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

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

[a]*Memorial University of Newfoundland, St. John's, NL, Canada*

Abstract

The creation of 3D models for visualization or quantitative analysis is commonplace in applied geophysics, where 3D models are often created, modified and queried throughout the life of a project. Typical geological models comprise a collection of surfaces that define the contacts between different rock units. The surfaces comprise tessellated triangles or other planar polygonal shapes. Computer modelling in geophysics involves numerical calculation of physical fields that are affected by spatial variations in the physical properties in the Earth. Most geophysical numerical methods discretize the Earth volume of interest on a mesh of space-filling elements. Often, 3D geological models are used to help constrain geophysical forward and inverse modelling; ideally, one wishes to generate a mesh-based geophysical model that conforms to the surface-based geological model. While there are meshing algorithms that can do so, they place certain restrictive requirements on the surface-based models. Unfortunately, those requirements are rarely met by existing software packages that are capable of building 3D surface-based models because those packages use automated interpolation methods that are not designed with those requirements in mind. To date there are no software packages for building 3D models that have met our needs for generating quality surface-based representations of the Earth for use in our geophysical computations. Therefore, we have developed a Java application named FacetModeller, designed for efficient manual creation, modification and analysis of quality 3D surface-based models destined for use in geophysical numerical modelling. The intention is not to duplicate the automatic model building tasks that other software packages can perform but, instead, to provide an effective tool for manual model building tasks.

Keywords: 3D model, geological model, meshing, modelling, numerical methods, Java

1. Introduction

1.1. 3D geological models

The creation of 3D models for visualization or quantitative analysis is commonplace in applied geoscience disciplines. In applied geophysics, 3D models are often created, modified and queried throughout the life of a project to determine the nature and composition of the Earth volume of interest (VOI). In mineral exploration, mineral deposit models are typically built from geological information available at sparse point locations across the Earth's topographical surface, for example observations of outcropping rocks, or beneath the Earth's topographical surface, for example downhole measurements or data collected from recovered drilling core. These observations or measurements indicate the positions and possibly the orientations of geological contacts between different rock units. Often, experts knowledgeable of the geological features of a particu-

*Principal corresponding author

**Corresponding author

Email addresses: `plelievre@mun.ca` (Peter G. Lelièvre), `acartermcauslan@mun.ca` (Angela E. Carter-McAuslan), `mwd553@mun.ca` (Michael W. Dunham), `djj326@mun.ca` (Drew J. Jones), `mvallee@mun.ca` (Marc A. Vallée), `cgfarqh@mun.ca` (Colin G. Farquharson)

lar site will interpret and interpolate information between the sparse data measurements, for example interpolated 2D cross sections, which also feed into the development of a 3D geological model.

Typical geological models comprise a collection of surfaces that define the contacts between different rock units. Refer to Caumon et al. (2009) for a discussion of some of the techniques typically used to generate those surfaces. The surfaces comprise tessellated triangles or other planar polygonal shapes. We refer to those triangles or polygons as “facets” and these types of models as “surface-based” models. Fig. 1 shows an example of such a model.

1.2. 3D geophysical models

Computer modelling in geophysics involves numerical calculation of physical fields that are affected by spatial variations in the physical properties in the VOI. Most geophysical numerical methods discretize the VOI on some sort of 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.

Often, 3D geological models are used to help constrain geophysical forward and inverse modelling (e.g. Phillips et al., 2001; Fullagar et al., 2004; McGaughey, 2006, 2007; Farquharson et al., 2008; Guillen et al., 2008; Lelièvre and Farquharson, 2013). Ideally, we want to generate a mesh-based model that conforms to the surface-based geological model. 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. The CGAL Project, 2017; Los Alamos National Laboratory, 2017; Si, 2017) and for improving such meshes (e.g. Klingner and Shewchuk, 2017). A thorough comparison and review of all available software is beyond the scope of this paper. However, we mention TetGen Si (2015, 2017), a freely available program that we use in this paper, but we are in no way arguing for the general use of TetGen versus other meshing software alternatives. For TetGen and most mesh generation methods, the input is a piecewise linear complex (PLC), a concept first introduced by Miller et al. (1996), representing the 3D domain to be meshed.

1.3. Piecewise linear complexes

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.

Si (2015) 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 each 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 and are sensible requirements for geological models that represent a solid volume of the Earth.

1.4. Mesh quality

The above requirements on a PLC must apply to any geological surface-based model with which we wish to perform geophysical 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. Rücker et al., 2006; Lelièvre et al., 2011; Jahandari and Farquharson, 2013; Ansari and Farquharson, 2014; Jahandari and Farquharson, 2014; Cai et al., 2017). 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 (see Shewchuk,

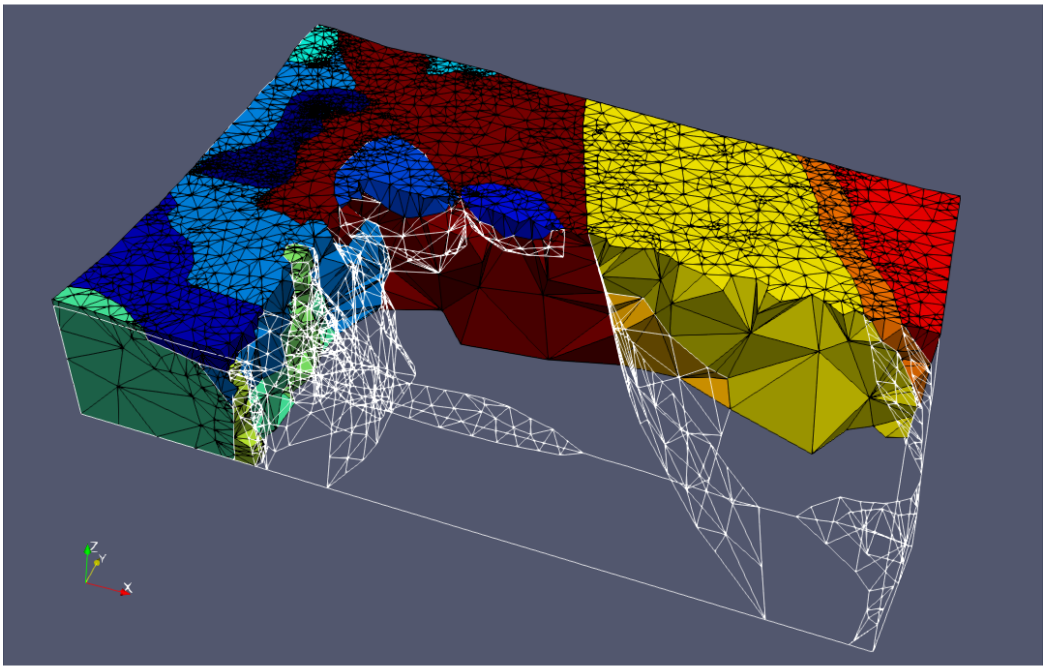

Figure 1: A mesh of tetrahedral cells that conforms to a surface-based model (a collection of tessellated triangles). Some of the tetrahedral cells have been removed from the southeast of the mesh to expose the surfaces lying within the mesh. The edges of the facets in the surface-based model are white, 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. This figure is a screenshot from ParaView (Kitware, Inc. and Los Alamos National Laboratory, 2017).

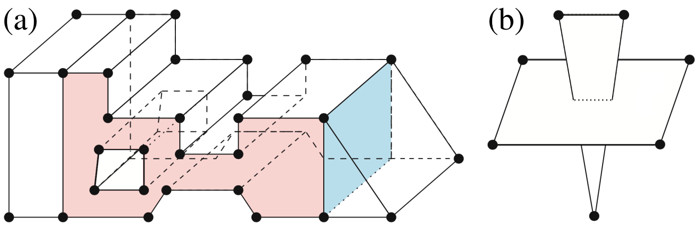

Figure 2: (a) A 3D piecewise linear complex: the pink shaded facet is on the boundary of the domain and has a hole in it; the blue shaded facet is an interior polygon separating two sub-domains. (b) A configuration of two facets that breaks the rules of a valid PLC because there is no edge at the intersection. Nodes are drawn as black dots and edge elements as black lines. Adapted from Si (2015).

2002). 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.

1.5. Existing surface-based 3D model building software

There are many software packages that are capable of building 3D surface-based models but to date there are none that have met our needs for generating quality surface-based representations of the Earth for use in our geophysical computations (e.g. see Lelièvre et al., 2011; Jahandari and Farquharson, 2013; Ansari and Farquharson, 2014; Jahandari and Farquharson, 2014). A thorough comparison and review of all the available software for 3D model building is beyond the scope of this paper, and is not possible given that many of the options are costly. Hence, we choose not to mention any specific software options here and instead we provide a general discussion of the limitations of existing software.

With all the available software, the end result is a surface-based geological model built with the help from some automated model building algorithms. Those automated procedures may, for example, interpolate curves and surfaces between sparse measured data located at outcrops and down boreholes, and automatically generate the triangular tessellations on the interpolated surfaces; again, refer to Caumon et al. (2009). The automated processing reduces the model building time but can introduce some unwanted artifacts, even if there is some user control on the automated processes.

Different software packages are designed for use in different fields, in which the constraints on the model characteristics can differ significantly. Hence, the models built may comprise several surfaces, each built independently, with little or no attempt made to avoid intersections between those surfaces or connect them properly. Such models do not meet the requirements of valid PLCs mentioned above and therefore can not be used for meshing and subsequent geophysical numerical modelling. For example, consider Fig. 3: to be geologically consistent, the ore body should be entirely contained within the troctolite; however, these two surfaces intersect each other at many places.

Sometimes quantitative data analyses, for example to estimate volumes of ore reserves, require that the surfaces intersect but without the more rigorous requirements of a valid PLC. Furthermore, the surfaces may be poorly triangulated, which is often the case when visualization is the main consideration, meaning the surfaces may be of inadequate quality for generating meshes for numerical modelling work. For example, consider the long, skinny triangular facets seen at the bottom left of Fig. 3 and elsewhere in that troctolite surface.

Recent research is attempting to ameliorate some of these issues through numerical procedures (e.g. Pellerin et al., 2014). 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. Zehner et al., 2015). Such workflows can only benefit from the availability of software that eases the development of such workflows (e.g. GeoRessources and CREGU, 2017; Pellerin et al., 2017) and provides helpful tools for manual model building tasks.

1.6. A new software for manual creation, manipulation and analysis of 3D surface-based models: FacetModeller

FacetModeller is a Java application that we have developed 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, to 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.

