Using GRASS and PostgreSQL/PostGIS for the development of the automatic preprocessing for a distributed vector-based hydrological model

Authors

- **Sonja Jankowfsky** , IRSTEA, UR HHLY, 69336 Lyon, LUNAM Université, IFSTTAR, GER, 44341, Bouguenais, *France*
- **Pedro Sanzana,** University of Chile, FCFM, Av. Blanco Encalada 2120, Santiago, *Chile*
- **Flora Branger**, IRSTEA, UR HHLY, 69336 Lyon, *France*
- **Isabelle Braud,** IRSTEA, UR HHLY, 69336 Lyon, *France*
- **Yvan Paillé,** IRSTEA, UR HHLY, 69336 Lyon, *France*
- **Florent Brossard,** IRSTEA, UR HHLY, 69336 Lyon, *France*
- **Fabrice Rodriguez,** LUNAM Université, IFSTTAR, GER, 44341, Bouguenais, *France*

KEYWORDS : Distributed hydrological modeling, vector-based model mesh, spatial discretization, preprocessing

Introduction

Many GIS tools have been developed for the preparation of the geographical input data of distributed hydrological models such as the hydrological sciences toolbox of GRASS [1], the raster terrain analysis plug-in of QGIS [2], the Taudem [3] extension of Mapwindow, the terrain analysis hydrology package of SAGA [4], etc. However, most of these preprocessing tools consider raster data, not vector data. Model meshes based on irregular vector geometries are mainly used by [5], [6] and [7]. Lagacherie et al. ([5]) developed the landscape discretization tool GEOMHYDAS in GRASS GIS, which allows the construction of vector-based HRU maps (Hydrological Response Units) based on a selective overlay of property layers such as land use, soil maps or sub-basins. Furthermore, special tools were developed to integrate man-made features such as ditches, roads and agricultural fields. Branger [6] simply used a land use map as model mesh. Rodriguez et al. [7] developed the preprocessing of the urban hydrological model URBS in MapInfo. Instead of HRUs, URBS uses Urban Hydrological Elements (UHEs) as model units, each consisting of one cadastral unit and half of the adjoining street.

In this paper, we go one step further by combining the HRU model mesh with UHEs in order to create a model mesh adapted for peri-urban areas. This particular model mesh needed the development of several scripts for the extraction of the flow routing in a mixed urban and rural environment. In order to obtain realistic flow paths, mesh optimization methods were developed treating concave or too large polygons or polygons with holes. Furthermore, scripts were developed allowing the integration of raster information (e.g. slope) into the model mesh without creating a large number of small polygons.

Software requirements

Following the experience of Branger et al. [8], who tested different open-source GISs such as OpenJump, SAGA, OrbisGIS, Mapwindow, PostGIS, QGIS and GRASS for the preprocessing of distributed hydrological models built within the LIQUID modeling framework, we chose GRASS GIS (http://grass.fbk.eu/) because of its topological data structure and functions for the development of the automatic preprocessing. However, some tasks, especially concerning the flow routing, have been shown to be easy to develop using PostGIS functions embedded in SQL queries and more complicated to develop using GRASS functions. Therefore, GRASS functions and PostgreSQL queries were integrated into Python scripts using the Pygresql and Grass libraries.

Developed method

UHE and HRU delineation

In a first step, the preprocessing developed by Rodriguez et al. [7] in MapInfo

for urban zones (UHEs) was transformed into GRASS-Python scripts. These scripts [9] allow additionally to create the real UHE geometry by dividing the road in halves and merging it with the adjoining cadastral unit.
In rural areas, the HRUs can be created with the m.seg/m.dispolyg GRASS script [5] using different polygon property layers and the drainage network as input data. The UHEs and HRUs are then overlayed using GRASS v.overlay.

In order to integrate raster properties, the slope segmentation GRASS-Python script was developed [10]. The script takes as input the HRU vector map, a slope raster map and a threshold value and extracts all polygons for which the standard deviation of the slope is larger than the threshold value. The extracted polygons are divided along two morphological boundaries defined by the inter quartile range. This range, defined as the difference between the third quartile (Q3) and the first quartile (Q1), allows the identification of homogeneous classes (Figure 1). The intersection lines are smoothed using the Snakes and Douglas-Peucker algorithm implemented in GRASS GIS, see Figure 1. The script works also with any raster property in addition to slope, such as elevation, aspect, etc.

