

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/323241162>

Representation of the Drainage Network in Urban and Peri-urban Areas Using a 2D Polygonal Mesh Composed of Pseudo-convex Elements

Chapter · February 2018

DOI: 10.1002/9781119476726.ch5

CITATIONS

0

READS

231

9 authors, including:



Jorge Gironás

Pontifical Catholic University of Chile

81 PUBLICATIONS 650 CITATIONS

[SEE PROFILE](#)



F. Rodriguez

Institut Français des Sciences et Technologies des Transports, de l'Aménagement ...

82 PUBLICATIONS 599 CITATIONS

[SEE PROFILE](#)



Ximena Vargas

University of Chile

31 PUBLICATIONS 158 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



VEGDUD [View project](#)



VegDUD [View project](#)

Representation of the Drainage Network in Urban and Peri-urban Areas Using a 2D Polygonal Mesh Composed of Pseudo-convex Elements

5.1. Definitions and context

5.1.1. General context and objectives

Population growth and economic development have led to an increase in urbanization and to the transformation of rural zones into urbanized landscapes. This results in several problems related to water management [XIA 07]. Among the main problems associated with urbanization, we can highlight the increase in peak discharge [SMI 01], buildings in flooding areas, the decrease in infiltration and groundwater recharge [WMO 08], changes in base flow and hydrological regimes [MEJ 14] and the degradation of water quality due to the increase in pollution originating from urban areas [LAF 06]. To manage these territories in a sustainable way, Geographical Information System (GIS) tools able to represent realistically urban and peri-urban zones are needed. Presently GIS tools allow the delineation of catchments of medium to large scales (larger than 10 km²), and to extract the drainage network. On the other hand, for urban and peri-urban areas, composed of a mosaic of small elements (1–100 ha), current tools do not allow a correct representation of the hydrological connectivity between the various elements of the landscape.

Chapter written by Pedro SANZANA, Sergio VILLAROEL, Isabelle BRAUD, Nancy HITSCHFELD, Jorge GIRONAS, Flora BRANGER, Fabrice RODRIGUEZ, Ximena VARGAS and Tomas GOMEZ.

There are few studies dedicated to the extraction of the geomorphological characteristics of small urban catchments [ROD 13, JAN 13]. The most common way of obtaining a spatial representation of catchments is the use of a Digital Terrain Model (DTM) [GIR 10, PAS 10, SAN 16, ZEC 94] that represents topography using a regular grid. However, these DTMs do not preserve small urban elements the size of which is often smaller than the selected DTM grid, such as some urban elements (streets, sidewalks, retention basins) or ditches. Another possibility is to use topographical representations based on triangles, Triangular Irregular Networks (TINs). Triangle meshes can capture quick changes in topography and allow the incorporation of artificial elements [BOC 12]. However, this considerably increases the number of elements in engineered zones where the topography is seldom flat and also in situations in which we want to represent small elements, characteristic of peri-urban areas. With TINs, landscape units are also lost (fields, hedges, ditches) whereas they form the base of the hydrological modeling. Another way to represent landscapes is to use irregular polygons, such as in the Geo-MHYDAS tool [LAG 10], developed for agricultural landscape, and Geo-PUMMA [SAN 17] for peri-urban areas. To use such a mesh to represent urban and peri-urban areas, a good quality mesh is required, which means elements that respect some geometrical criteria that allow a good connectivity between hydrological elements, and consequently an accurate representation of water pathways. This means elements with a gravity center inside the polygon, and convex or pseudo-convex elements. Convex elements that are very long and thin (such as roads) may also be problematic as they can act as a barrier to water flow [JAN 11]. In the following, we will assume that an irregular/heterogeneous mesh, composed of pseudo-regular elements (pseudo-convex or pseudo-squared) can be considered as a good quality mesh to extract the drainage network required for hydrological modeling. If not, we will speak of badly-shaped elements.

This chapter presents a module implemented in QGIS, belonging to the GEO-PUMMA toolbox [SAN 17]. This plugin allows a triangulation of surface elements with a vectorial format, and thus the segmentation of a mesh composed of heterogeneous polygons. This module uses the MeshPy library that is included in the Triangle® software [SHE 06]. This module allows the generation of vectorial meshes (composed of irregular polygons and linear elements) for the representation of urban and peri-urban landscapes with a high resolution, whereas classical tools do not provide a realistic solution for the determination of drainage networks in complex landscapes, highly modified by the human activity, with in particular road networks and sewer networks.

The first part describes the basic concepts needed to understand the generation of an irregular mesh in Hydrological Response Units (HRUs) [FLU 95]. Then, we present how the badly-shaped elements that affect the model mesh quality are selected. In the chain that allows the model mesh improvement, the plugin

implemented in QGIS allows the decomposition of this type of polygon in triangles. We present the implementation of the TriangleQGIS and its installation in Windows, as well as the options to use it and the final result on the mesh of the Mercier catchment, a subcatchment of the Yzeron catchment, located in the suburbs of Lyon, France. This is an experimental catchment, representative of peri-urban catchments, where hydrological modeling works were conducted using the PUMMA model [FUM 17, JAN 14] and for which the required information was available at the required resolution. For one of the steps, we will also use a GIS layer from the El Guindo catchment in Chile [SAN 17].

To perform the segmentation and the generation of the drainage network, the TriangleQGIS plugin, as well as the Python scripts of Geo-PUMMA, implemented in GRASS-GIS will be used. The plugin and the Geo-PUMMA scripts are available at <https://forge.irstea.fr/projects/geopumma>. A Geo-PUMMA installer, as a virtual box for Windows users, is also available at the following link: <http://doi.org/10.5281/zenodo.821563>. The TriangleQGIS plugin can only be used in Windows and works properly until the 2.14 version of QGIS (in the virtual box, the installed version of QGIS is the 2.8). It can be used without the virtual box. On the other hand, if the user wants to realize all the steps described in the chapter to obtain the hydrological mesh, it is necessary to use Geo-PUMMA and the 6.4 version of GRASS¹, as well as the libraries that are included in the virtual box. The use of the virtual box avoids the independent installation of all the libraries with compatibility problems that can occur. Vectorial GIS layers used in the examples are located in the /home/geopumma/Database_Tutorial_GeoPUMMA_v1 folder of the virtual box and are also provided with this chapter.

The specifications of the GIS layers used in the example are the following:

Thematic content	HRUs layers of the Mercier and El Guindo subcatchment
Geographical accuracy	The minimum represented area is 10 m ²
Thematic accuracy	Associated with the input layer
Projection system	Lambert 93 – French Geodesic Network 1993 (Mercier, Lyon France) WG84/19S (El Guindo, Chili)

¹ The Geo-PUMMA tool was developed with the 6.4 version of GRASS that is not compatible with version 7 of GRASS. The functions arguments used in this chapter are different between versions 6 and 7 of GRASS. Therefore, we recommend the installation of the virtual box that will avoid errors linked to the calling of GRASS 6 functions from version 7.

5.1.2. Derivation of input GIS layers

The first step is to gather all the GIS layers that are relevant for urban and natural areas of the catchment. In urban areas, we need a vectorial GIS layer representing the cadastral elements. It is also necessary to have the following layers: buildings, built public zones such as streets, paths, squares, parks, sport areas, regulation infrastructures and all the structures that have a hydrologically relevant role. If these data are not available, it is necessary to digitize them from maps or orthophotos. Digital Elevation Models must have a high spatial resolution, a maximum grid size of 2 m being the acceptable limit. From this information, we start delineating the catchment boundaries and its subcatchments. This task requires specific tools, as, in this kind of highly modified landscape, water pathways do not necessarily follow the topography. Jankowsky *et al.* [JAN 13] present a methodology that integrates sewer networks and limits manual operations, combining an object-oriented approach like the one presented in this chapter for urban areas and standard DTM analyzes for rural areas. It is also necessary to collect, through field surveys, information about water flow direction in sewer networks, ditches or hydraulic structures. Once the catchment boundary is determined, all other GIS layers (land use, subcatchments, soil type, geology and others) are clipped so that all the layers have the same boundaries as the catchment ones. Subsequent steps rely on the overlay of all these layers.

Figure 5.1 shows the main GIS layers used for the generation of the hydrological mesh. These vectorial layers which are overlaid to get the input map for the algorithm are composed of polygons (land use, soil types, subcatchments, geology, etc.) and polylines (waste water and rain water sewer networks, combined sewer networks, channels, water courses, etc.). At the end of this data preparation step, land use, soil type, geology and all the other layers considered relevant for the represented hydrological processes should be available.

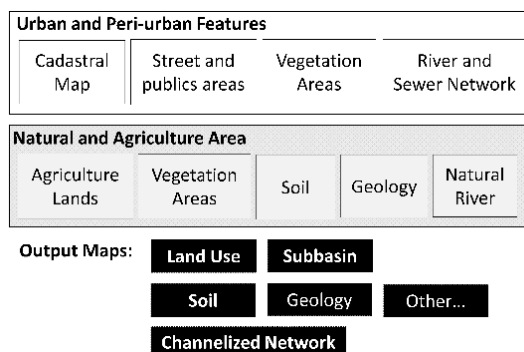


Figure 5.1. Digitization and preparation of basic GIS layers

All these layers must be overlaid to produce the initial vectorial mesh. Two options can be used to achieve this intersection: the first one uses the *m.seg* script, developed with the Geo-MHYDAS tool [LAG 10], which allows intersecting polygons and polylines. The second option uses the union option of QGIS applied to all layers in order to get a unique input layer. Then, it is necessary to split the polygons crossed by a polyline using the vector edition tool (split), and finally, clean the topology with the script *p.clean_topology.py* which belongs to Geo-PUMMA. This way, interfaces (limits between polygons) are preserved in the final layer and possible topology errors are corrected.

QGIS options used to join various GIS layers:

- Union tool: *Vector > Geo-treatment tools > Union...*
- Edition layer tool: *Enter the edition model then edit > Split the elements*

5.1.3. Identification of badly-shaped HRUs and methodology to improve the model mesh quality

The generation process of the GIS input layer includes all the relevant elements of the urban and peri-urban landscape. In the case of HRUs, polygons may be very irregular and heterogeneous. They must be segmented so that the drainage network representation is not altered. In this chapter, we do not deal with the generation of Urban Hydrological Elements (UHEs) that correspond to urban elements, composed of a cadastral field, the sidewalk and half of the adjacent road [ROD 08]. Their delineation can be performed using scripts available in Geo-PUMMA. In the following, we do not segment UHEs, the shape of which was rather regular in the treated case study. However, UHEs can also have very irregular shapes and the tools presented in this chapter may also be applied to UHEs.