In the remainder of this paper, we introduce the model building philosophy of FacetModeller and provide an overview of the different components of the Graphical User Interface (GUI). Two 3D model building examples are provided to demonstrate some of the more important features of FacetModeller and provide a deeper discussion of the model building approach and the GUI components. The first example builds a simpler synthetic model and illustrates the basic workings of FacetModeller. The second example builds a more complicated model based on real-Earth geology to illustrate how some more difficult model building tasks are dealt with in FacetModeller.

2. Model components in FacetModeller

2.1. Sections

FacetModeller was originally designed to build models by connecting different 2D sections into a 3D model. A section is a planar slice through a 3D model. A section may be a vertical cross section, a horizontal depth section, or any planar rectangular section of arbitrary orientation. Each section may have an image associated with it. For example, if a geologist has drawn an interpolated cross section, that image can be imported into FacetModeller, calibrated (spatially registered), and digitized by defining nodes and facets that are attached to it.

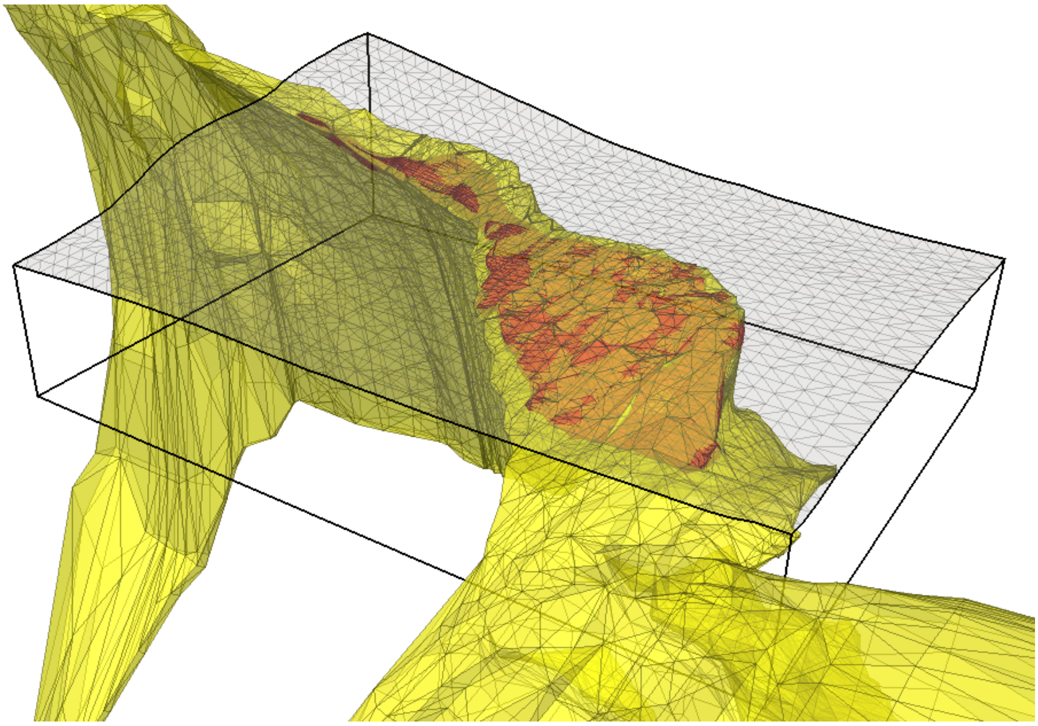

Figure 3: Perspective views of the Voiseys Bay ovoid ore body (red), troctolite (yellow) and topography surface (gray) represented by surfaces comprising tessellated triangles. The surfaces are transparent and have darker lines delineating the edges of their triangular facets. Reproduced from Lelièvre et al. (2012).

FacetModeller has grown since its conception to provide functionality far beyond the strategy of building 3D models from 2D sections. However, we find it helpful to maintain the idea of sections so that different parts of the model can be assigned to different "sections", be they concrete, spatially-connected objects or abstract containers for organizing different pieces of information.

2.2. *Nodes and facets*

Nodes and facets are the basic building blocks of any PLC. Nodes are attached to different sections. Facets are specific connections between nodes.

2.3. *Groups*

FacetModeller distinguishes different types of nodes and facets by assigning them to different "groups". Each group has user-defined drawing colours associated with it. The concept of groups can be used to help organize the different parts of a model.

3. Model building example 1: folded sedimentary layers cross-cut by igneous units

In this example we build a synthetic model that was originally developed for teaching purposes. The model is built from the geological map and interpreted geological sections in Fig. 4. The geological scenario involves sedimentary layers that have been folded into a synform plunging to the south-east and then cross-cut by igneous units.

3.1. *Initial steps*

There are two main data input types for FacetModeller:

1. Images that must be digitized. For example, geological maps, interpolated vertical cross sections, or interpolated horizontal depth sections.
2. Pre-defined 3D model surfaces. Each may be a tessellation of facets or a collection of unconnected nodes lying on the surface.

In this first example we work with the first type of data. The interpolated section images should show the contacts between

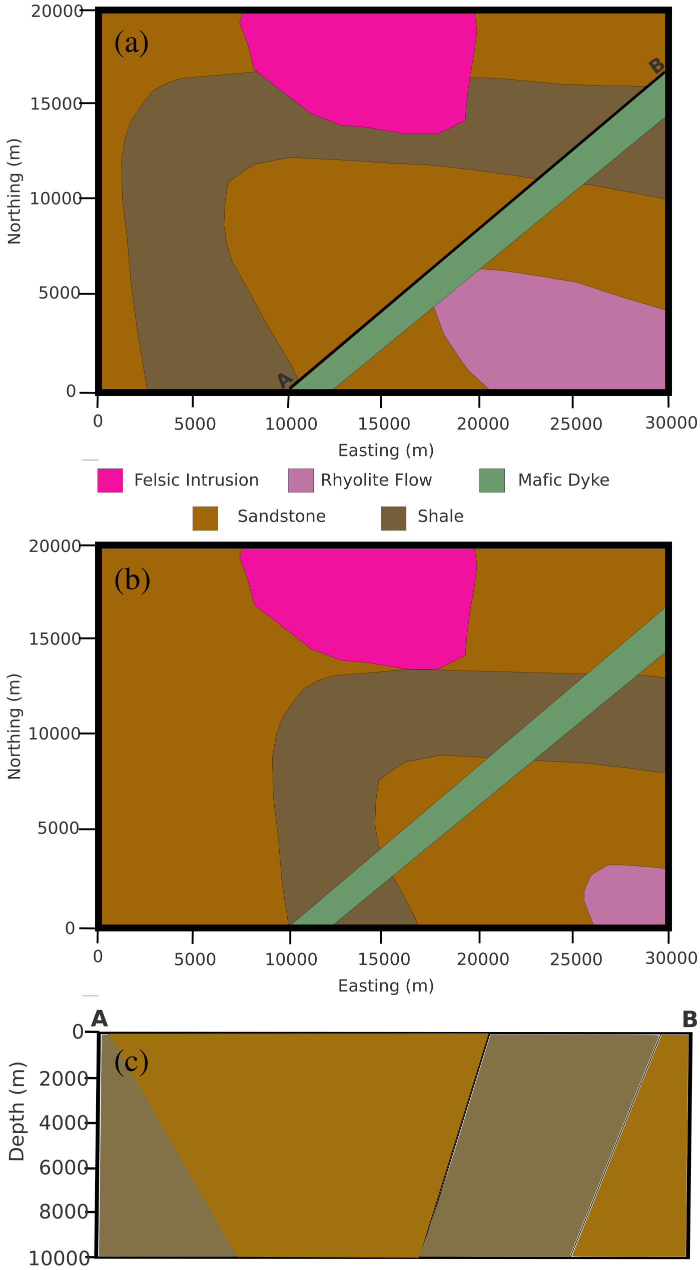

Figure 4: The (a) geological map, (b) horizontal depth section at -10 km and (c) vertical cross section used for our first model building example. The folded sedimentary layers (brown and tan units) and rhyolite flow (pale pink) have been cross-cut by a mafic dike (green unit) and a granitic pluton (hot pink unit) that were emplaced after the layers were folded. The cross section in (c) runs roughly southwest-northeast along the northwest side of the dike, as indicated by the black line labelled A–B in panel (a).

rock units. Geological maps may also contain topography contours that can be digitized.

Each section image must be loaded and calibrated such that image pixel coordinates can be transformed to spatial coordinates. To perform the calibration, the user is asked to click on two points on the image and provide 3D spatial coordinates for those points.

Before the model components (nodes and facets) can be created, groups must be defined to which the model components will be assigned. In this example, we define eight different groups, listed in Table 1. We have chosen drawing colours for these groups such that the nodes and facets created are set apart from the colours in the images.

3.2. Defining nodes

There are several cursor interaction modes, or "click modes", available for interacting with the 2D viewing panel. These click modes allow the user to perform various model building tasks by clicking or dragging with the mouse on the 2D viewing panel. Defining nodes on the current section is as simple as clicking on the 2D viewing panel. Each click defines a new node at the location of the cursor. These nodes are stored in memory as image pixel coordinates and are tied to the plane of the current section.

Fig. 5 shows the FacetModeller GUI after loading and calibrating the section images, defining groups, and adding nodes on each of the sections. Because this 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. The 2D viewing panel displays the currently selected section image and any nodes or facets selected for display. All objects specified for drawing are projected onto the plane of the current section. The 2D viewing panel is where users define nodes and facets via cursor interactions. The 3D viewing panel is only used for viewing: no model building

Table 1: Group definitions for our first model building example.

Name	Drawing colour	Description
pluton	green	nodes and facets around the pluton
dike_NW	red	nodes and facets on the north-west side of the dike
dike_SE	pink	nodes and facets on the south-east side of the dike
shale-mudstone	yellow	nodes and facets on the contact between mudstone and shale
sandstone-shale_NW	blue	nodes and facets on the north-west contact between sandstone and shale
sandstone-shale_SE	orange	nodes and facets on the south-east contact between sandstone and shale
boundary	white	nodes and facets on the boundary of the model volume
intersection	black	used to indicate some nodes that lie on particular intersections between surfaces

occurs directly through user interaction with the 3D viewing panel.

In Fig. 5, the geological map is displayed in the 2D viewing panel. In the 3D viewing panel, the outlines of the images of the three sections are drawn with black lines to help reference the images in 3D. The different coloured nodes correspond to the different groups they belong to. Alternatively, the nodes can be coloured based on the sections they are attached to. In the 2D panel, the nodes on the geological map section are drawn as filled circles and the nodes on the other sections are drawn as empty circles. In the 3D viewing panel, all nodes are drawn as filled circles. We have added nodes fairly coarsely, given this example is for demonstration purposes.