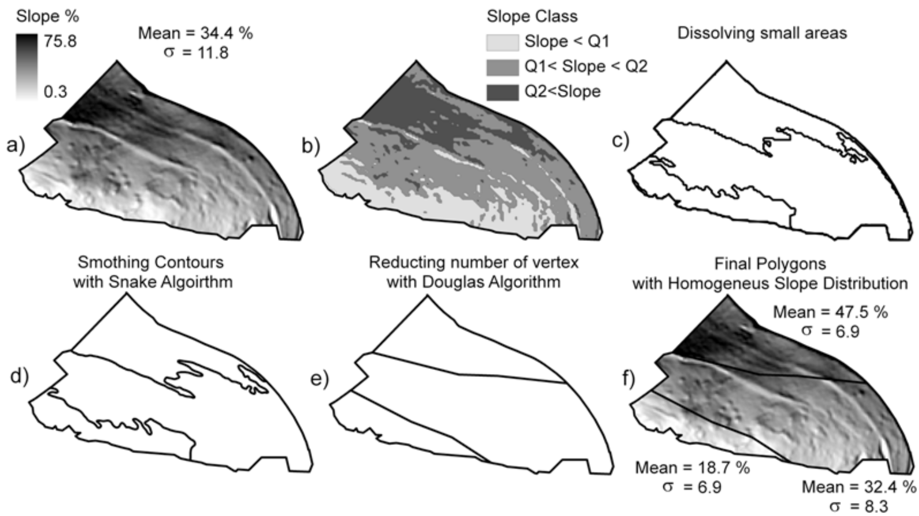

FIGURE 1

Segmentation of a polygon using homogenous slope classes [10]

Three scripts were developed in order to adapt the model mesh to numerical constraints [10] :

1. The polygon_hole GRASS-Python script divides polygons with islands along the nearest and farthest distance from the island to the surrounding polygon.
2. The convexity segmentation script extracts concave polygons for which the convexity index (perimeter of a convex polygon with the same area/concave polygon) exceeds a certain threshold value and partitions them using a C++ triangle algorithm [11]. The triangle algorithm was preferred to the v.delaunay triangulation available in GRASS as it creates larger and less long-drawn polygons. In order to reduce the number of model units and to keep the object oriented character of the irregular model mesh, the triangles were re-unified with a GRASS-Python script under the constraint of a convexity threshold.
3. The area segmentation script is similar to the convexity segmentation script, but uses a maximal polygon area as threshold value.

Flow transfer and routing

For the peri-urban areas typical flow transfer, three principles were applied:

1. The overland flow paths follow the topography towards the lowest neighbor model unit.
2. The surface runoff on impervious areas is intercepted by sewer pipes. Therefore, the flow transfer from the UHEs to the closest drainage network is direct.
3. The lateral subsurface flow depends on the water table level and can be multi-directional towards all neighbors.

Three PostgreSQL/GRASS-Python scripts were developed for the determination of these flow transfer paths [12]. Scripts 1 and 2 are based on recursive functions, which extract single flow directions between model units. The overland flow paths are determined sub-basin by sub-basin, whereas the urban surface runoff connections are independent of the sub-basins. The polygon-polygon neighborhood relations in script 3 where extracted with GRASS functions. However, PostgreSQL queries had to be used to extract the neighborhood relations of model units (polygons) and river reaches (lines). For this, the polygons were transformed into boundaries. A small buffer around the centroid of the boundaries was then intersected with the river reaches in order to determine only the adjacent river reach.

Concerning the drainage network three scripts were developed :

1. The script river_direction orientates all network reaches in upstream direction.
2. The script num_river numbers all network reaches starting from the outlet.
3. The script river_h_s calculates the mean altitude and slope per network reach.

Results

Thanks to the developed scripts, the model meshes and flow routing for two small catchments (Chaudanne and Mercier, respectively 4.1 and 6.8 km^2) in the peri-urban area of Lyon city, France, could be determined as shown in Figure 2 for the Chaudanne catchment. On this catchment, we obtained 2945 HRUS with average area of 1394 m^2 and 522 drainage reaches.

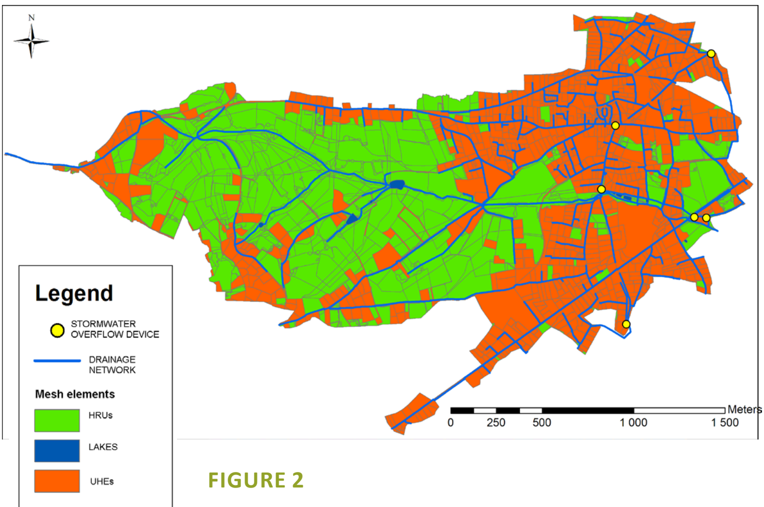

FIGURE 2

Final model for the Chaudanne catchment consisting of UHEs (red) and HRUs (green)

The mesh optimization improved considerably the flow paths and the area difference between the model units, as can be seen in Figure 3. Before the mesh optimization (Figure 3a) some of the flow paths cross the river network due to large and concave polygons, which could be corrected by means of the mesh optimization (Figure 3b).

