesh derived from the PLC. For example, when finite volume or finite element methods are used to simulate physical phenomena, numerical modelling accuracy and solution times can depend critically on the quality of the unstructured mesh [e.g. 16–21]. Obtaining an acceptable mesh quality may only be possible if the input PLC is itself of a high enough quality.

The definition of mesh quality depends on the intended application and numerical methods employed but is generally related to the geometry of the tetrahedral mesh cells [22]. A general guideline is that tetrahedral cells with very small or large dihedral angles should be avoided. Hence, similar considerations apply to the input PLC: triangular facets with very small or large vertex angles should be avoided.

There are many software packages that are capable of building 3D surface-based models but to date there are none that have fully met our needs for generating quality PLCs for use in our geophysical computations [e.g. 17–20]. Often, such software uses automated procedures that may, for example, interpolate curves and surfaces between sparse measured data, and automatically generate the triangular tessellations on the interpolated surfaces. Such automated processing reduces the model building time but can introduce some unwanted artefacts, even if there is some user control on the automated processes.

Recent research is attempting to ameliorate some of these issues through numerical procedures [e.g. 23]. However, automated model building tools may provide less-than optimal results and workflows must then be devised using different model building and mesh generation software [e.g. 24]. Such workflows can only benefit from the availability of software that eases the development of such workflows [e.g. 25,26] and provides helpful tools for manual model building tasks.

2. Software description

2.1. Software architecture

FacetModeller is a Java application designed for efficient manual creation, modification and analysis of quality 3D PLCs, for example geological surface-based models destined for use in geophysical numerical modelling. FacetModeller is not intended to be a replacement for other 3D model building software packages: the intention is not to duplicate the automatic model building tasks that such packages can perform, but instead provide an efficient tool for manual model building tasks. FacetModeller is designed to help users honour the requirements of a valid PLC and control the quality of their surface-based models.

3D surface-based models in FacetModeller comprise the following components:

- Different parts of the model can be assigned to different "sections". These may be: concrete, spatially-connected objects, i.e. planar slices through the 3D model; or abstract containers for organizing different pieces of information. Each section may or may not have an image associated with it.
- The basic building blocks of a surface-based model are nodes and facets. Nodes are attached to different sections. Facets are specific connections between nodes.
- Different types of nodes and facets are distinguished by assigning them to different "groups". Each group has user-defined drawing colours associated with it. The concepts of sections and groups can be used to help organize the different parts of a model.

There are two main data input types for FacetModeller:

- images that must be digitized; for example, geological maps, interpolated vertical cross sections, or interpolated horizontal depth sections
- pre-defined 3D model surfaces, where each surface may be a tessellation of facets or a collection of unconnected nodes lying on the surface.

Fig. 3 shows the FacetModeller GUI. As FacetModeller is a Java program, the GUI may look different on different operating systems. The FacetModeller window is split into three main panels. On the left are various buttons and selection boxes that allow users to select which objects they wish to display and work with. In the centre of the GUI is a 2D viewing panel and on the right is a 3D viewing panel. In the 2D viewing panel, users can define nodes and facets via cursor interactions. The 3D viewing panel is only used for viewing: no model building occurs directly through user interaction with the 3D viewing panel. The 2D viewing panel