Figure 5.2 shows examples of badly-shaped HRUs that were obtained after GIS vectorial layers overlay. They have an irregular shape: either non-convex or long and thin. Generally, these elements are associated with road sections or footpaths, green areas such as hedges, or the shape of agricultural fields (Figure 5.2).

These kinds of elements, that generally have their gravity center outside the polygon, lead to problems in the hydrological connectivity. This explains why it is necessary to segment them into smaller, pseudo-regular elements, without significantly increasing the final number of elements, which would affect computing performance.

Shape descriptor	Formula	Range
Form Factor, FF	$16A/P^2$	[0,1]
Convexity Index, CI	P_{convex}/P	[0,1]
Solidity	A/A_{convex}	[0,1]
Compacity	$((4/\pi)A)^{0.5}$	[0,1/ π]

Table 5.1. Considered geometrical factors. A is the polygon area, P , its perimeter and A_{convex} and P_{convex} are the area and perimeter of the convex polygons inside which the polygon is included

A first selection of badly shaped elements is performed based on the computation of geometrical descriptors proposed by Russ [RUS 02]. Table 5.1 presents the main descriptors used that must be computed for each initial mesh. This can be achieved in QGIS, using the computation of geometrical parameters such as area and perimeter. It is also possible to get from QGIS the area of the convex polygon hull inside which the polygon is included, and then the values of the corresponding perimeter and area, P_{convex} and A_{convex} . The other option is to use the *p.shape_factors.py* Geo-PUMMA script.

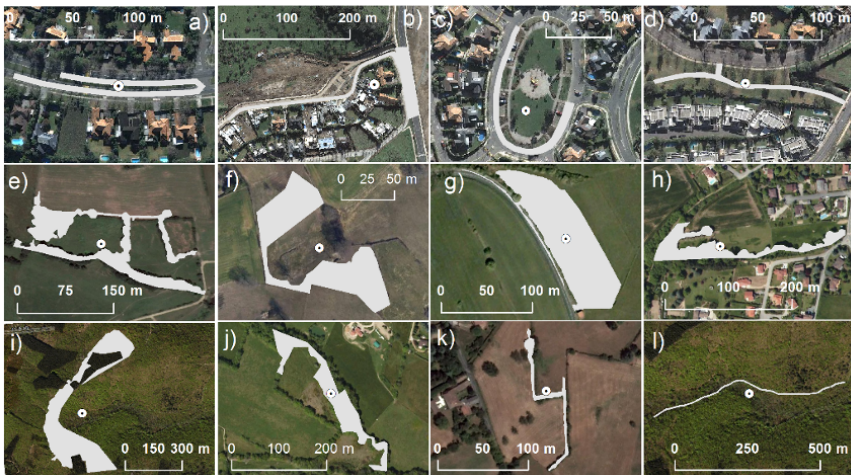


Figure 5.2. Examples of HRUs with very irregular shapes (gray) that must be improved. These are road sections and footpaths (a, b, c, d, l), green areas (e, g, h, j, k) or agricultural fields (f, i). For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

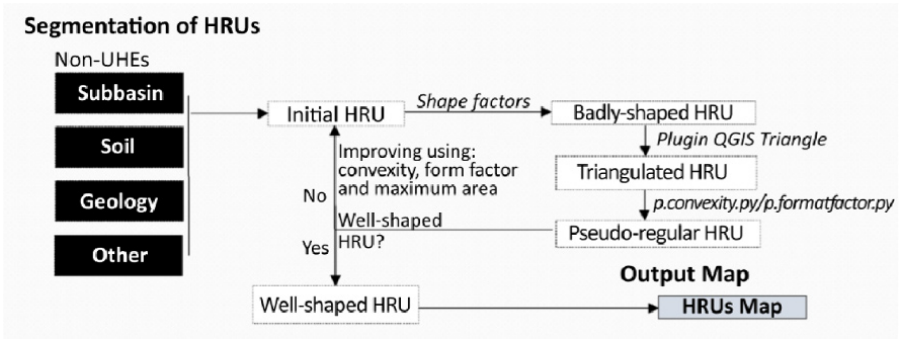


Figure 5.3. Steps for the segmentation of badly-shaped HRUs

QGIS tools for the estimation of geometrical factors:

- Open the calculator=> Geometry tool (this tool includes functions that compute geometries such as length, area, perimeter).

Figure 5.3 presents the HRUs generation step. In a first step, we identify badly-shaped elements from geometrical descriptors presented in Table 5.1. In a second step, the TriangleQGIS plugin is used for the triangulation of badly-shaped elements. Finally, the last step is the dissolution of triangles into pseudo-regular HRUs, according to the selected geometrical descriptor. This step allows the generation of a limited number of polygons that at best respect the initial polygon boundary, while keeping a limited number of polygons.

Criteria that allow the identification of HRUs leading to problems in the extraction of the drainage network are the following:

1) Non-convex HRUs: this type of element can generate problems in hydrological connectivity, because tools that compute the water flow distance between the polygon and its neighbor use the gravity center as the representative point of the polygons. In general, polygons with a Convexity Index $IC \geq 0.70$ do not generate problems. In our example, we use a threshold value of the convexity index ICT that is equal to $ICT = 0.75$ to define non-convex elements.

2) Very long and thin HRUs: this kind of element can generate problems in the hydrological connectivity when an average altitude is assigned to a polygon and considered as representative. If the element is very long, this mean altitude is no longer representative and the element can act as a barrier to water flow and

erroneously modify water flux direction. To avoid this problem, we recommend the segmentation of those polygons into smaller elements. In general, polygons with a Form Factor threshold (FFT) > 0.20 do not lead to problems.

3) Too large HRUs: this kind of element can generate heterogeneities in water fluxes, in particular if the latter are computed between large polygons, adjacent to much smaller ones. In our example, we will consider that an HRU area cannot exceed 2 ha.

Figure 5.4 shows an example of badly-shaped elements that must be improved through the segmentation procedure. This subcatchment forms part of the Mercier catchment that is the main case study of this chapter.

In the following, we describe the tool that allows the automatic segmentation of each badly-shaped element from GIS layers in the .shape format.



Figure 5.4. *Badly-shaped elements in the Mercier subcatchment. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip*

5.2. Implementation of the TriangleQGIS module and general methodology

5.2.1. Used technologies

The use of GIS for preprocessing input information for a hydrological model has become a requirement, given the large amount of geographical information that must be manipulated. GIS platforms facilitate those treatments. The most used (among free platforms) are QGIS and GRASS. QGIS has a Graphical User Interface that can be incorporated into other GIS systems or databases and includes plugins developed by the users. All these reasons led to the decision to implement the Triangle plugin in QGIS. The plugin allows the use of the Triangle® software and working directly with vectorial layers in the .shape format, without needing to export layers to the native format of Triangle® (.node, .ele, .ploy, .area, .edge and .neigh files).

The Python language is widely used to implement plugins in QGIS and we used it as the most appropriate for the implementation of our plugin. In addition, thanks to Python libraries, it is possible to emulate the use of QGIS and GRASS-GIS and simplify the preprocessing step of distributed hydrological models.

5.2.2. Context and general methodology

The main motivation for developing the QGIS plugin presented in this section is the segmentation of highly non-convex polygons. In general, available triangulation algorithms allow the generation of triangles that form a convex closure, but that do not preserve boundaries of the original polygon. In addition, classical triangulations do not allow the imposition of restrictions on the area or angles of triangles that compose the polygon. These features are included in the Triangle® software, developed by Jonathan Shewchuck [SHE 96]. This software also provides the most efficient and robust triangulation method to improve our 2D model mesh. The TriangleQGIS plugin allows the activation of the Triangle® software from QGIS, using polygonal elements in the .shape format.

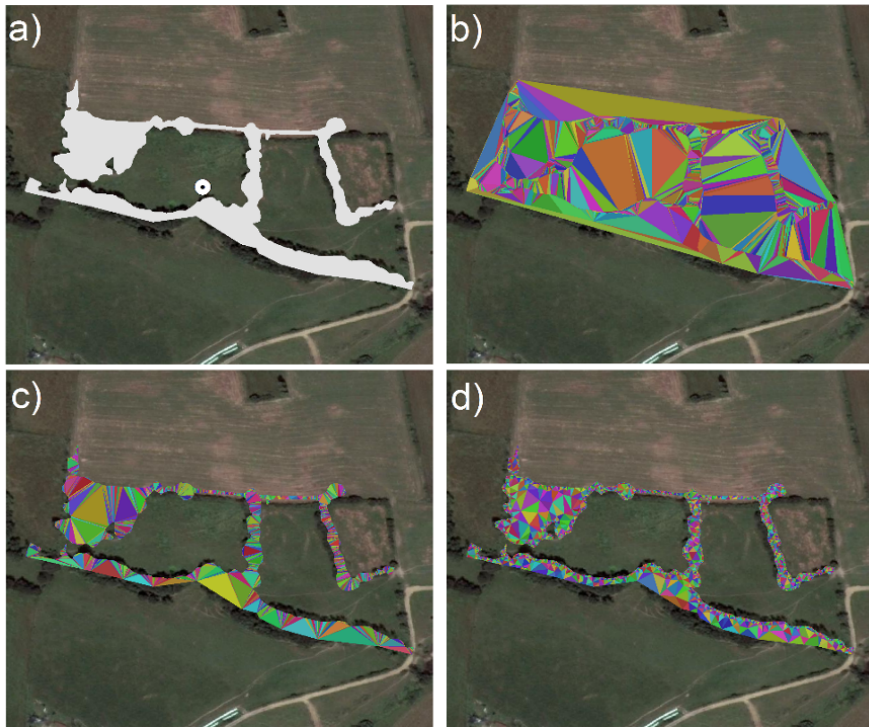


Figure 5.5. a) HRU representative of a green area. b) Classical Delaunay triangulation: 4,667 triangles. c) Triangulation without restrictions on area using Triangle®: 2,340 triangles. d) Triangulation with angle restriction to 20°: 5,247 triangles. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

QGIS tool to realize a triangulation:

- Triangulation tool: *Vector > Geometrical tools > Delaunay triangulation.*

Figure 5.5 shows an HRU associated with a green area and the results of its triangulation using various options of the software. Figure 5.5(b) shows the triangulation of the whole convex domain, known as “Delaunay” triangulation, obtained from the “vector” menu of QGIS. The drawback of this option is that the triangular mesh uses only nodes and can only generate a convex envelope. The original boundary of the polygon is lost. This is not the mesh we want to obtain, as we want to preserve the original polygon shape. Figure 5.5(c) shows the triangulation obtained using the Triangle® tool. The decomposition in triangles uses the original polygon, without generating the convex envelope. Figure 5.5(d) shows a triangulation also obtained with the Triangle® tool, but with an additional restriction of the maximum area of the triangles ($A_{max} = 2$ ha). To fulfill this criterion,

additional nodes are introduced inside the polygon. This is automatically done using the Triangle® software to respect the imposed criteria on areas (or angles). The Triangle® software also proposes other options that allow putting restrictions on angles of the generated triangles for each polygon. For all these reasons, this software was very well suited to improve the geometry of the HRUs and was implemented in QGIS.