3.3. Defining facets

Defining facets is a challenging part of the model building process and much of the development process for FacetModeller has been concerned with this task. For this example, we begin by defining facets around the pluton. We use the selection boxes on the left of the GUI to display the geological map in the 2D viewing panel and we select only the "pluton" node group for drawing. Again, in the 2D panel, the nodes on the geological map section are drawn as filled circles and the nodes on the other sections are drawn as empty circles. Fig. 6 shows a view halfway through defining the facets around the pluton.

Sometimes, the projection used in the 2D viewing panel leads to difficulties when defining facets between multiple sections because the nodes on those sections may be drawn close to each other (for example, see Fig. 5). To help in this situation, the 3-by-3 grid of buttons on the bottom left of the tool panel can be used to shift the drawing location of the nodes on the selected "Other Sections" (drawn as empty circles) in relation to those in the "Current Section" (drawn as filled circles). In Fig. 6 we have shifted the "Other Section" nodes downwards in the 2D viewing panel.

Facets in a valid PLC must be planar and the simplest method for ensuring that a facet is planar is to make it a triangle. There is a click mode for defining triangular facets. As the cursor is moved around in the 2D viewing panel, candidate facets are displayed with white dashed outlines in both the 2D and 3D viewing panels, as shown in Fig. 6. FacetModeller automatically determines all the possible candidate triangular facet definitions within some tuneable neighbourhood around the cursor and displays the facet with centroid closest to the cursor. Often, it is possible to build the model with different triangular tessellations with different quality characteristics; to aid in the generation of quality PLCs, FacetModeller displays the smallest vertex angle of the candidate facet in a bar above the 2D viewing panels (see Fig. 6). Once the desired candidate facet is found, the user generates that facet by clicking the mouse. The new facet is then added to the model and drawn using the colour specified for its group.

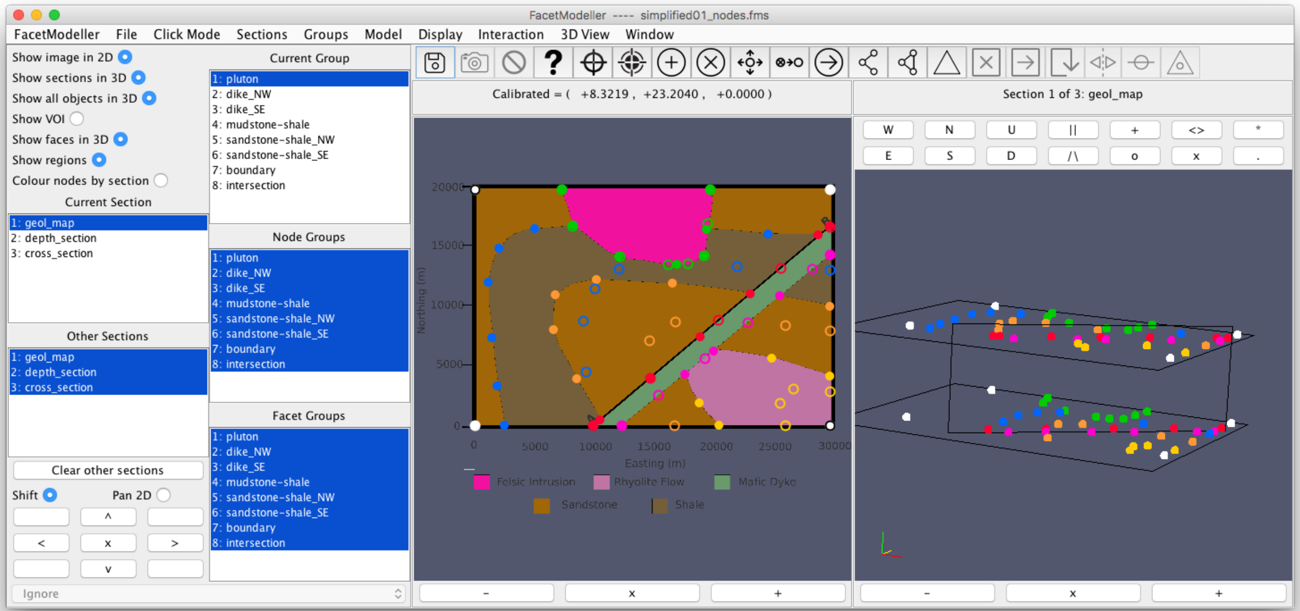

Figure 5: Example 1: the three section images have been loaded into FacetModeller and calibrated, several groups have been defined, and nodes have been created on the two depth sections. The 2D viewing panel shows an overhead view of the geological map. The view in the 3D viewing panel is roughly from the south-east, elevated slightly.

In the 2D viewing panel, facets are drawn as transparent coloured patches with black edges: this allows all facets to remain visible regardless of the projection used for the 2D viewing panel. In the 3D viewing panel, facets are drawn non-transparent so that facets closer to the viewing camera obscure those behind them: this provides a more realistic and intuitive 3D viewing experience but can require users to rotate the 3D view such that the new facets being defined are visible. Having these two different drawing choices (transparent vs non-transparent facets), and having two different but simultaneously available views, are both helpful for many manual model building tasks because they improve visualization of the model being built.

3.4. Dealing with surface intersections

In Fig. 7 we have finished defining facets around the pluton and we have defined some facets lying on the north-western-most sandstone-shale contact, named "sandstone-shale_NW". For illustrative purposes, we have done this in a purposefully naive manner, the result being that some facets intersect each other, thus not honouring the requirements of a valid PLC. To fix these intersections, we could redefine the facets that make up the pluton surface, for example as seen in Fig. 8. However, this alters the shape of the pluton, which may not be preferable: the facets for the pluton are all oriented vertically in Fig. 7 but are no longer strictly vertical in Fig. 8.

An alternative approach is to insert new nodes along the intersection of these two surfaces. To help with this task, FacetModeller provides the ability to change the projection of the 2D viewing panel to that of the current 3D view: see Fig. 9. With the new view to help us, we begin with the model in Fig. 7 and use the "Add Node in Facet" click mode to add some new nodes in the planes of the currently existing facets. The use of that click mode is demonstrated in Fig. 7.

We add these new nodes to the "intersection" group and draw them black. These new nodes are not attached to any of the section images, that is, they do not strictly lie on any of the

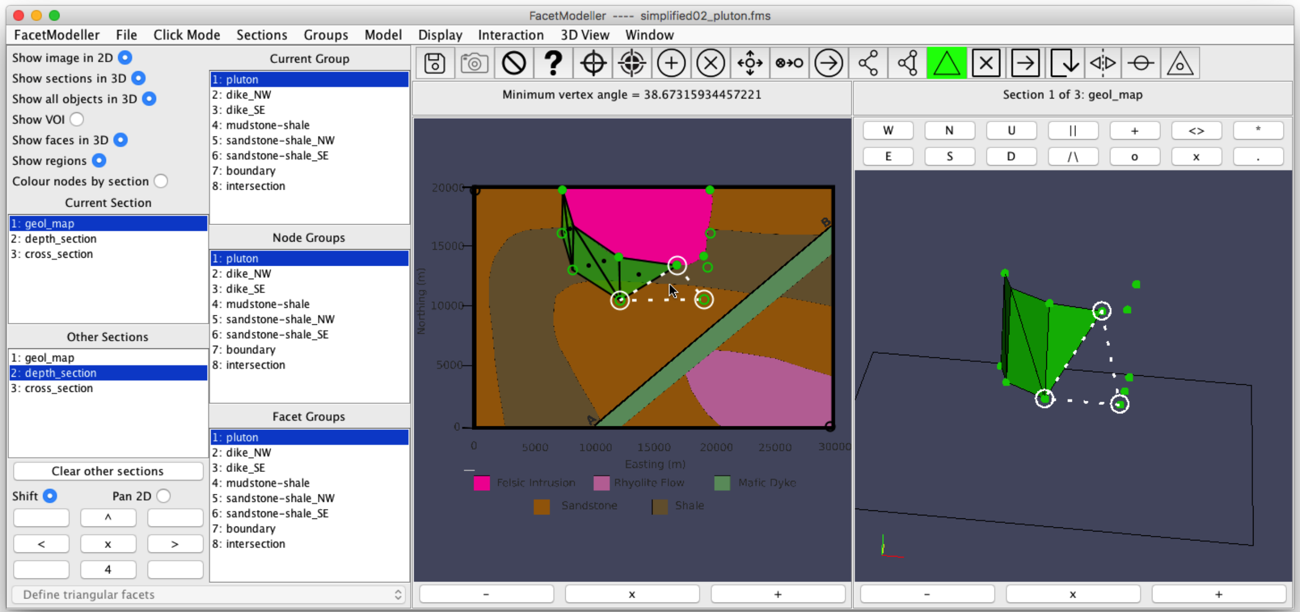

Figure 6: Example 1: triangular facets are being defined. Five facets have been defined, drawn as green triangles with black edges in both the 2D and 3D viewing panels, and with small black dots at their centres in the 2D viewing panel only (which are handles that can be used later for deleting the facets). The current candidate facet is drawn with white dotted lines along the facet's edges and large unfilled white circles around the facet's nodes. The location of the cursor is indicated on the 2D viewing panel. The "Other Section" nodes (drawn as unfilled circles) have been shifted downwards in the 2D viewing panel compared to their location in Fig. 5. The view in the 3D viewing panel is roughly from the south, elevated slightly.