To decompose the polygons of our model mesh, we chose to generate a Delaunay triangulation with restrictions (using Triangle®). However, even if the elements obtained after this step are convex, there is a large number of triangles. This can be detrimental to the computing performance of the numerical model that will use this mesh. In addition, we would like to get polygons as close as possible to the shape of the initial polygons.

As a consequence, the triangulation step is followed by a step called “dissolution”. It allows the preservation of the shape of HRUs, as elements of the landscape. The dissolution step (which is a grouping process) takes a triangle and combines it with its neighbor having the largest area. This process is continued until the maximum threshold for the criteria is exceeded. This dissolution step is applied to all triangles that have not been dissolved. Finally, this decomposition leads to final polygons that are more satisfactory for use in the hydrological model.

Polygons, the areas of which are larger than the maximum areas fixed by the user, may also be segmented in the same way as non-convex polygons that have their gravity center outside the polygon, using a restriction on the maximum area of the triangles, both in the initial triangulation phase and in the dissolution phase of those triangles.

In section 5.2.3, we present the work realized for the implementation and the execution of the plugin in QGIS (version 2.14 or below), complemented with the presentation of the use of some scripts from Geo-PUMMA in order to provide a whole view of the proposed methodology.

5.2.3. Structure of the QGIS plugin

To implement a plugin in QGIS, it is necessary to create some files and classes that implement some methods, or other files, containing all the relevant information for the plugin. A QGIS plugin called “Plugin Builder” already exists. It allows the user to generate the basic structure of a functional plugin. The files generated by this plugin are the following:

- *init.py*: this is the starting point of the plugin. Its role is only to initialize and set as output the main plugin;

– *metadata.txt*: this file contains the plugin metadata including: its name, the QGIS version required for its use, the author(s);

– *mainPlugin.py*: this class is the main class of the plugin, in which the specific functionalities of the plugin are implemented;

– *resources.qrc*: this file contains references to files, external to the plugin, like images or other files. It must be compiled using *pyrcc4* to produce the *resources.py* file, allowing the use of the external files by the plugin;

– *mainPlugin.ui*: this file contains the graphical interface of the main plugin. It must be compiled using *pyuic4* to get the *mainPlugin.py* file. This will allow the use of the plugin graphical interface;

– *mainPluginDialog.py*: the main objective of this class is the initialization of the plugin graphical interface.

5.2.4. Basic used library: MeshPy

One of the interesting characteristics of the MeshPy library is its ability to impose restrictions on the maximum area and on the minimum angle of the triangles that ensures the mesh quality.

This library is essential to the plugin as the Triangle® is incorporated into this library, so that it can interact with QGIS vectorial layers. Thus it allows the performance of triangulations with restrictions on the maximum surface or the minimum angle. Moreover, the library allows the generation of 2D meshes (with Triangle®) as well as 3D meshes (using tetrahedrons in TetGen12). The use of this library for the 2D triangulation is very similar to what would be done using Triangle® directly.

We also used the “Plugin Builder” plugin to get the basic structure of a functional plugin. We also mention the “Plugin Reloader” plugin that allows the immediate reloading of the plugin during its development, thus avoiding having to load QGIS each time the plugin must be loaded. To realize the graphical interface, we used QtCreator, which allows easy editing of the plugin graphical interface, called *mainPlugin.ui*. QtCreator is available here: <https://www.qgis.org/en/site/getinvolved/development/qgisdevelopersguide/qtcreator.html> .

5.2.5. Installation of the plugin in Windows

To run the TriangleQGIS plugin, the user must install QGIS (version 2.14 or below), Python 2.7 and the MeshPy library. The following steps were conducted on Windows XP and Windows 7.0 computers.

Install the following archives:

- 1) Quantum GIS: QGIS-OSGeo4W-2.4.0-1-Setup-x86_64;
- 2) Python: python-2.7.amd64;
- 3) MeshPy library: MeshPy-2014.1.win-amd64-py2.7.

Once these tools are installed, the folder “meshpy”, located in C:\Python27\Lib\site-packages, is copied into the folder C:\Programes\QGIS Chugiak\apps\Python27\Lib\site-packages as shown in Figure 5.6.

It is necessary to verify in the Python windows of QGIS that the MeshPy library is running correctly. For that, the user can type the following command: “import meshpy”. If there is no output message, it means that everything is running correctly. If not, previous steps must be verified. Figure 5.7 shows the Python windows and the result of the “import meshpy” command.

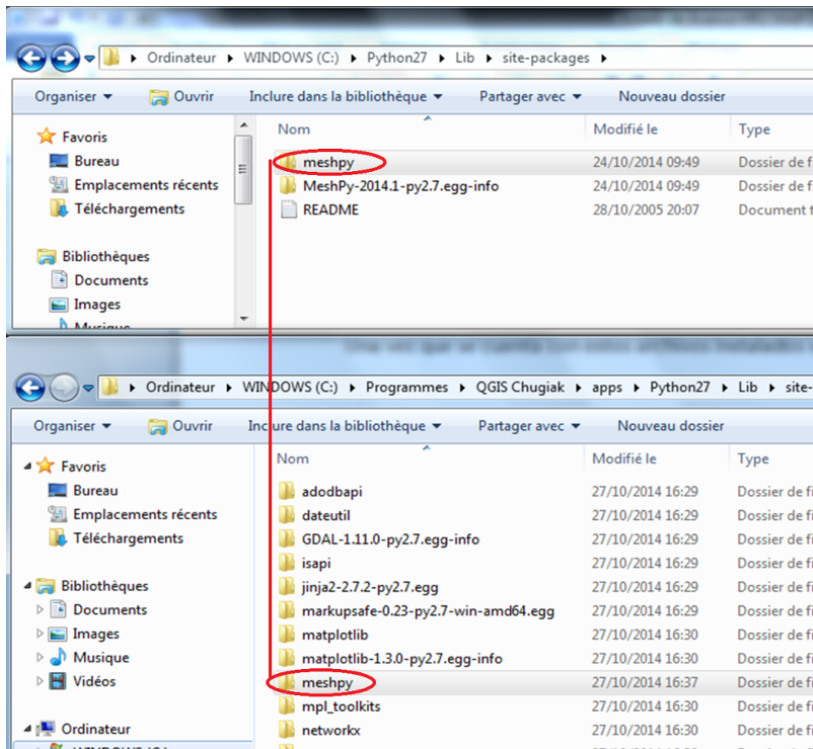


Figure 5.6. Adding the MeshPy library to Python2.7. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

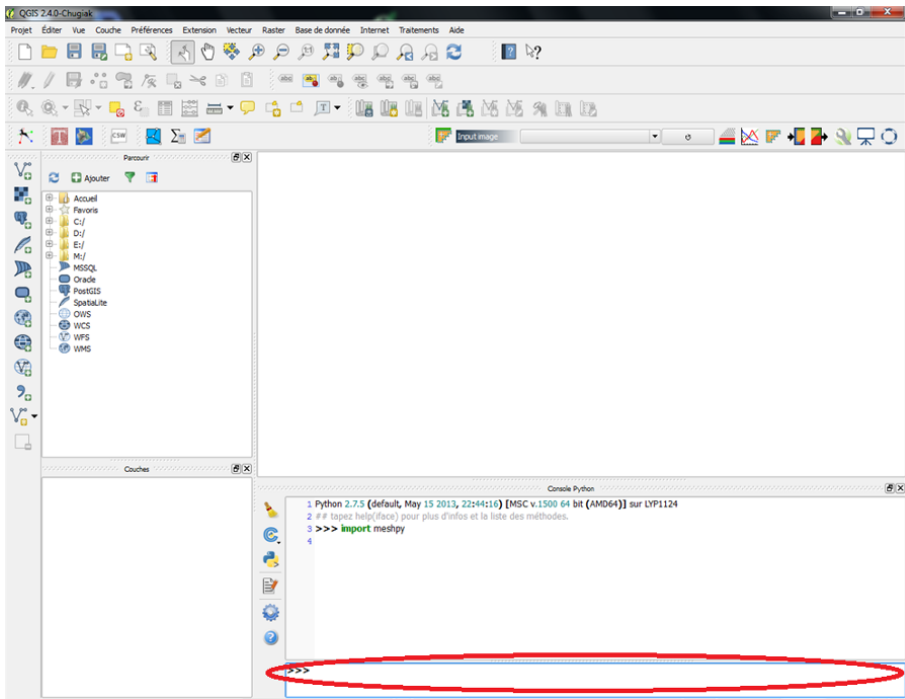


Figure 5.7. Verification of the correct installation of the MeshPy library in the Python windows in QGIS

The last step is to copy the TrianglePlugin archive to the folder where the QGIS plugins are located: “C:\Users\[username]\.qgis2\python\plugins” (Figure 5.8).

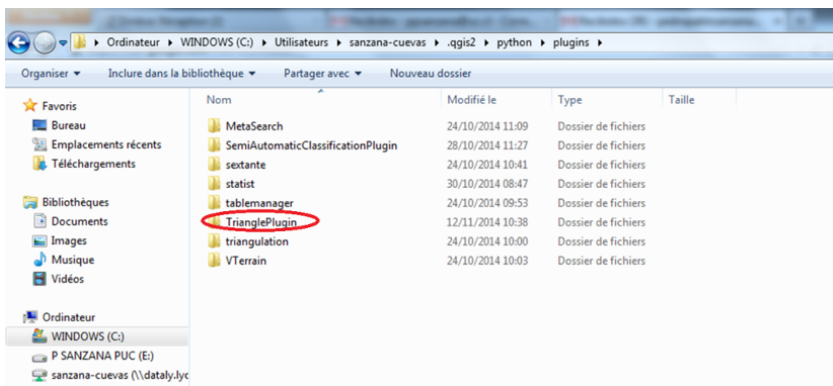


Figure 5.8. Folder where the TrianglePlugin archive must be copied

Finally, like any QGIS plugin, it must be activated from the QGIS toolbar by selecting Plugins>Manage and Install Plugins and activating TrianglePlugin (Figure 5.9).