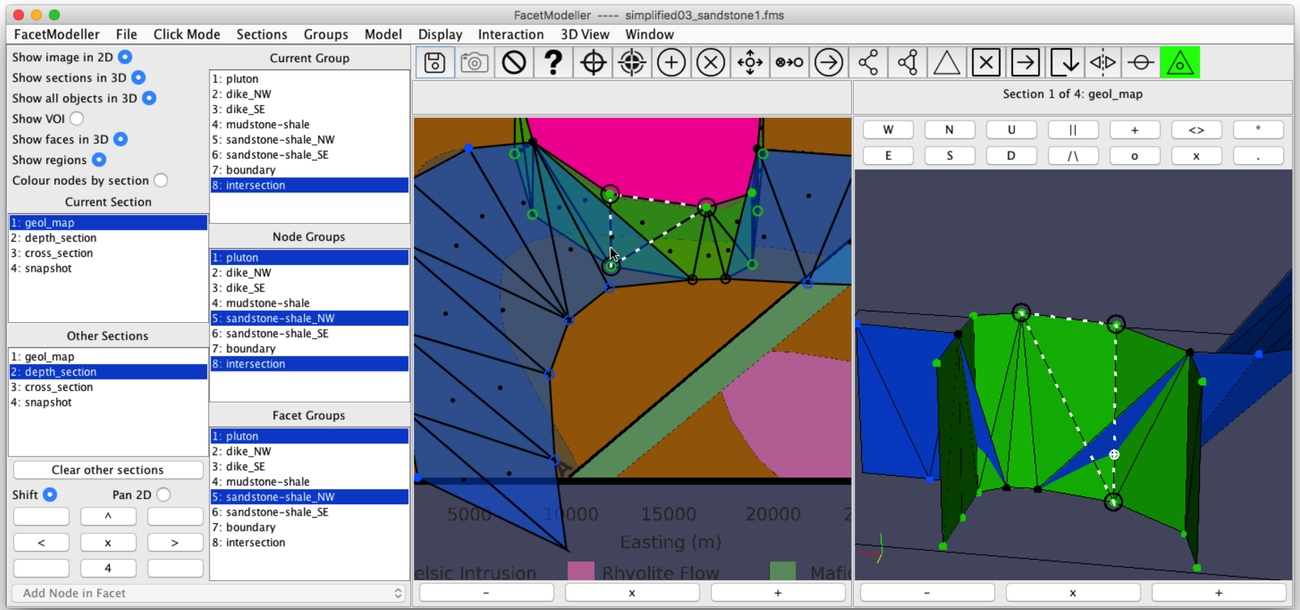

Figure 7: Example 1: preliminary triangular facets have been defined around the pluton (facets drawn green) and on the north-western-most sandstone-shale contact (facets drawn blue). Four nodes have been moved into the group named “intersection” and are drawn as filled black circles to help indicate the intersection of these two surfaces. Overlays have been added to demonstrate the use of the “Add Node in Facet” click mode: the currently selected facet, in which a new node can be added, is drawn with white dotted lines along the facet’s edges and large unfilled black circles are drawn around the facet’s nodes; a candidate node, on the plane of the current working facet, is drawn as a small white circle and cross in the 3D viewing panel, corresponding to the location of the cursor in the 2D viewing panel. The 2D viewing panel has been zoomed compared to as seen in Fig. 6. The view in the 3D viewing panel is roughly from the north, elevated slightly.

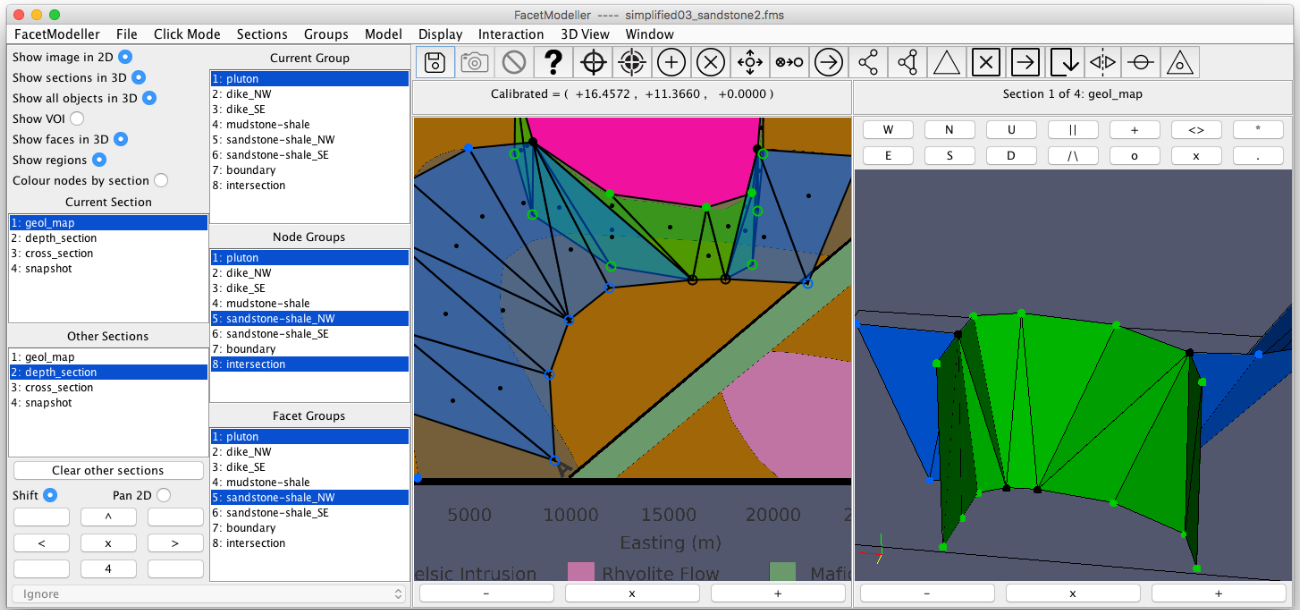

Figure 8: Example 1: an alternative tessellation of the pluton surface (facets drawn green) that represents a significantly different geometry for the pluton compared to Fig. 7.

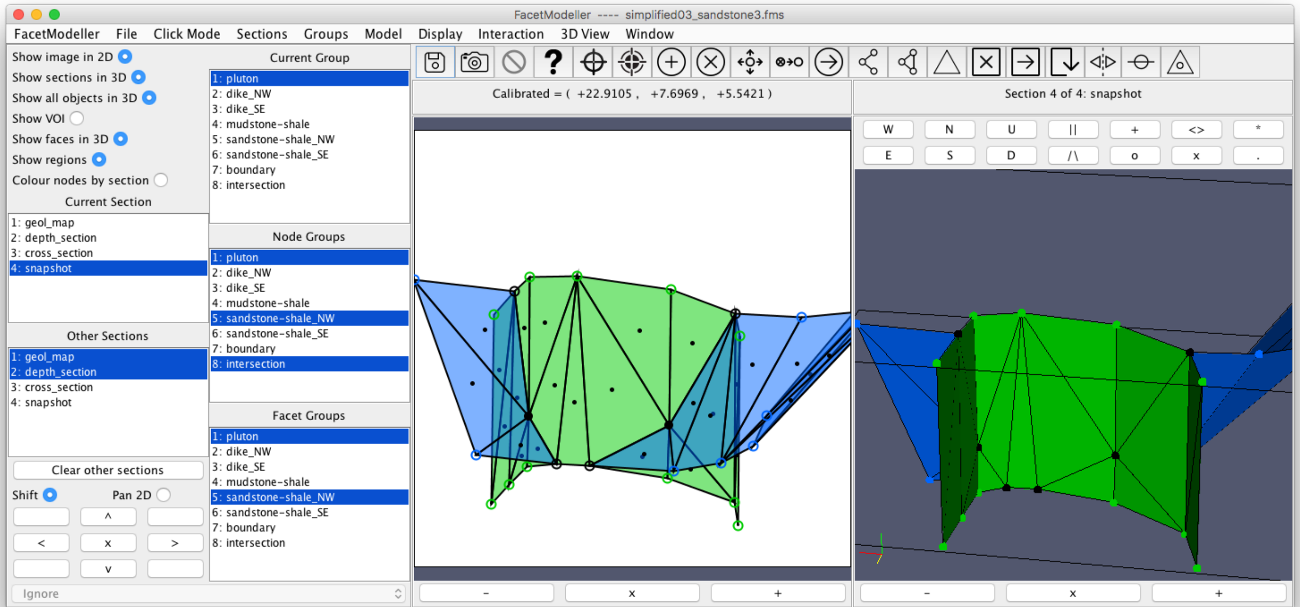

Figure 9: Example 1: the problematic intersections in Fig. 7 have been fixed by adding two new nodes (black, on the interior of the pluton surface) without having to change the geometry of the pluton (facets drawn green).

2D planes defined by those sections. Hence, they are stored in memory as 3D points in spatial units. We then delete the offending facets and define new facets to sew the two surfaces together properly, as seen in Fig. 9. Essentially, we have cut off the parts of the sandstone-shale surface that were poking through the pluton surface.

The remainder of the model building process for this example follows the steps above for the other surfaces. Fig. 10 shows the final model with all nodes and facets defined.

4. Model building example 2: a porphyry copper system with many intersecting surfaces

In this example we build a model of the core of a porphyry copper system including different geological phases. Several surfaces were built in another program, Gocad by Paradigm (see Mallet, 1992), from outcrop mapping and geological inference on the dip and depth extension. Those surfaces make contact with each other along their boundaries but they have been built independently and they are not sewn together following the requirements of a valid PLC. Furthermore, the original surfaces from Gocad were of insufficient quality for generating quality meshes for numerical modelling work: for example, see Fig. 11. While there are tools in Gocad to sew model surfaces together, the procedure generally results in a further deterioration of the tessellation quality.

4.1. Initial steps

Considering the poor quality of the original surfaces (Fig. 11), we used the Autodesk Meshmixer program (Autodesk, 2017a) to re-mesh these surfaces, that is, re-generate surface tessellations of improved quality that still honour the spatial boundaries of the original surfaces and are not significantly different from the originals. Explaining the process of using Meshmixer for this purpose is beyond the scope of this paper and we refer interested readers to the Meshmixer documentation (Autodesk, 2017b). Fig. 12 shows the same surfaces in Fig. 11 but after re-meshing using Meshmixer.

In this example we work with the second type of data input mentioned previously: pre-defined tessellated surfaces. The complication here is that all nodes must be associated with a particular section but there is no image to be digitized. A section without image must be defined, which can be a rectangle of any chosen size and orientation, and the nodes and facets are loaded into that section. Those nodes are stored in memory as 3D points in spatial units and they do not necessarily lie on the 2D plane defined by the section. We have the choice here of loading the surfaces into different sections or different groups, or both. Here we load them into the same section, named "all", which is used as a container for information rather than a real spatial object. We load each surface into different groups so we can distinguish the different surfaces by group colour. We are then able to select only a subset of the surfaces required for any particular task, for example two intersecting surfaces that must be sewn together, as performed below.

Fig. 12 shows the FacetModeller GUI after performing all the initial steps mentioned above and loading only two of the several surfaces that must be sewn together to create the final model. Below we demonstrate the process of sewing those two surfaces together. We then show the final model where all intersecting surfaces have been sewn together into a valid PLC.

Careful investigation of the 3D viewing panel in Fig. 12 indicates two problems with the surfaces. First, there are still a few poor quality triangular facets that we were unable to remove using MeshMixer. Those facets would lead to poor quality cells in a tetrahedral mesh. Second, the two surfaces do not intersect at shared facet edges, thus breaking the requirements of a valid PLC.