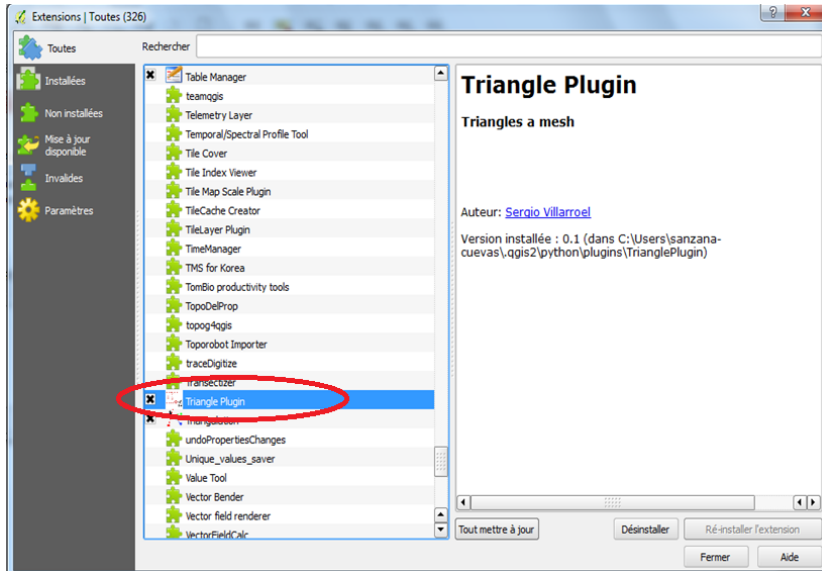


Figure 5.9. Activation of the TriangleQGIS plugin

Figure 5.10 shows the TriangleQGIS plugin interface.

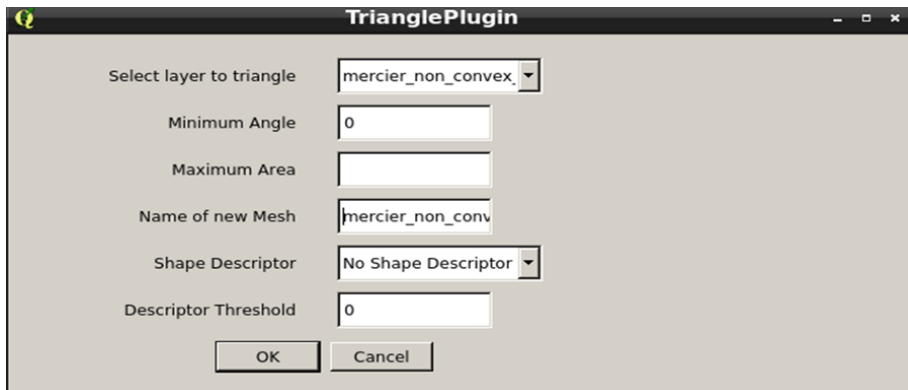


Figure 5.10. The TrianglePlugin interface

5.2.6. Installation of the virtual box, QGIS plugin and Geo-PUMMA

To install the “Virtual Box Machine”, it is recommended to visit the website (Figure 5.11) (<https://www.virtualbox.org/>) and to download, from the “download” tab, files corresponding to the operating system (Figure 5.12).



Figure 5.11. Landing page of VirtualBox

Download VirtualBox

Here, you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox 5.1.10 platform packages.** The binaries are released under the terms of the GPL version 2.
 - [Windows hosts](#)
 - [OS X hosts](#)
 - [Linux distributions](#)
 - [Solaris hosts](#)

Figure 5.12. “Download” section of the VirtualBox website

To install VirtualBox, the downloaded archive must be executed, following the instructions of the dialog box. Figure 5.13 shows the first screen. It is only necessary to click “Next”. The second screen (Figure 5.14) allows the installation folder to be chosen. Accept the default options. Click “Next”. The third screen (Figure 5.15) allows the selection or not of the option for the creation of shortcuts. Choose the selected options and click “Next”. The fourth screen (Figure 5.16) gives a warning on the possibility of losing the connection during installation. Click “Yes” to continue. The fifth screen (Figure 5.17) allows the validation of the choices before launching the installation. If everything is OK, click “Install”. Windows users may receive a warning similar to the one shown in Figure 5.18. Click on “Install” to continue the installation process.

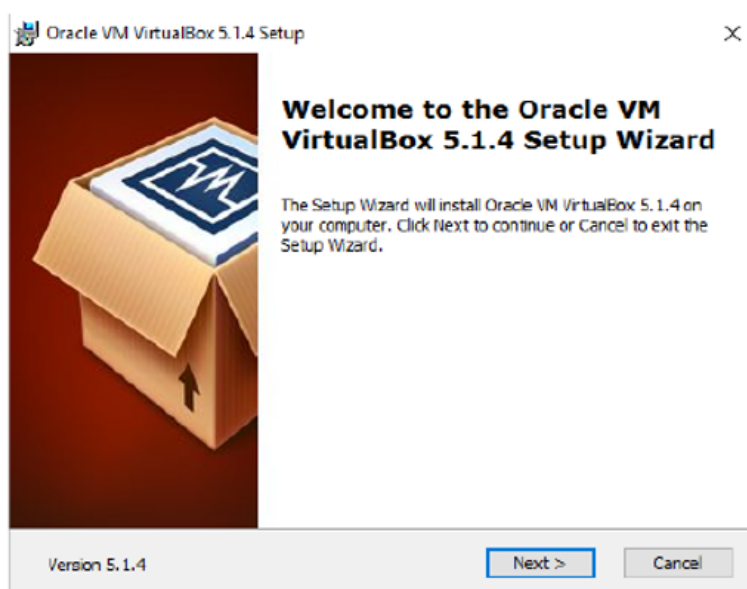


Figure 5.13. First screen of the installation dialog box

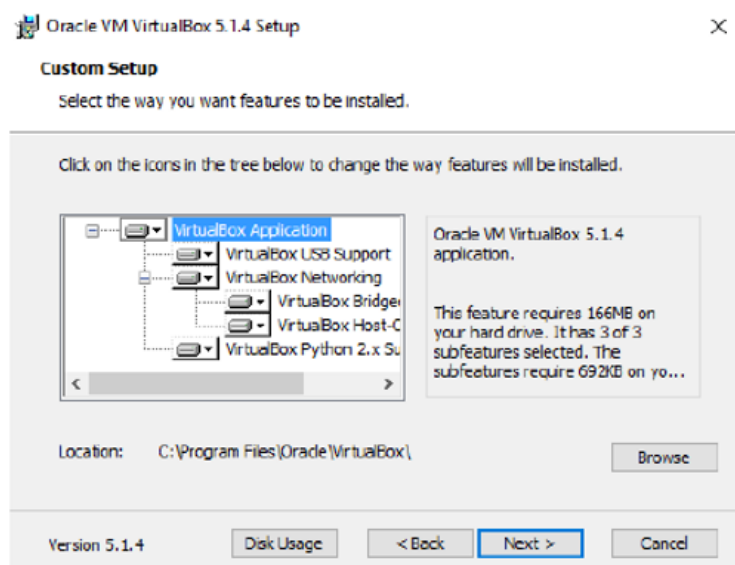


Figure 5.14. Second screen of the installation dialog box

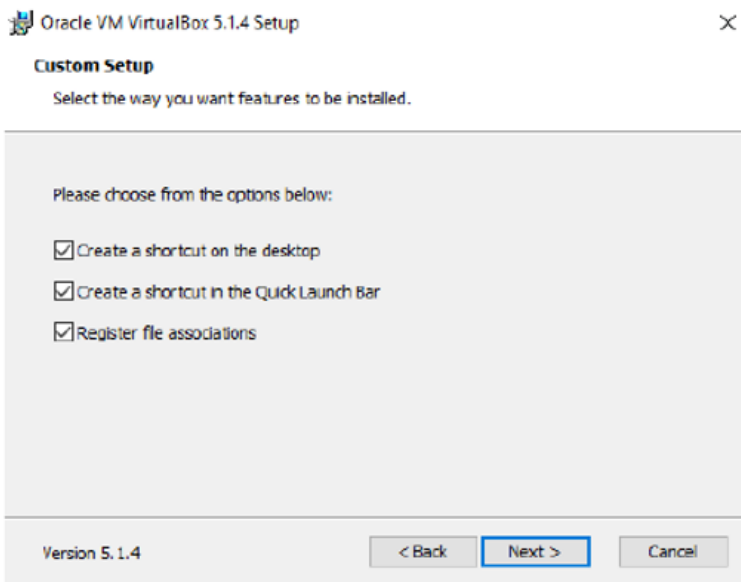


Figure 5.15. Third screen of the installation dialog box



Figure 5.16. Fourth screen of the installation dialog box

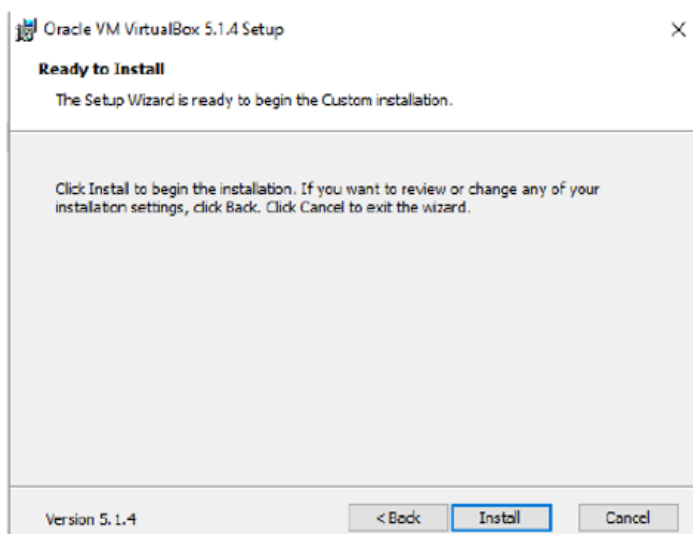


Figure 5.17. Fifth screen of the installation dialog box

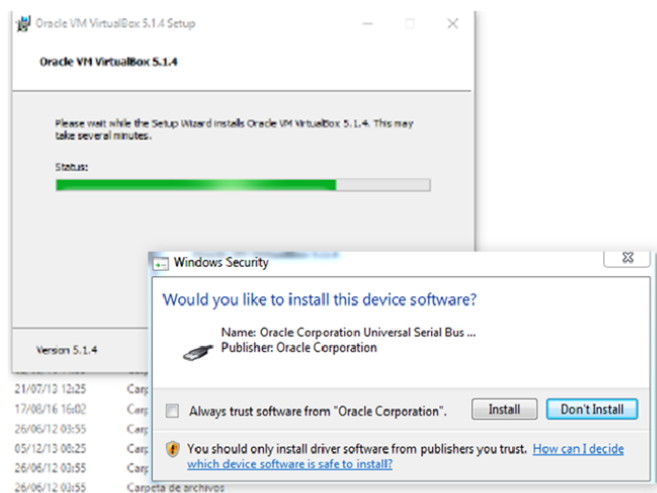


Figure 5.18. Sixth screen of the installation dialog box

Once the installation is terminated and successful, open VirtualBox (Figure 5.19).