4.2. Improving surface quality

There are several possible manual approaches available in FacetModeller to fix the first issue but the one we take here is to pinch out the offending facets. We use the "Merge Nodes" click mode, where the user must click on two nodes: for all facets that contain the first node, the first node is replaced with the second node, and the first node is then removed from the model

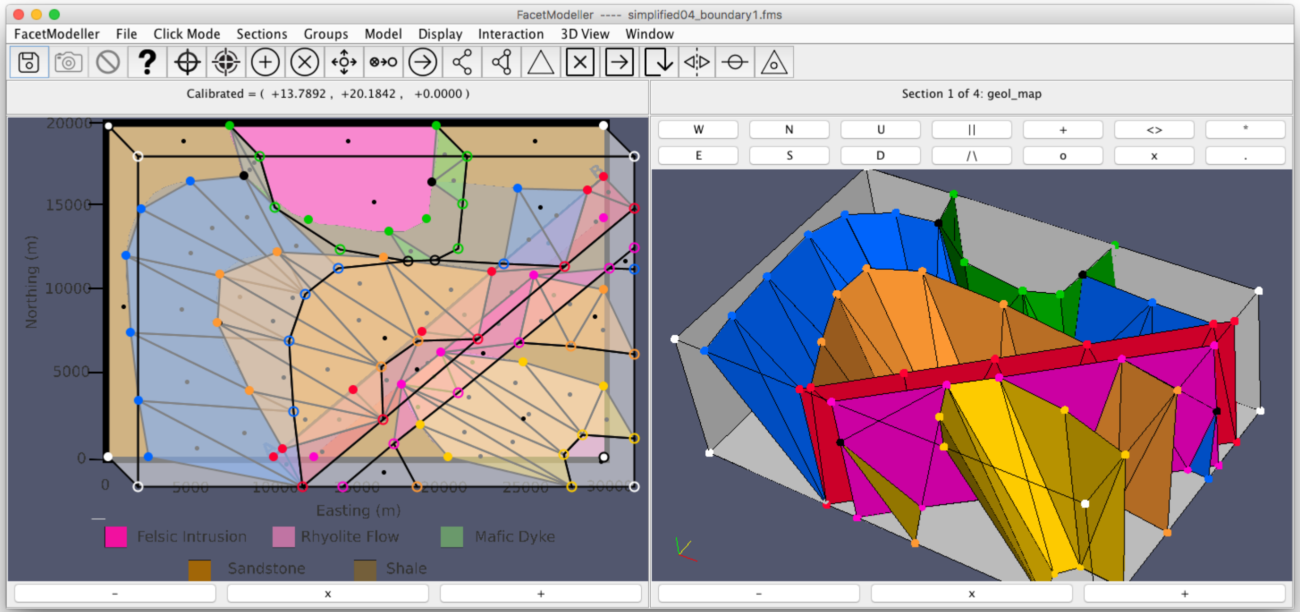

Figure 10: Example 1: the final model with all nodes and facets defined. Some of the boundary facets (light grey) are not shown in the 3D viewing panel because they would obscure the other surfaces. The tool panel, usually on the left of the FacetModeller GUI, has been hidden to increase the size of the model viewing panels. The nodes on the depth section (drawn as unfilled circles) have been shifted to the south-east in the 2D viewing panel compared to their location in Fig. 5. The 3D viewing panel shows an elevated view roughly from the south-east.

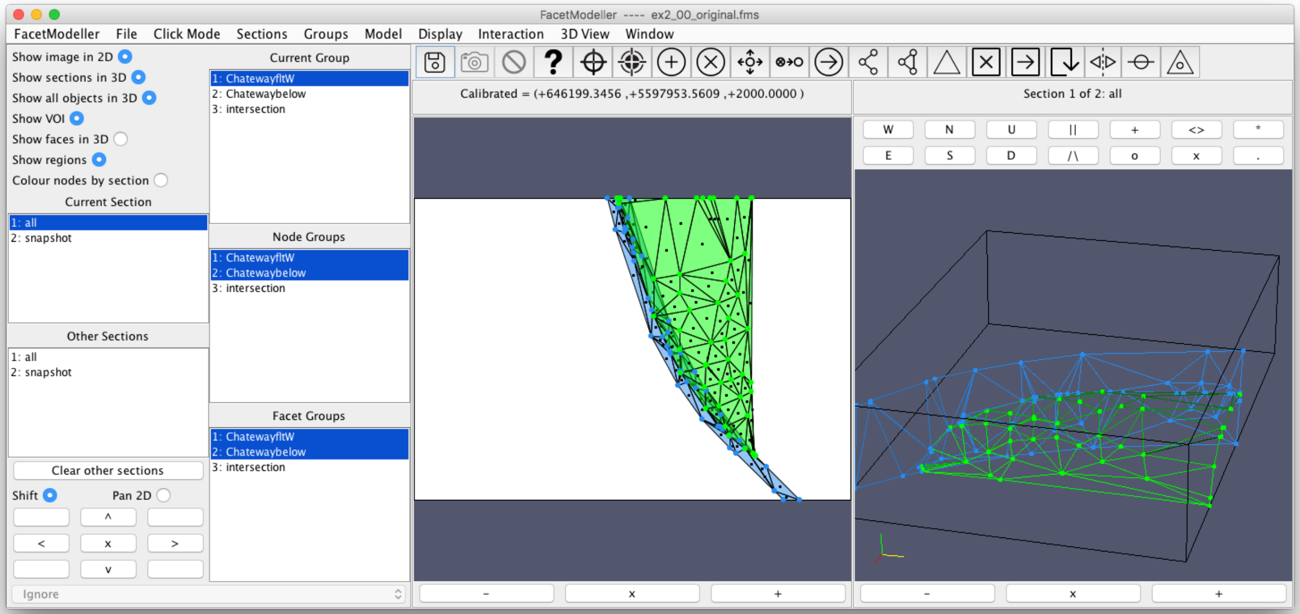

Figure 11: Example 2: two surfaces (blue and green) have been loaded into FacetModeller, each into a different group. The 2D viewing panel shows an overhead view (north up the page). The 3D viewing panel shows an elevated view roughly from the north-east. In the 3D viewing panel, the nodes and the edges of the surface facets are drawn in their respective colours (instead of the alternative drawing option of solid facet patches with black edges that was used in the figures for Example 1.)

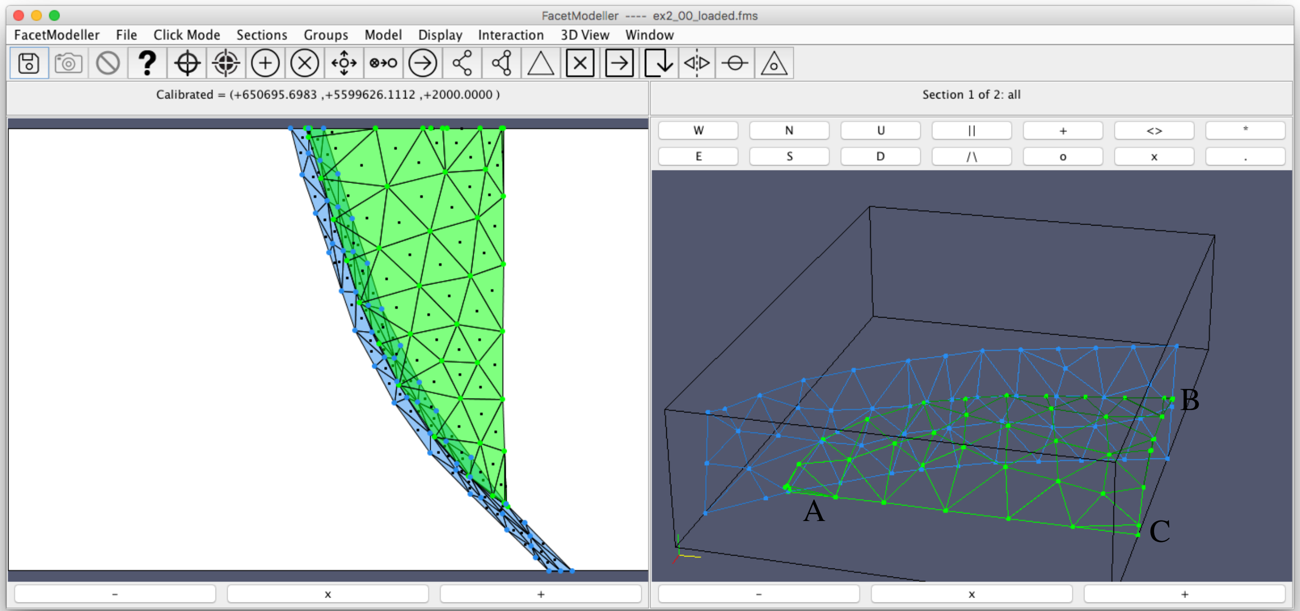

Figure 12: Example 2: as in Fig. 11 but after re-meshing the surfaces using Autodesk Meshmixer. Labels “A”, “B” and “C” indicate locations of poor quality triangles in the tessellated green surface. The tool panel has been hidden in the FacetModeller GUI here and in the figures that follow.

entirely. When the two nodes are connected by a facet edge, the result is the desired pinching out of one or more facets containing that edge element. After this merging, the pinched-out facets will have zero-area and must be deleted from the model, which can be done manually or automatically using FacetModeller's model cleaning tools. Fig. 13 shows the result of performing the above procedure on the offending facets.

FacetModeller provides other options for manually improving the quality of surfaces. As already demonstrated in our first model building example, new nodes could be added on a surface and then integrated into the surface by altering the tessellation. Alternatively, existing nodes can be moved. Edge-flip operations can also be performed, similar to those involved when creating Delaunay triangulations (e.g. see Shewchuk, 1996). All of those operations may alter the shape of the surface away from the original. If that is a concern, two versions of the surfaces can be kept loaded in FacetModeller: the original surface that remains unaltered and another version that will be altered. Users can then visually reference their alterations against the original versions of the surfaces, which can be hidden or deleted once they are no longer required.

4.3. Sewing the surfaces together

We must now sew the two surfaces together along their intersection such that the requirements of a valid PLC are honoured. To aid us here, we change the group association of the nodes on the intersecting boundary of the green surface. We associate them with a new group named "intersection". The nodes in that group are drawn red: see Fig. 13.

We now use the "Merge Nodes" click mode functionality along the intersection to replace nodes on the blue surface with the new red nodes along the intersection. This effectively sews the surfaces together: see Fig. 14. This approach is really only sensible when the nodes on the two surfaces, on either side of the intersection, are close together such that the geometry of the surfaces does not change significant. Hence, we have not done this along the entire length of the intersection and we will demonstrate an alternate approach on the remainder.

There is a small length of the intersection that remains to be sewn: see the bottom left of the 3D viewing panel in Fig. 14, to which we zoom into in Fig. 15. As indicated by the annotations in Fig. 15, we divide that length of un-sewn intersection into two parts. For the part near the "D" label, the green surface has a node (red, to the right of the "D" label) that does not have a neighbouring node on the blue surface, so the node merging approach used previously is not adequate. To sew this part, we first delete all the facets in the blue surface that lie around the "D" node. This results in a void in the blue surface, which we fill by defining new facets in such a way that they use the required "D" node.

For the other part of the intersection, near the "E" label in Fig. 15, we have the opposite scenario: the blue surface has a node (blue, to the left of the "E" label) that does not have a neighbouring node on the green surface. Here, we could follow the same procedure as above. Instead, we delete the blue "E" node, which automatically deletes the facets containing that node, and we define new facets to sew up the resulting void.

Fig. 16 shows the voids left after deleting the existing facets and node. Fig. 17 shows the surfaces after new facets are defined to close the voids and sew up the intersection. Note that care must be taken when sewing the voids together such that the facets defined have edges that connect the nodes along the intersection.

There are a total of seventeen tessellated surfaces for this model, including the topography surface. They all must be cleaned and sewn following the steps above. The result is in Fig. 18, which represents a valid PLC for providing to a meshing program like TetGen. Fig. 1 shows a mesh generated from that PLC using TetGen.

5. Other functionality, applications and possible extensions

The examples above have demonstrated much of the functionality of FacetModeller but here we provide readers with a more complete picture. First, although this paper has mentioned 3D models exclusively thus far, FacetModeller can also be used for building 2D models. For 2D models, there is only a single

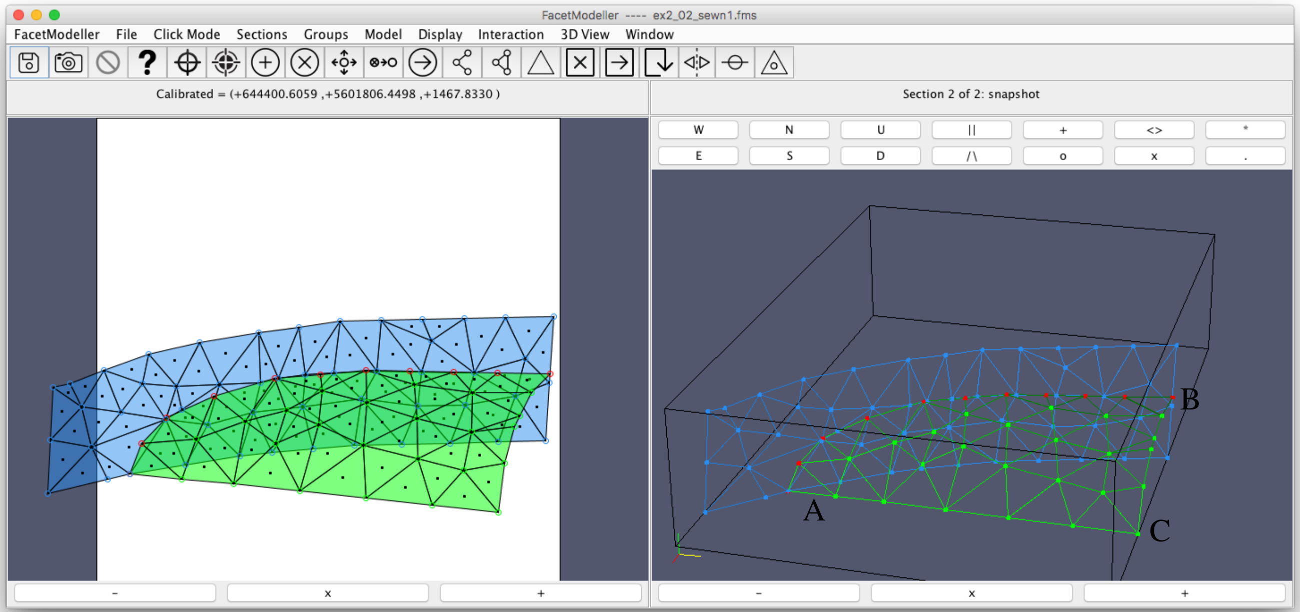

Figure 13: Example 2: as in Fig. 12 but some poor quality triangles in the green surface have been removed. Where the two surfaces intersect, the nodes on the green surface have been moved to a new group and drawn red (note that one red node is barely visible in the 3D viewing panel because it lies slightly behind the blue surface). Labels “A”, “B” and “C” indicate the same locations from Fig. 12.

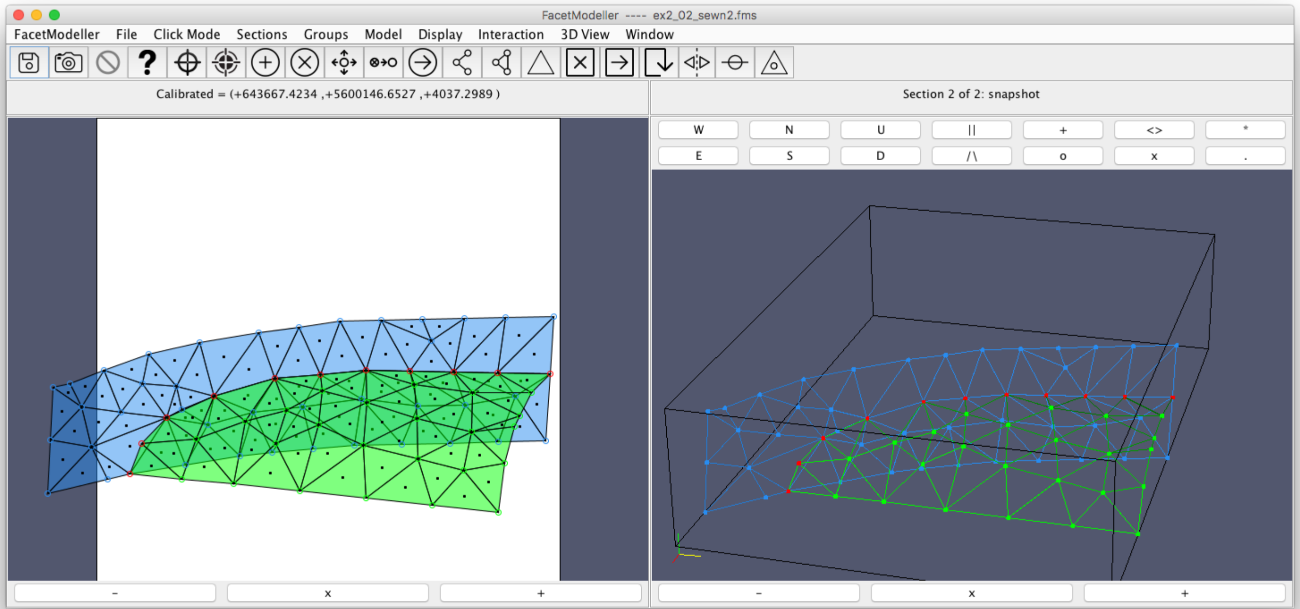

Figure 14: Example 2: the surfaces have been sewn together most of the way along their intersection.

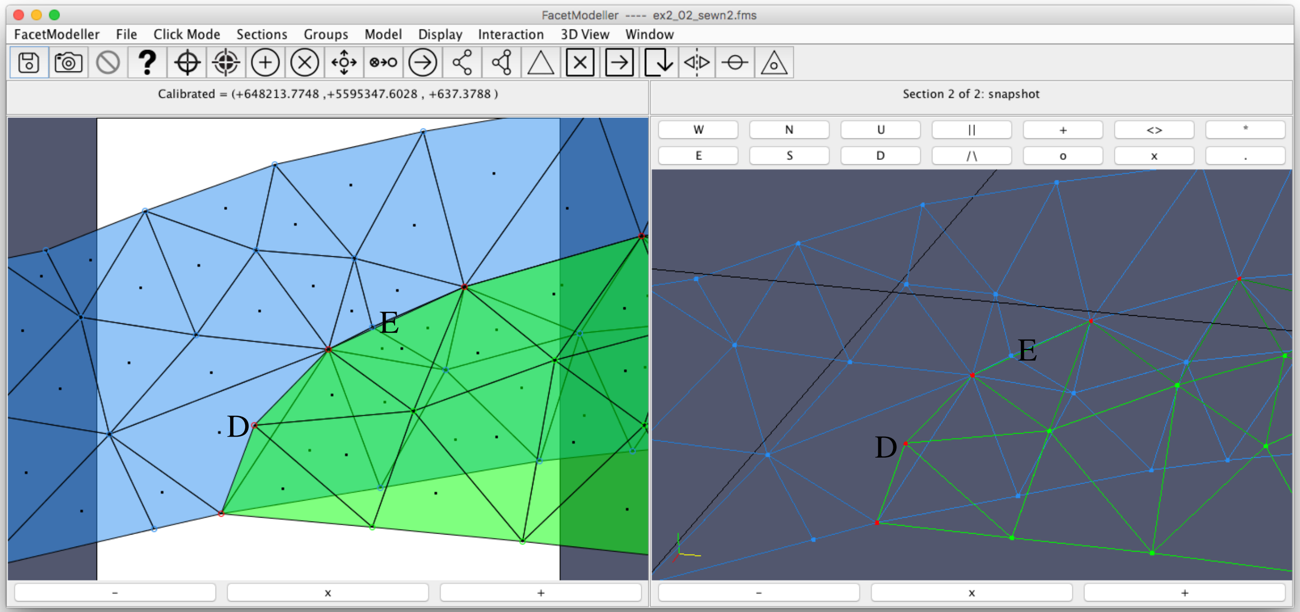

Figure 15: Example 2: as in Fig. 14 but with the viewing panels zoomed into a section of the model where the two surfaces must still be sewn along their intersection. Labels “D” and “E” indicate nodes in locations where the two surfaces must still be sewn together.