What follows is related to the virtual box installation, allowing the running of Geo-PUMMA. The source of the scripts are available here² and the virtual box installation package here³. For that, click on “File”, then select “Import Virtual Service”. A screen similar to that shown in Figure 5.20 appears. Select the Virtual Box file and click on “Next”. This operation can take some time given the size of the file.

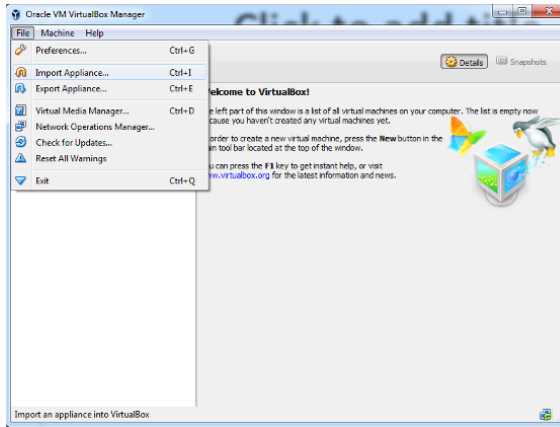


Figure 5.19. Main screen of Virtual Box, option File ==> Import Appliance

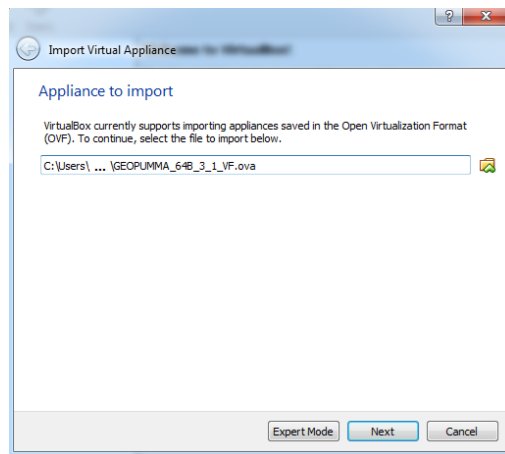


Figure 5.20. Screen to import virtual services

2 <https://forge.irstea.fr/projects/geopumma>.

3 <http://doi.org/10.5281/zenodo.821563>.

Once the virtual machine is loaded, it is necessary to verify if the operating system allows the use of 64-bit virtual boxes. If this is not the case, only the Ubuntu option (32-bit) will appear (see Figure 5.21 for the details of the verification of the existence of a problem or not).

If the 64-bit option does not appear, try the following commands:

- 1) switch off the computer;
- 2) switch it on again and immediately press the ESC or F10 button to access the BIOS (some computers allow the access to this functionality with another button);
- 3) in the BIOS menu, activate the option that allows the use of virtual machines;
- 4) accept and restart after saving the modifications.

Once the computer is switched on again, the Ubuntu option (64-bit) should be available.

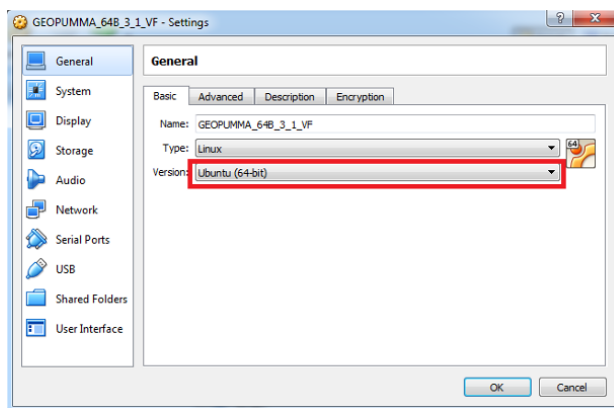


Figure 5.21. Window for the configuration of the virtual box

The disk space allocated to the virtual box cannot be accessed from the original operating system. However, it is possible to configure a shared folder that will be accessible from the operating system and from the virtual box. To do that, in the configuration menu, go to “Shared folders” and select the desired location by clicking on the chosen folder. Create a folder named “name_folder” on the hard disk and select it. Select also the option “Auto-mount” and “make permanent”. Then click “OK” to close the window and click again on “OK” to add the folder (see Figure 5.22 for the visualization of these steps).

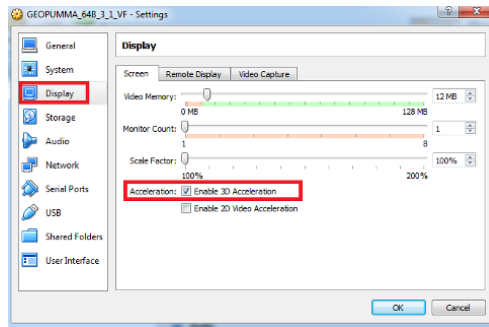


Figure 5.24. Option to activate 3D acceleration. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

Then, click to show the launch screen of the virtual box (Figure 5.25).



Figure 5.25. Welcome screen of the Geo-PUMMA virtual box

5.3. Illustration of the TriangleQGIS plugin and some Geo-PUMMA scripts

Explanations provided in this chapter allow an understanding of how the tools are working. If the user wishes for more details on the use of a specific GRASS or QGIS command presented in this chapter, they can refer to the Geo-PUMMA User's manual available at this link⁴ that provides details on the various steps.

⁴ <https://forge.irstea.fr/projects/geopumma/files>.

The following steps illustrate the triangulation of badly shaped polygons. As explained in the methodology, it is first necessary to select non-convex elements, elements which are too long and thin, and elements with an area which is too large.

The triangulation step is important to improve the hydrological model mesh. That is why, according to the chosen improvement criteria (non-convexity, maximum area, or long and thin elements), it could be necessary to add nodes to get a satisfactory triangulation. This is not necessary when polygons are triangulated because of the convexity criterion. On the other hand, nodes are added inside the polygon for polygons with an area which is too large, and nodes are added at the polygon edges in order to get smaller triangles for long and thin polygons.

5.3.1. Insertion of nodes for long and thin polygons

The method to add nodes at the edges of long and thin elements is described here. A maximum distance of 5 m between two nodes on the boundary of long and thin elements is recommended. To insert nodes, the *v.split* GRASS command can be used as explained below.

```
v.split input=mercier_long_elements output=mercier_long_elements_split length=5
```

This way, we ensure that the triangulation will lead to small-sized triangles that allow us to build satisfactory polygons in a second step. Figure 5.26 shows the initial polygon, associated with a road segment (Figure 5.26(a)) with nodes initially separated with irregular distances (Figure 5.26(b)). After inserting nodes with a maximum distance, a polygon with nodes separated by a maximum distance of 5 m is obtained (Figure 5.26).

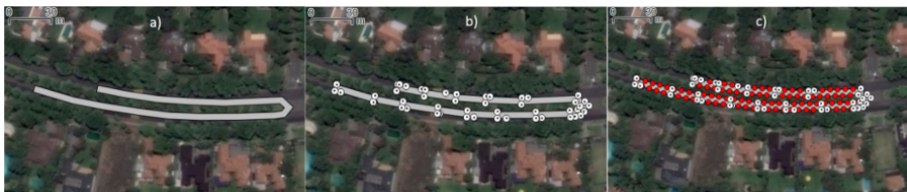


Figure 5.26. a) Polygon associated with a road segment, b) with the initial nodes and c) the newly inserted nodes in red that ensure a maximum distance of 5 m separating the nodes. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

GRASS function to insert nodes:

- v.split

QGIS tool to verify the nodes:

- Triangulation tools: *Vector > Geometric tools > Nodes extraction*

5.3.2. Triangulation using the TriangleQGIS plugin

The TriangleQGIS plugin can be used directly with vectorial layers in the .shape format. These layers can be obtained from various GISs (ArcGIS, GRASS-GIS, OpenJump, OrbiGIS, etc.). We recommend saving then again in the QGIS .shape format, using options described in Figures 5.27. It is recommended to do this step for all the files that contain elements selected from the convexity criteria (*mercier_non_convex*), the maximum area criterion (*mercier_big_areas*) and the long and thin elements criterion (*mercier_long_elements*). Figure 5.27 shows the steps to follow (1–5 to save the files in the vectorial format).

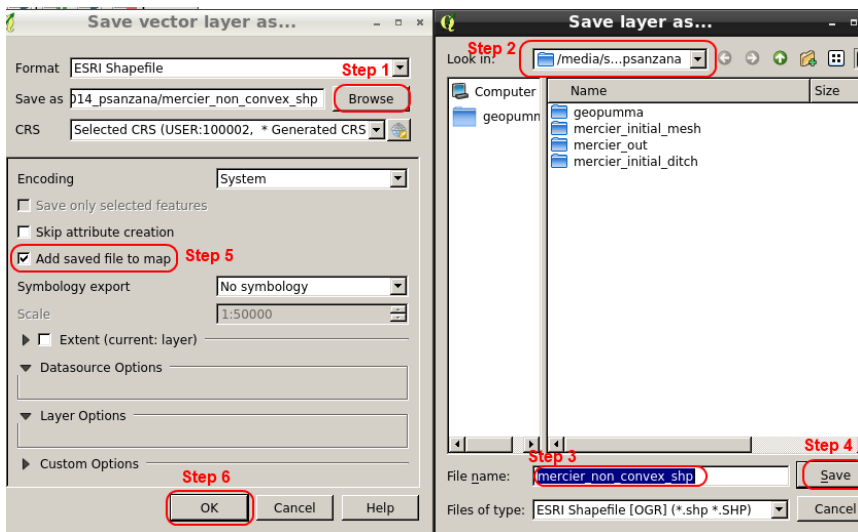


Figure 5.27. Steps to save a file in the QGIS vectorial format. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

First, the identifiers of the badly-shaped elements must be loaded. Elements selected from the convexity criterion are triangulated first. Figure 5.28 shows the selected badly shaped elements from the convexity criterion.

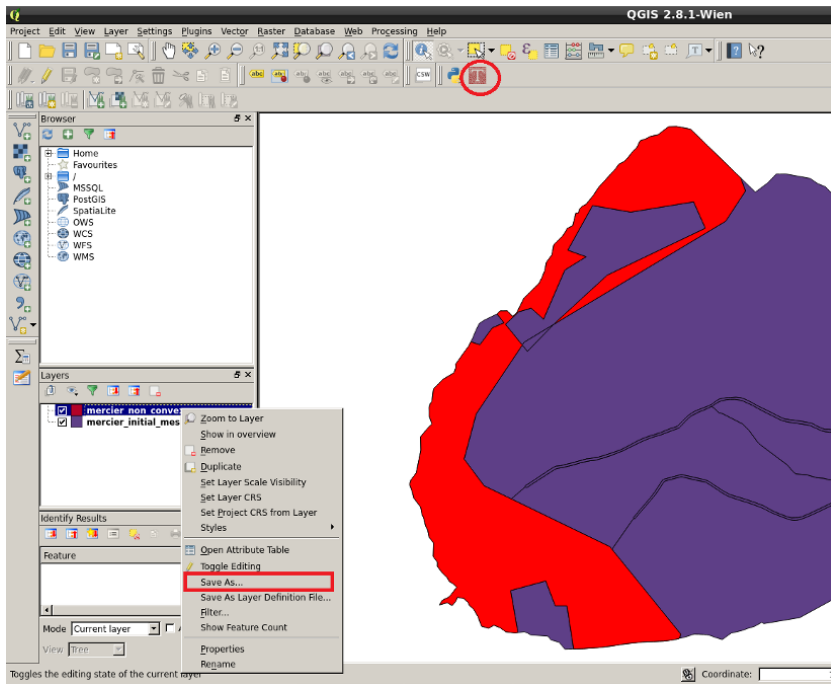


Figure 5.28. HRU selected with a Convexity Index = 0.590. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

Figure 5.29 shows the interface for choosing the various triangulation options available in the TriangleQGIS plugin. Table 5.2 shows how to fill the fields for non-convex elements.

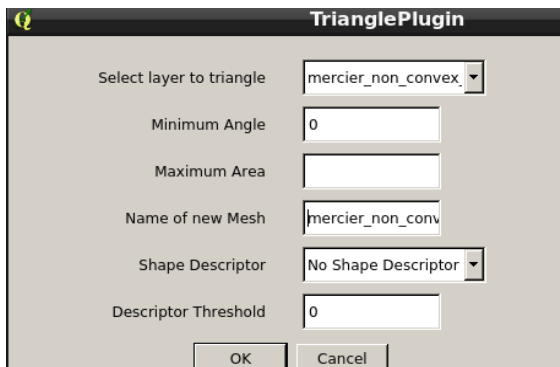


Figure 5.29. Interface of the TriangleQGIS plugin

Case	Non-convex elements layer. No constraint in the triangulation	Non-convex elements layer. Triangulation with constraint on the maximum area 20,000 m ²	Non-convex elements layer. Triangulation with constraint of 30° on the minimum angle
Input GIS layer to triangle	mercier_non_convex_shp	mercier_non_convex_shp	mercier_non_convex_shp
Minimum angle	0	<i>Keep empty</i>	30
Maximum area	<i>Keep empty</i>	20,000	<i>Keep empty</i>
Name of the new mesh	mercier_non_convex_t	mercier_non_convex_area	mercier_non_convex_angle
Shape described	No shape descriptor	No shape descriptor	No shape descriptor
Descriptor threshold	0	0	0

Table 5.2. Options to generate the various triangulations with the TriangleQGIS plugin (to fill the required fields in the graphical interface)

Table 5.3 shows the result of each triangulation. For the new generated vectorial mesh, comparison criteria are the number of triangles and statistics on areas and minimum angle. In the first example, a triangulation without restriction on the maximum area and minimum angle was selected. In this case, the triangulation uses only existing nodes on the initial boundary (the total number of obtained triangles is 174). In the second example, the triangulation is realized using a restriction on the maximum area of 2 ha. In this case, the program inserts new nodes inside the polygon so that the generated triangle size is lower than 2 ha. The total number of triangles is 203 triangles. In the third example, the triangulation is realized with a restriction on the angles that must be larger than 30°. This requires inserting new nodes inside the polygon as previously. The total number of triangles is 814 triangles. The number of triangles generated from a single polygon can dramatically increase the final number of elements in the model. We see that up to 800 triangles can be generated.

Although the triangulation with restriction on the area (Figure 5.30(c)) or restriction on the angles (Figure 5.30(d)) leads to well-shaped elements, the final element number is very large and includes very small triangles, so that the final result cannot be considered as satisfactory for the modeling. The method providing

the best compromise between the number of final elements and “good shape” was finally retained, i.e. the triangulation without restriction (Figure 5.30(b)).

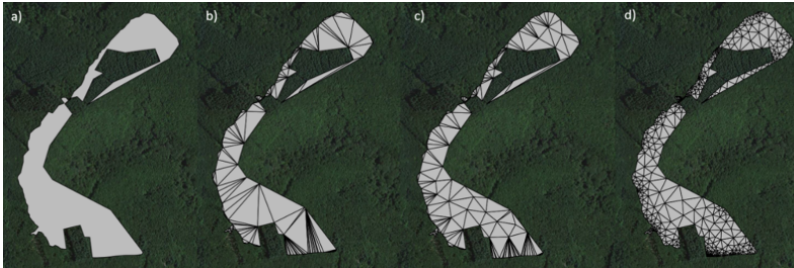


Figure 5.30. a) Non-convex polygon, b) triangulated without restriction, c) triangulated with restriction on the maximum area (2 ha) and d) triangulated with restriction on the minimum angle 30°

Polygon	No. Δ	Min area m ²	Max area m ²	Median area m ²	Min angle °
Initial (Figure 5.30(a))	-	-	-	89.35	11
Triangulated without restriction (Figure 5.30(b))	174	0.17	6.618	510	1.6
Triangulated with restriction on the maximum area (Figure 5.30(c))	203	1.58	1.928	437	2.1
Triangulated with restriction on the minimum angle (Figure 5.30(d))	814	0.12	1.417	109	30

Table 5.3. Number of triangles (Δ), area and angles for the element in Figure 5.30

For long and thin elements, generally associated with road segments, the triangulation step is also relevant. Possible options for the triangulation are described in Table 5.4. Figure 5.31 shows an HRU associated with a road segment. As the Mercier subcatchment is rather rural, we use here a .shp example taken in the El Guido subcatchment in Chili. Initially, with the options of Table 5.4, this road segment is represented by an ensemble of nodes (Figure 5.31(a)). Then a triangulation according to various criteria is realized: a triangulation without restriction (Figure 5.31(b)), with restriction on the area that must not exceed 200 m² (Figure 5.31(c)) and with restriction on the angles that must be larger than 30° (Figure 5.31(d)). In a second step, nodes with a maximum distance of 5 m are inserted in the polygon initial boundary and a triangulation without restriction is performed.

Case	Triangulation without restriction (initial nodes)	Long and thin element triangulation with restriction on the maximum area of 200m ² (original nodes)	Long and thin element triangulation with restriction on the minimum angle of 30° (original nodes)	Long and thin element triangulation without restriction (nodes with $d_{max}=5m$)
Input GIS layer to Triangle	test_slim_long_split_dmax_5m	test_slim_long_split_dmax_5m	test_slim_long_split_dmax_5m	test_slim_long_split_dmax_5m
Minimum angle	0	0	30	0
Maximum area	<i>Keep empty</i>	200	<i>Keep empty</i>	<i>Keep empty</i>
Name of the new mesh	test_slim_long_T	test_slim_long_T	test_slim_long_T	test_slim_long_T
Shape descriptor	No shape descriptor	No shape descriptor	No shape descriptor	No shape descriptor
Descriptor threshold	0	0	0	0

Table 5.4. Options for the triangulation of long and thin elements using the TriangleQGIS plugin (to fill out information required in the graphical interface fields)



Figure 5.31. a) HRU associated with a road segment with the original nodes. b) Triangulation without restriction. c) Triangulation with restriction on the maximum area of 200 m^2 . d) Triangulation with restriction on the minimum angle of 30° . e) HRU where nodes are separated with a maximum distance of 5 m (f) and its triangulation without restriction on the area or on the angles. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

Table 5.5 shows the results obtained for each triangulation.

Polygon	No. Δ	Min area m^2	Max area m^2	Median area m^2	Min angle $^\circ$
Initial (Figure 5.31(a))	-	-	-	3,439	71
Triangulated without restriction (Figure 5.31(b))	105	0.07	467	32.7	4.4
Triangulated with restriction on the maximum area (Figure 5.31(c))	218	0.01	201	15.7	21.1
Triangulated with restriction on the minimum angle (Figure 5.31(d))	476	0.01	78	7.2	30
Triangulated with restriction and after insertion of nodes (Figure 5.31(f))	278	0.01	125	12.3	1.2

Table 5.5. Number of triangles (Δ), area and angles for the element in Figure 5.31

The triangulation without restriction of the polygon leads to very long polygons, because the distance between nodes is not controlled by the user (Figure 5.31). Triangulation with restriction on the maximum area (Figure 5.31(c)) and on the minimum angles (Figure 5.31(d)) introduces new nodes inside the polygon that will not lead to good results once these triangles are dissolved according to the Form Factor (see section 5.3.3). The polygon that is triangulated after insertion of nodes on the boundary leads to smaller and equidistant edges, better suited for the subsequent dissolution step according to the Form Factor criterion (section 5.3.3).

For very large elements, the segmentation is realized with a constraint on the maximum area. Figure 5.32 shows an HRU representative of a polygon with large area, 19 ha, that is triangulated according to different options: without any restriction (Figure 5.32(b)), restriction on the maximum area fixed to 2 ha (Figure 5.32(c)) and restriction on the minimum angle fixed to 30° (Figure 5.32(d)). Table 5.6 shows the options that must be chosen to run these various triangulations and Table 5.7 presents the results with statistics associated with each triangulation.