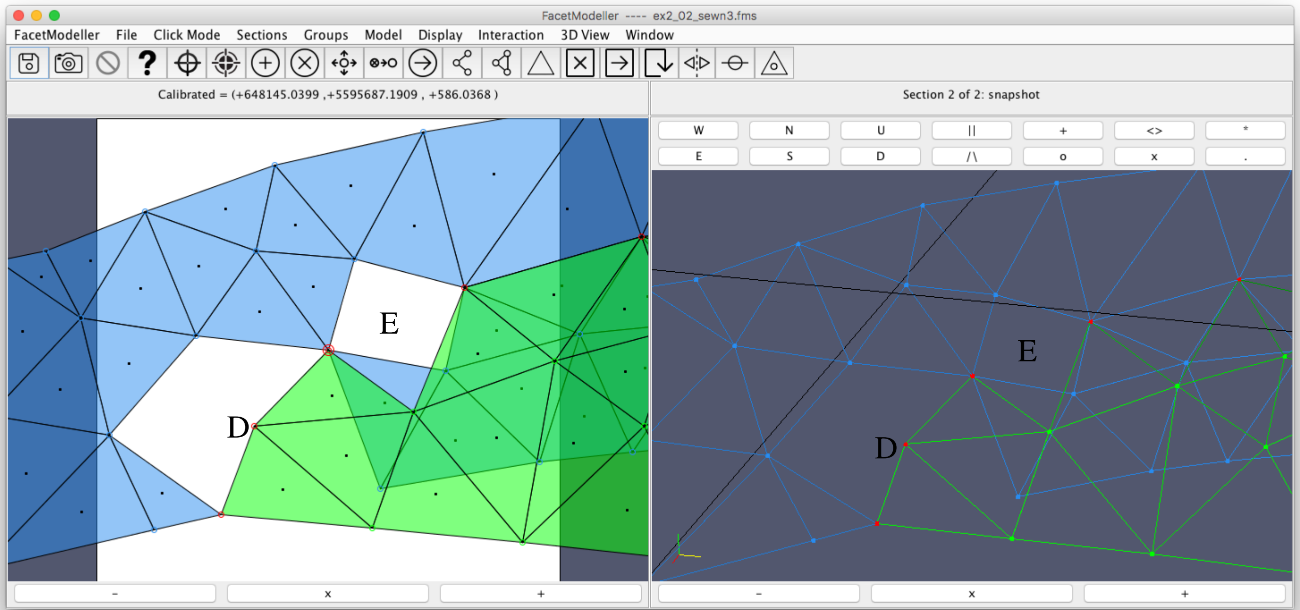

Figure 16: Example 2: as in Fig. 15 but with several facets and one node deleted, leaving voids in the surfaces. Labels “D” and “E” indicate the same locations from Fig. 15.

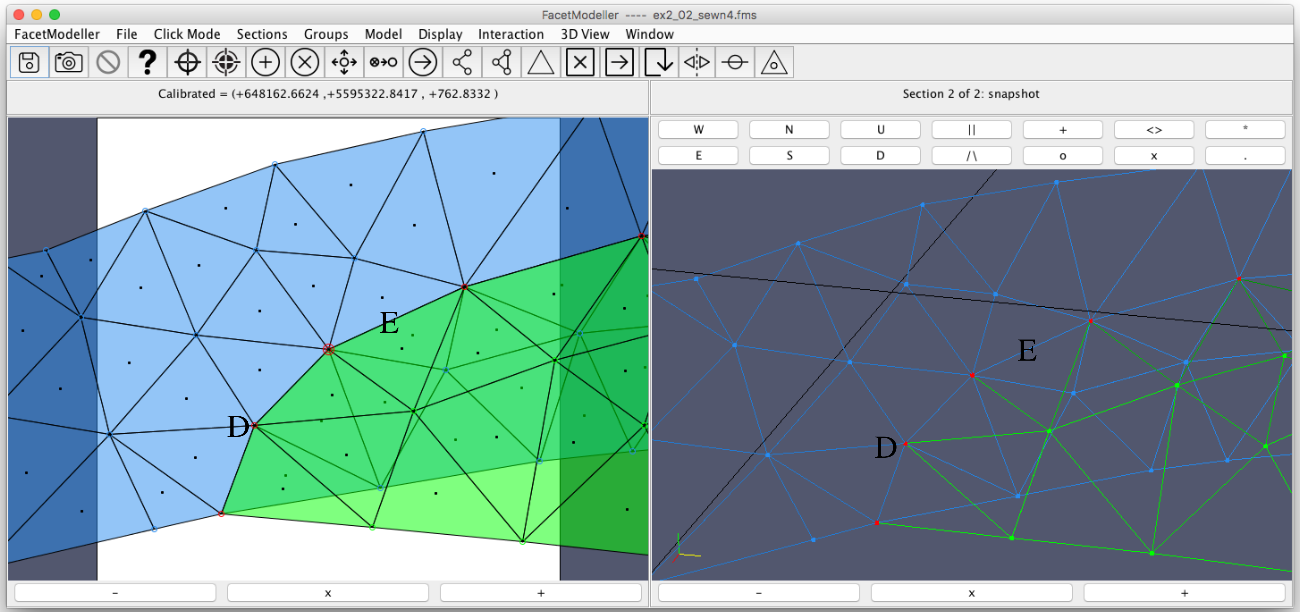

Figure 17: Example 2: as in Fig. 16 but with the voids now filled appropriately such that the two surfaces are completely sewn together at their intersection. Labels "D" and "E" indicate the same locations from Fig. 15 and Fig. 16.

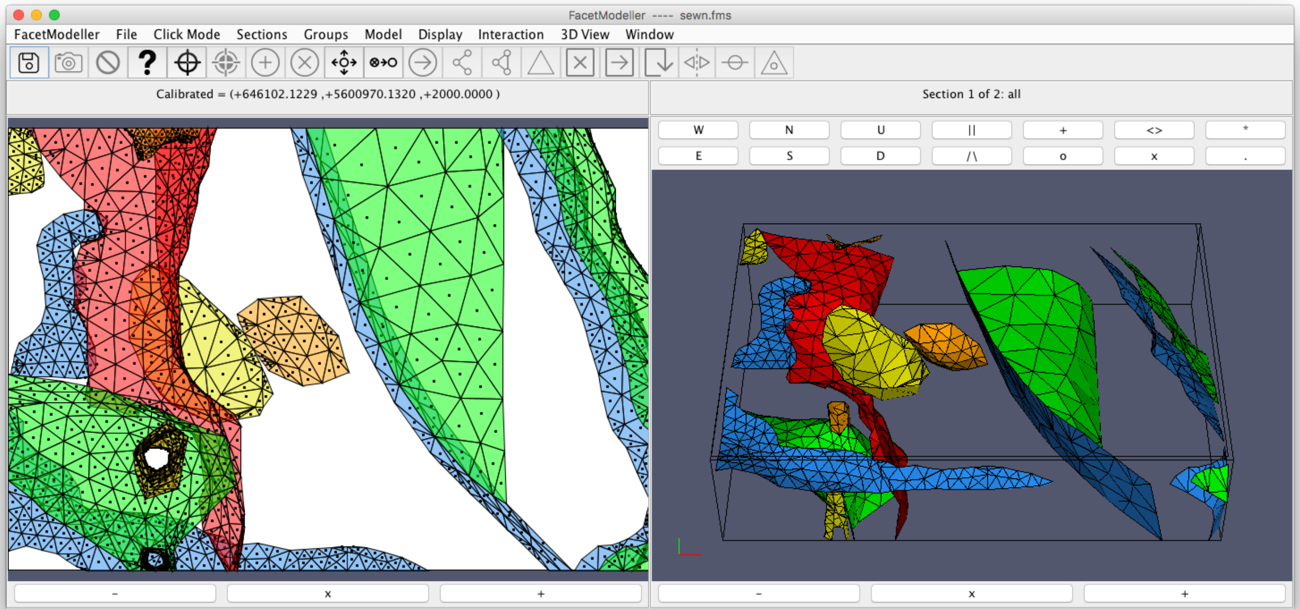

Figure 18: Example 2: the final model with all nodes and facets defined. Again, the facets around the boundary of the VOI are not shown here because they would obscure the other surfaces in the 3D view. Similarly, the topography surface is not shown. The 3D viewing panel shows an elevated view from the south.

section and facets are no longer triangles but line-segment elements that connect pairs of nodes. There is a reduced tool set required for building 2D models and hence the 2D version of the GUI is somewhat simplified from that seen in the figures above.

We have mentioned some of the click modes currently available in FacetModeller but below we provide a complete list of the tasks that users are able to perform:

- set the origin of the 2D and 3D viewing panels
- spatially register a section image
- obtain information about the different model components
- add nodes at specific locations on spatially calibrated images
- add nodes on a facet edge or anywhere on the plane of a facet
- delete, move and merge nodes
- define triangular facets, or polygonal facets with more than three nodes
- delete facets
- reverse the node order in facets, which is helpful for numerical modelling methods that might provide different results depending on the order (e.g. Okabe, 1979)
- change the group and sections associated with different nodes or facets
- perform edge-flip operations.

Some of those tasks require a single click while others require multiple clicks, and some support click-and-drag interaction. For some tasks, temporary overlays are drawn in the 2D or 3D viewing panels, for example to indicate facets currently being worked with (as in Fig. 6 and Fig. 9) or a candidate node about to be defined (as in Fig. 9). The drawing colours for those overlays, and for any model components, can be changed as desired by the user.

FacetModeller allows users to obtain various information about the model, via dialogs or through mouse interaction, including:

- the number of nodes and facets for the entire model, for each group and for each section
- the 3D spatial coordinates for a specific node
- the number of facets associated with a specific node
- the node indices associated with a specific facet
- the order of the nodes in a specific facet
- the minimum vertex angle in a specific facet.

FacetModeller also allows users to highlight specific nodes and facets by their index (order listed in memory), which can be helpful for cross-referencing against lines in imported or exported files.

FacetModeller's model cleaning tools provide users the ability to identify and delete the following:

- facets with zero area
- facets with duplicated nodes
- facets with fewer than three unique nodes
- non-planar facets
- duplicate facets
- duplicate nodes.

The current version of FacetModeller does not detect intersecting facets that break the requirements of a valid PLC – see Fig. 2(a). Instead, we rely on TetGen for that task, which identifies the indices of the offending facets such that they can be easily identified in FacetModeller.

The model can be saved in various ASCII formats:

- .node, .ele and .poly files (refer to the TetGen documentation for those file formats)
- .vtu files for visualization, for example in ParaView (Kitware, Inc. and Los Alamos National Laboratory, 2017)
- "FacetModeller session" files that enable users to save their work and restart their model building session at later times.

Although we have designed FacetModeller for building geological surface-based models, the software may be helpful in other fields. For example, Lelièvre and Grey (2017) developed a software package they named JMorph, for performing morphometric measurements on digital images of fossil assemblages. JMorph makes measurements in what is essentially a 2D environment, the 2D data being a specific digital image. 3D data for morphometric applications might correspond to point clouds from laser scanning or stacked section images from computerized tomography (CT) scanning. That information could easily be imported into FacetModeller, which could thereby act as a 3D extension to JMorph, provided that new click modes were developed for the required 3D morphometric measurements.

6. Conclusion

We have developed a new software tool, FacetModeller, that is custom-built for efficient manual creation, manipulation and analysis of 3D surface-based models. FacetModeller's three major assets are:

1. the speed in which manual model building tasks can be performed
2. functionality that helps users honour the requirements of a valid PLC, and improve the quality of their PLC
3. providing two simultaneous views of the model being built, each view providing different but complimentary information to the user.

These assets are important when manual model building is the only viable option, for example when automatic methods provide inappropriate results such as poor quality tessellations or intersecting facets.

FacetModeller is written in Java and will therefore run on any operating system that can run Java, including Linux, Mac and Windows. FacetModeller is freely available for academic, non-commercial use as source code or a compiled JAR file packaged with required libraries. The code has been developed with extensibility in mind: developers can define new click modes using the ClickTask interface (a Java interface). Other data processing tools could also be added to FacetModeller through its modular design, which follows the model–view–controller software architecture pattern.

FacetModeller has been used in several studies in which complicated geological models were generated for numerical modelling work (Lelièvre and Farquharson, 2013; Carter-McAuslan et al., 2015; Darijani and Farquharson, 2016; Dunham et al., 2016; Jahandari and Farquharson, 2016; Jones et al., 2016; Nalepa et al., 2016). Without FacetModeller, those studies would not have been possible. The program should be of use to anyone wishing to build 3D models for geophysical numerical modelling work on unstructured meshes. FacetModeller should also be of interest to those working on rectilinear meshes: more complicated 3D models can be built in FacetModeller using an unstructured surface-based representation before interpolating the model onto a rectilinear mesh to generate a pixellated approximation. We hope that FacetModeller will be of use to other researchers undertaking 3D numerical modelling studies, in geophysics or other fields.

Ansari, S., Farquharson, C. G., 2014. 3d finite-element forward modeling of electromagnetic data using vector and scalar potentials and unstructured grids. Geophysics 79, E149–E165, doi:10.1190/geo2013–0172.1.

Autodesk, 2017a. Autodesk Meshmixer, http://www.meshmixer.com, last accessed November 2017.

Autodesk, 2017b. Meshmixer Manual, https://www.mmmanual.com, last accessed November 2017.

Cai, H., Hu, X., Li, J., Endo, M., Xiong, B., 2017. Parallelized 3d csem modeling using edge-based finite element with total field formulation and unstructured mesh. Computers and Geosciences 99, 125–134.

Carter-McAuslan, A., Lelièvre, P. G., Farquharson, C. G., 2015. A study of fuzzy c-means coupling for joint inversion, using seismic tomography and gravity data test scenarios. Geophysics 80 (1), W1–W15, doi:10.1190/geo2014–0056.1.

Caumon, G., Collon-Drouaillet, P., Le Carlier de Veslud, C., Viseur, S., Sausse, J., 2009. Surface-based 3D modeling of geological structures. Mathematical Geosciences 41, 927–945.

Darijani, M., Farquharson, C. G., 16–21 October 2016. Synthetic modeling and joint inversion of gravity and seismic refraction data for overburden stripping in the athabasca basin, canada. In: SEG Technical Program Expanded Abstracts 2016. Dallas, pp. 5011–5016, doi:10.1190/segam2016–13947873.1.

Dunham, M., Ansari, S., Farquharson, C. G., 16–21 October 2016. Application of 3d marine csem finite-element forward modeling to hydrocarbon exploration in the flemish pass basin offshore newfoundland. In:

SEG Technical Program Expanded Abstracts 2016. Dallas, pp. 917–921, doi:10.1190/segam2016–13779957.1.

Farquharson, C. G., Ash, M. R., Miller, H. G., January 2008. Geologically constrained gravity inversion for the Voisey's Bay ovoid deposit. The Leading Edge 27 (1), 64–69.

Fullagar, P. K., Pears, G., Hutton, D., Thompson, A., 2004. 3D gravity and aeromagnetic inversion for MVT lead-zinc exploration at Pillara, Western Australia. Exploration Geophysics 35 (2), 142–146, doi:10.1071/EG04142.

GeoRessources, CREGU, 2017. RINGMesh: A free programming library for geological model meshes, http://www.ring-team.org/software/ringmesh, last accessed November 2017.

Guillen, A., Calcagno, P., Courrioux, G., Joly, A., Ledru, P., 2008. Geological modelling from field data and geological knowledge Part II. Modelling validation using gravity and magnetic data inversion. Physics of the Earth and Planetary Interiors 171, 158–169.

Jahandari, H., Farquharson, C., 2013. Forward modeling of gravity data using finite-volume and finite-element methods on unstructured grids. Geophysics 78, G69–G80, doi:10.1190/geo2012–0246.1.

Jahandari, H., Farquharson, C., 2014. A finite-volume solution to the geophysical electromagnetic forward problem using unstructured grids. Geophysics 79 (6), E287–E302, doi:10.1190/geo2013–0312.1.

Jahandari, H., Farquharson, C. G., 16–21 October 2016. 3d minimum-structure inversion of magnetotelluric data using the finite-element method and tetrahedral grids. In: SEG Technical Program Expanded Abstracts 2016. Dallas, pp. 998–1003, doi:10.1190/segam2016–13866304.1.

Jones, D., Ansari, S., Farquharson, C. G., 16–21 October 2016. Synthesizing time-domain electromagnetic data for graphitic fault zones and associated uranium deposits in the athabasca basin, canada. In: SEG Technical Program Expanded Abstracts 2016. Dallas, pp. 2206–2210, doi:10.1190/segam2016–13866371.1.

Kitware, Inc., Los Alamos National Laboratory, 2017. ParaView: an open-source, multi-platform data analysis and visualization application, https://www.paraview.org, last accessed November 2017.

Klingner, B. M., Shewchuk, J. R., 2017. Stellar: A tetrahedral mesh improvement program, https://people.eecs.berkeley.edu/ jrs/stellar/, last accessed November 2017.

Lelièvre, P., Carter-McAuslan, A., Farquharson, C., Hurich, C., March 2012. Unified geophysical and geological 3D Earth models. The Leading Edge 31 (3), 322–328.

Lelièvre, P. G., Farquharson, C. G., 2013. Gradient and smoothness regularization operators for geophysical inversion on unstructured meshes. Geophysical Journal International 195 (1), 330–341, doi:10.1093/gji/ggt255.

Lelièvre, P. G., Farquharson, C. G., Hurich, C. A., 2011. Computing first-arrival seismic traveltimes on unstructured 3D tetrahedral grids using the fast marching method. Geophysical Journal International 184 (2), 885–896.

Lelièvre, P. G., Grey, M., 2017. Jmorph: Software for performing rapid morphometric measurements on digital images of fossil assemblages. Computers and Geosciences 105, 120–128.

Los Alamos National Laboratory, 2017. LaGriT: Los Alamos Grid Toolbox, http://lagrit.lanl.gov, last accessed November 2017.

Mallet, J. L., 1992. Gocad: A computer aided design program for geological applications. In: Turner, A. K. (Ed.), Three-Dimensional Modeling with Geoscientific Information Systems. Vol. 354 of NATO ASI Series (Series C: Mathematical and Physical Sciences). Springer, Dordrecht, pp. 123–141.

McGaughey, J., 2006. The common earth model: A revolution in mineral exploration data integration. In: Harris, J. (Ed.), GIS For the Earth Sciences. Vol. 44. Geological Association of Canada Special Publication, Ch. 25, pp. 567–576.

McGaughey, J., 2007. Geological models, rock properties, and the 3d inversion of geophysical data. In: Milkereit, B. (Ed.), Proceedings of Exploration 07: Fifth Decennial International Conference on Mineral Exploration. pp. 473–483.

Miller, G. L., Talmor, D., Teng, S., Walkington, N., Wang, H., 1996. Control volume meshes using sphere packing: Generation, refinement and coarsening. Proceedings of the 5th International Meshing Roundtable, Sandia National Laboratories, 47–61.

Nalepa, M., Ansari, S., Farquharson, C. G., 16–21 October 2016. Finite-element simulation of 3d csem data on unstructured meshes: An example from the east coast of canada. In: SEG Technical Program Expanded Abstracts 2016. Dallas, pp. 1048–1052, doi:10.1190/segam2016–13949192.1.

Okabe, M., 1979. Analytic expressions for gravity anomalies due to homogeneous polyhedral bodies and translations into magnetic anomalies. Geophysics 44 (4), 730–741.

Pellerin, J., Botella, A., Bonneau, F., Mazuyer, A., Chauvin, B., Lévy, B., Caumon, G., 2017. RINGMesh: A programming library for developing mesh-based geomodeling applications. Computers and Geosciences 104, 93–100.

Pellerin, J., Lévy, B., Caumon, G., Botella, A., 2014. Automatic surface remeshing of 3d structural models at specified resolution: A method based on voronoi diagrams. Computers and Geosciences 62, 103–116.

Phillips, N., Oldenburg, D., Chen, J., Li, Y., Routh, P., December 2001. Cost effectiveness of geophysical inversions in mineral exploration: applications at San Nicolas. The Leading Edge 20 (12), 1351–1360, doi:10.1190/1.1487264.

Rücker, C., Günther, T., Spitzer, K., 2006. Three-dimensional modelling and inversion of DC resistivity data incorporating topography – I. modelling. Geophysical Journal International 166 (2), 495–505, doi:10.1111/j.1365–246X.2006.03010.x.

Shewchuk, J. R., May 1996. Triangle: engineering a 2D quality mesh generator and delaunay triangulator. In: Lin, M. C., Manocha, D. (Eds.), Applied Computational Geometry: Towards Geometric Engineering. Vol. 1148 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, pp. 203–222, DOI:10.1007/BFb0014497.

Shewchuk, J. R., September 2002. What is a good linear element? interpolation, conditioning, and quality measures. Proceedings of the 11th International Meshing Roundtable, Sandia National Laboratories, 115–126.

Si, H., February 2015. TetGen, a delaunay-based quality tetrahedral mesh generator. ACM Trans. on Mathematical Software 41 (2).

Si, H., 2017. TetGen: a quality tetrahedral mesh generator and a 3D delaunay triangulator, http://wias-berlin.de/software/tetgen/, last accessed November 2017.

The CGAL Project, 2017. CGAL: Computational Geometry Algorithms Library, https://www.cgal.org, last accessed November 2017.

Zehner, B., Börner, J. H., Görz, I., Spitzer, K., 2015. Workflows for generating tetrahedral meshes for finite element simulations on complex geological structures. Computers and Geosciences 79, 105–117.