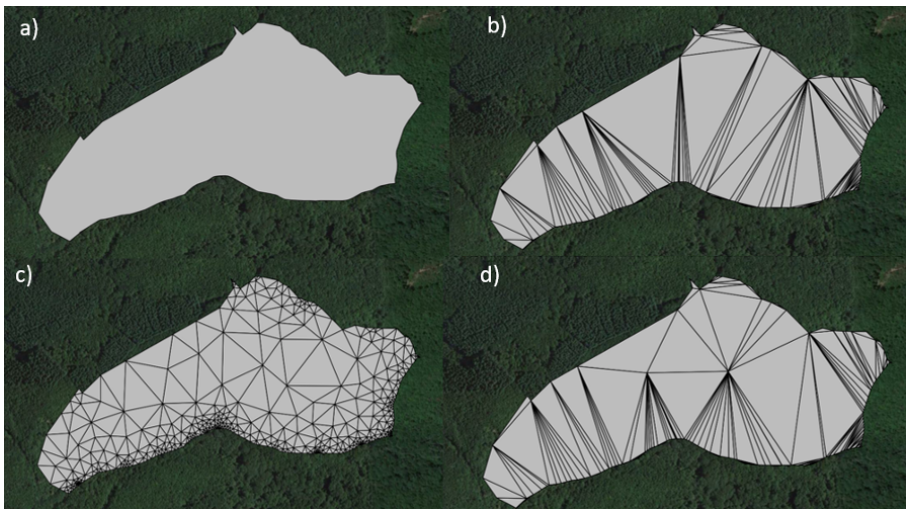


Figure 5.32. a) Very large polygon, b) triangulated without restriction, c) triangulated with restriction on the maximum area (2 ha), and d) triangulated with restriction on the minimum angle 30°

Case	Very large element. No constraint in the triangulation	Very large element. Triangulation with constraint on the maximum area 20,000 m ²	Very large element. Triangulation with constraint of 30° on the minimum angle
Input GIS Layer to Triangle	mercier_big_area_shp	mercier_big_area_shp	mercier_big_area_shp
Minimum angle	0	0	30
Maximum area	Keep empty	20,000	Keep empty
Name of the new mesh	mercier_big_area_t	mercier_big_area_area	mercier_big_area_angulo
Shape descriptor	No shape descriptor	No shape descriptor	No shape descriptor
Descriptor threshold	0	0	0

Table 5.6. Options to generate the triangulation of a very large polygon with the *TriangleQGIS* plugin (to fill out the required fields in the graphical interface)

	No. Δ	Min area m ²	Max area m ²	Median area m ²	Min angle °
Initial (Figure 5.32(a))	-	-	-	192.14	22.8
Triangulated without restriction (Figure 5.32(b))	160	0.45	24.27	1.19	1
Triangulated with restriction on the maximum area (Figure 5.32(c))	164	0.45	19.34	1.16	1
Triangulated with restriction on the minimum angle (Figure 5.32(d))	841	0.09	5,532	228	30

Table 5.7. Number of triangles (Δ), area and angles for the very large element in Figure 5.32

The number of triangles obtained after the triangulation without restriction is 160, with a maximum area of 24,271 m². This decomposition does not fulfill the maximum area restriction of 20,000 m² and it is necessary to insert nodes inside the polygon to generate decomposition with smaller triangles. The triangulation with restriction on the maximum area results in 164 triangles that indeed lead to a maximum area of 2 ha. The triangulation with restriction on the angles leads to more regular polygons but their number increases up to 841 triangles. The most appropriate segmentation is therefore the second one, with restriction on the maximum area.

In summary, from the presented examples, the following recommendations can be made for the various types of badly-shaped elements:

- 1) non-convex HRUs: triangulation without restriction on areas and angles;
- 2) very long and thin HRUs: triangulation without restriction on areas and angles, but starting from a polygon where the distance between boundary nodes is lower than 5 m;
- 3) very large HRUs: triangulation with restriction on the maximum area with a typical value of 2 ha (new nodes are inserted inside the polygon to fulfill this criterion).

GRASS tools to insert nodes:

- v.split

QGIS tools to verify nodes:- Geometrical treatment tools: *Vector > Geometry tools > Node extraction***QGIS tool to perform triangulation:**

- TriangleQGIS plugin

Figure 5.33 shows the triangulation of all the polygons that required it in the Mercier case study subcatchment (see Figure 5.4).

The use of the plugin implemented in QGIS allows us to use the Triangle® software directly on vectorial layers. It can be used to perform a Delaunay triangulation or triangulation with constraints, as shown in the previous figures. Our case study deals with the improvement of a distributed hydrological model mesh, but this plugin can also be used for other types of meshes requiring improvement of the elements' shape.

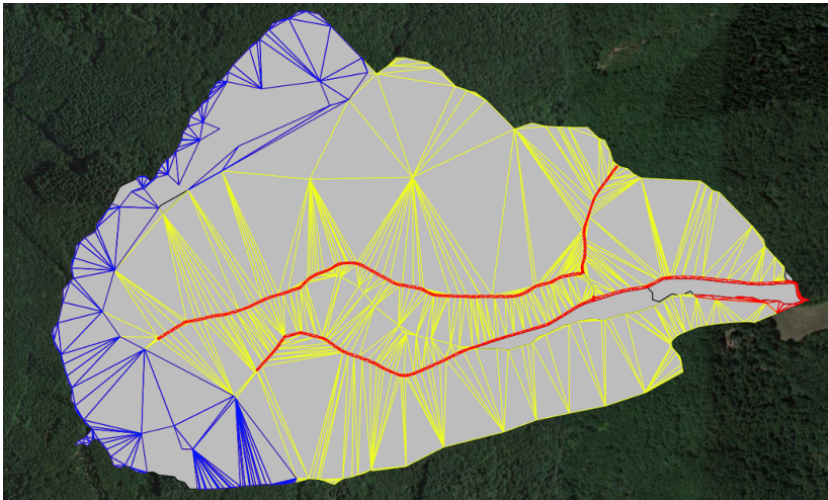


Figure 5.33. Mesh where all the badly-shaped elements have been segmented using the TriangleQGIS plugin

5.3.3. Dissolution of triangulated elements

Once the most appropriate triangulation tool has been selected to segment each element, dissolution of the obtained triangles can be performed using scripts from the Geo-PUMMA toolbox. This dissolution step allows us to rebuild pseudo-regular

polygons that respect convexity and form factor criteria. The final result is therefore closer to the initial polygons, with a smaller number of elements than if all the triangles were kept. This step cannot be performed in QGIS and requires the use of the Geo-PUMMA toolbox and GRASS functions. These steps are however presented for completeness of the presented example. The interested reader can refer to Sanzana *et al.* [SAN 17] and to the Geo-PUMMA User's Manual [GEO 17] for further details.

The script allowing the segmentation of non-convex elements is:

p.B8.a.convexity_segmentation.py

Script:	p.B8.a.convexity_segmentation.py
Inputs:	<ul style="list-style-type: none"> • Mercier_non_convex_triangle layer (clean topology) • Mercier_non_convex layer (pre-Triangle) • A threshold convexity index • A maximum area value
Output:	<ul style="list-style-type: none"> • Mercier_non_convex_processed

Figure 5.34. shows the result of the segmentation using this script.

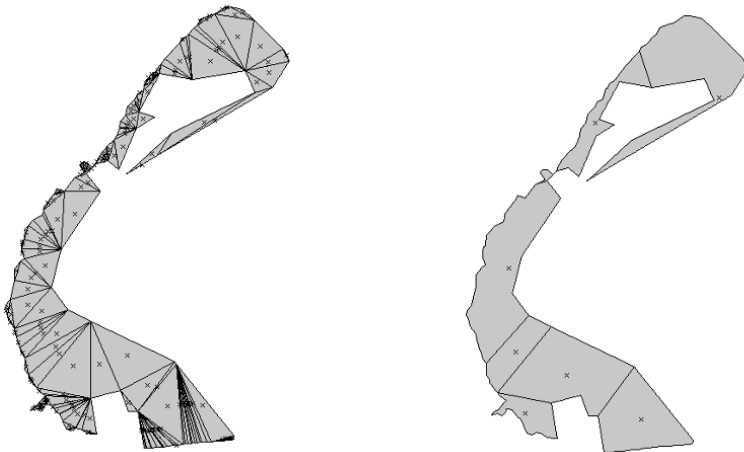


Figure 5.34. Polygons dissolved using a convexity criterion ($CI > 0.75$)

The script allowing the segmentation of elements according to the Form Factor criterion is:

p.B8.b.formfactor_segmentation

Script:	p.B8.b.formfactor_segmentation.py
Inputs:	<ul style="list-style-type: none"> • Mercier_long_elements_triangle layer • A maximum area value • A threshold value for the Form Factor
Output:	<ul style="list-style-type: none"> • Mercier_long_elements_processed

Figure 5.35 shows the result of the segmentation using this script.

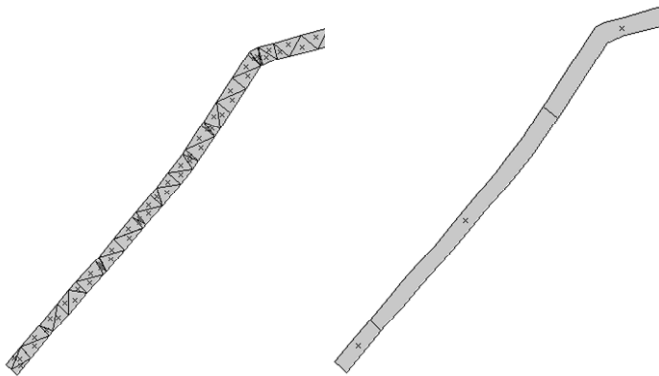


Figure 5.35. Polygon dissolved according to the Form Factor criterion $FF = 0.20$

The script allowing the segmentation of very large elements is:

p.B8.a.convexity_segmentation.py

Script:	p.B8.b.formfactor_segmentation.py
Inputs:	<ul style="list-style-type: none"> • Mercier_big_areas_triangle layer • Mercier_big_areas layer (pre Triangle) • A threshold on the convexity index • A maximum area value
Output:	<ul style="list-style-type: none"> • Mercier_big_areas_processed

Figure 5.36 shows the result of the segmentation using this script.

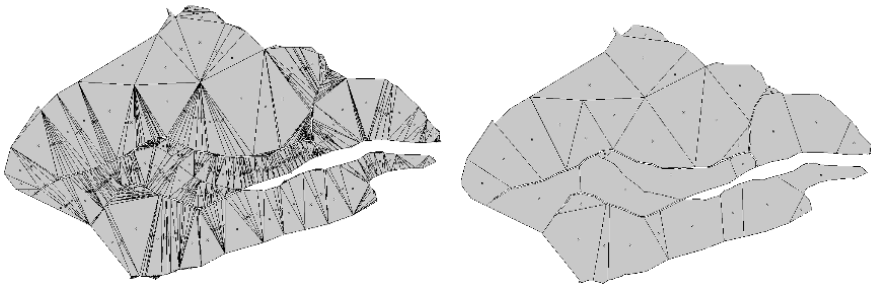


Figure 5.36. Polygon dissolved according to the maximum area criterion $A_{max} = 2 \text{ ha}$

It can be seen that the dissolution dramatically reduces the final number of polygons required to represent HRUs. These elements, once segmented into pseudo-regular elements, allow a routing of the flow of water from each element to the catchment outlet. This routing tree will be satisfactory as illustrated in section 5.3.4.

5.3.4. Effect of the model mesh improvement

In a mesh generated to represent hydrolandscapes, the segmentation procedure will directly affect the hydrological connectivity. The flow routing from one element up to the catchment outlet can be obtained using the *p.olaf.py* script that also belongs to the Geo-PUMMA toolbox [SAN 17]. Flow routing is computed for each HRU and follows the topography toward the lowest neighbor or a river reach. Figure 5.37 shows one example for a subcatchment of the Mercier catchment. The figure shows the initial segmentation and topography contour lines (Figure 5.37(a)), the flow routing using the initial mesh (Figure 5.37(b)), the model mesh after improving non-convex elements only (Figure 5.37(c)) and the model mesh once all the elements requiring it have been improved (Figure 5.37(d)).

Topology problems are solved using the proposed segmentation, even if the final number of elements is increased as compared to the initial mesh. However, the segmentation of non-convex and very thin and long elements with threshold values of 0.75 for the Convexity Index and 0.2 for the Form Factor leads to a good compromise between accuracy and the increase in the number of elements.

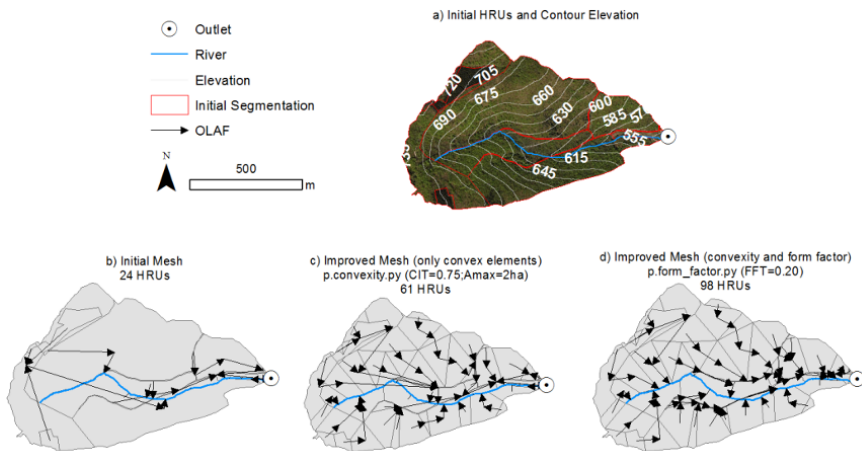


Figure 5.37. a) Subcatchment in the Mercier and topography contour lines. Illustration of the effect of badly shaped elements improvement. b) Extraction of the drainage network using the initial mesh, c) the mesh where only non-convex elements have been improved, and d) mesh where all the badly-shaped elements are segmented. “OLAF” arrows correspond to flow direction between model elements. For a color version of the figure, see www.iste.co.uk/baghdadi/qgis4.zip

5.4. Acknowledgments

The plugin was developed thanks to a program in support of PhD theses at the Vicerrectoría Académica de la Pontificia Universidad Católica. This VRI-UC project led to the production of a manuscript written in Spanish and entitled: “Introducción al Trabajo de Título CC6908: Generación automática de mallas 2D para modelar cuencas periurbanas mediante el concepto de Unidades de Respuesta Hidrológica (URHs) usando QGIS”. This manuscript was written by Sergio Villarroel, student of the University of Chile, under the supervision of Nancy Hitschfeld, from the Computing Sciences Department. The authors were funded by the ECOS-CONICYT C14U02. We also thank the FONDECYT No. 1131131, CEDEUS (FONDAP 15110020), FONDECYT ENL009/15 and IRSTEA-Lyon for their support. Jorge Gironás acknowledges CIGIDEN (FONDAP 15110017). The Mercier catchment is part of OTHU (Observatoire de Terrain en Hydrologie Urbaine). This work was partially developed within the framework of the Panta Rhei Research Initiative of the International Association of Hydrological Sciences.

5.5. Bibliography

- [BOS 12] BOCHER E., MARTIN J.Y., “TAnaTo2: A Tool to Evaluate the Impact of Natural and Anthropogenic Artefacts with a TIN-Based Model”, in BOCHER E., NETELER M. (eds), *Geospatial Free and Open Source Software in The 21st Century*, Springer, Berlin, 2012.
- [FLU 95] FLÜGEL W.A., “Delineating hydrological response units by geographical information system analyses for regional hydrological modelling using PRMS/MMS in the drainage basin of the River Bröl, Germany”, *Hydrological Processes*, vol. 9, nos. 3–4, pp. 423–436, 1995.
- [FUA 17] FUAMBA M., BRANGER F., BRAUD I. *et al.*, “Value of distributed water level and soil moisture data in the evaluation of a distributed hydrological model: Application to the PUMMA model in the semi-rural Mercier catchment (6.6 km²) in France”, *Journal of Hydrology*, 2017.
- [GEO 17] GEO-PUMMA TEAM, Geo-PUMMA Tutorial v.1, available at: <https://forge.irstea.fr/projects/geopumma>, 2017.
- [GIR 10] GIRONÁS J., NIEMANN J.D., ROESNER L.A. *et al.*, “Evaluation of methods for representing urban terrain in stormwater modeling”, *Journal of Hydrologic Engineering*, vol. 15, pp. 1–14, 2010.
- [JAN 11] JANKOWFSKY S., Understanding and modelling of hydrological processes in small peri-urban catchments using an object oriented and modular distributed approach. Application to the Chaudanne and Mercier sub-catchments (Yzeron Catchment, France), PhD Thesis, University of Grenoble, France, available at: <http://tel.archives-ouvertes.fr/tel-00721988>, 2011.
- [JAN 13] JANKOWFSKY S., BRANGER F., BRAUD I. *et al.*, “Comparison of catchment and network delineation approaches in complex suburban environments: application to the Chaudanne catchment, France”, *Hydrological Processes*, vol. 27, no. 25, pp. 3747–3761, 2013.
- [JAN 14] JANKOWFSKY S., BRANGER F., BRAUD I. *et al.*, “Assessing anthropogenic influence on the hydrology of small peri-urban catchments: Development of the object-oriented PUMMA model by integrating urban and rural hydrological models”, *Journal of Hydrology*, vol. 517, pp. 1056–1071, 2014.
- [LAF 06] LAFONT M., VIVIER A., NOGUEIRA S. *et al.*, “Surface and hyporheic oligochaete assemblages in a French suburban stream”, *Hydrobiologia*, vol. 564, pp. 183–193, 2006.
- [LAG 10] LAGACHERIE P., RABOTIN M., COLIN F. *et al.*, “Geo-MHYDAS: a landscape discretization tool for distributed hydrological modeling of cultivated areas”, *Computers & Geosciences*, vol. 36, no. 8, pp. 1021–1032, 2010.
- [MEJ 14] MEJÍA A., DALY E., ROSSEL F. *et al.*, “A stochastic model of streamflow for urbanized basins”, *Water Resources Research*, vol. 50, no. 3, pp. 1984–2001, 2014.

- [PAS 10] PASSALACQUA P., DO TRUNG T., FOUFOULA-GEORGIU E. *et al.*, “A geometric framework for channel network extraction from lidar: nonlinear diffusion and geodesic paths”, *Journal of Geophysical Research*, vol. 115, no. F01002, 2010.
- [ROD 08] RODRIGUEZ F., ANDRIEU H., MORENA F., “A distributed hydrological model for urbanized areas – model development and application to case studies”, *Journal of Hydrology*, vol. 351, nos. 3–4, pp. 268–287, 2008.
- [ROD 13] RODRIGUEZ F., BOCHER E., CHANCIBAULT K., “Terrain representation impact on periurban catchment morphological properties”, *Journal of Hydrology*, vol. 485, pp. 54–67, 2013.
- [RUS 02] RUSS J.C., *The Image Processing Handbook*, 4th edition, CRC Press, Boca Raton, 2002.
- [SAN 16] SANGIREDDY H., STARK C.P., KLADZYK A. *et al.*, “GeoNet: An open source software for the automatic and objective extraction of channel heads, channel network, and channel morphology from high resolution topography data”, *Environmental Modeling and Software*, vol. 83, pp. 58–73, 2016.
- [SAN 17] SANZANA P., GIRONÁS J., BRAUD I. *et al.*, “A GIS-based urban and peri-urban landscape representation toolbox for hydrological distributed modeling”, *Environmental Modelling & Software*, vol. 91, pp. 168–185, 2017.
- [SHE 96] SHEWCHUCK J., “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator”, in LIN M.C., MANOCHA D. (eds), *Applied Computational Geometry: Towards Geometric Engineering*, Berlin, Heidelberg, 1996.
- [SMI 01] SMITH J., LYNN M., MORRISON J. *et al.*, “The Regional Hydrology of Extreme Floods in an Urbanizing drainage Basin”, *Journal of Hydrometeorology*, vol. 3, pp. 267–282, 2001.
- [WMO 08] WMO, Urban flood risk management – a tool for integrated flood management, associated programme on flood management, Technical report, 2008.
- [XIA 07] XIAO Q., McPHERSON E.G., SIMPSON J.R. *et al.*, “Hydrologic processes at the urban residential scale”, *Hydrologic Processes*, vol. 21, pp. 2174–2188, 2007.
- [ZEC 94] ZECH Y., SILLEN X., DEBOURCES C. *et al.*, “Rainfall runoff modelling of partly urbanized watersheds, Comparison between a distributed model using GIS and other models. Sensitivity analysis”, *Water Science and Technology*, vol. 29, nos.1–2, pp. 163–170, 1994.