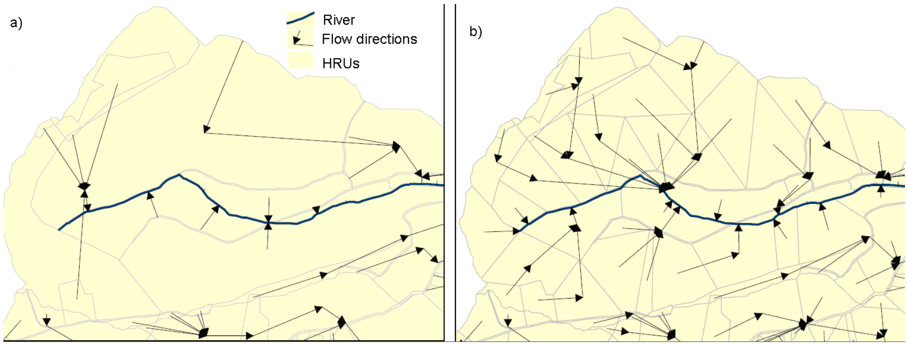

FIGURE 3

Determined overland flow paths for a) the non-optimized mesh and for b) the optimized mesh

Encountered problems

The study revealed that not all tasks can be completed automatically. Especially polygon-line relations are not well represented in current GIS software and they often need manual treatments. Furthermore, no open-source GIS function could be found determining the middle street line. Therefore, we corrected manually the street line vectors from BD© Topo (IGN). A proper vector topology is necessary for most of the steps. We solved this problem by applying the cleaning functions of GRASS (v.clean) to the input layers and to the output layers of the major preprocessing steps.

Conclusions and perspectives

This new model mesh combining UHEs and HRUs as irregular vector geometries is promising for object oriented modeling as the catchment area is composed of many single landscape objects. Jankowfsky [13] used these model meshes to apply the object-oriented Peri-Urban Model for Landscape Management (PUMMA). The scripts are open-source and can also be applied for other hydrological models.

Nevertheless, the irregularity of the model mesh as well as the different kind of drainage in peri-urban catchments (overland flow, pipe drainage, subsurface flow) reveal many questions concerning the runoff routing. Some of these problems, such as the correction of concave or too large polygons, as well as the integration of slope information were treated in this study. Other problems involving long polygons (e.g. roads or hedgerows) acting as flow barriers during the calculation of the flow routing still need to be addressed. A possibility would be to use the convexity segmentation script and replace the convexity index as threshold with a compactness index. Furthermore, segmentation with altitude classes similar to the slope segmentation would improve the flow routing, which is governed by the altitude differences of adjacent model units.

The developed scripts should be available soon for download at [14] and can be obtained from the authors in the meantime. In order to run the scripts, PostgreSQL 8.3/PostGIS (pginstaller [15]), GRASS [1] and PyGreSQL 4.0 [16] have to be installed.

[1] GRASS development team, 2012. GRASS Wiki: Hydrological sciences. Retrieved August 30, 2012 from http://grass.osgeo.org/wiki/Hydrological_Sciences

[2] Quantum GIS development team, 2012, Quantum GIS. Retrieved August 30, 2012 from http://www.qgis.org

[3] David Tarboton Hydrology Research group Utah State University, 2012, Terrain Analysis Using Digital Elevation Models (TAUDEM). Retrieved August 30, 2012 from http://hydrology.usu.edu/taudem/taudem5.0/index.html

[4] SAGA development group, 2012. SAGA-GIS, Terrain Analysis - Hydrology. Retrieved August 30, 2012 from http://sourceforge.net/apps/trac/saga-gis/wiki/ta_hydrology

[5] LAGACHERIE, P., RABOTIN, M., COLIN, F., MOUSSA, R., and VOLTZ, M. Geo-mhydas: A landscape discretization tool for distributed hydrological modeling of cultivated areas. *Computers & Geosciences*, 36(8) (2010), 1021 – 1032.

[6] BRANGER, F. Utilisation d'une plate-forme de modélisation environnementale pour représenter le rôle d'amenagements hydro-agricoles sur les flux d'eau et de pesticides. Application au bassin versant de la Fontaine du Theil (Ille-et-Vilaine). PhD thesis, INP- Grenoble.2007.

[7] Rodriguez, F., Andrieu, H., and Creutin, J. Surface runoff in urban catchments: morphological identification of unit hydrographs from urban databanks. *Journal of Hydrology*, 283 (2003), 146–168.

[8] BRANGER, F., JANKOWFSKY, S., VANNIER, O., VIALLET, P., DEBIONNE, S., BRAUD, I. Use of open-source GIS and data base software for the pre-processing of distributed hydrological models, *Geospatial free and open source software in the 21rst century*, Bocher, E., Neteler, M. (Eds), Lecture notes in Geoinformation and Cartography, 2011, 12 pp.

[9] PAILLÉ, Y. Automatisation du prétraitement des données spatiales pour la modélisation hydrologique distribuée en zone périurbaine à l'aide du logiciel libre GRASS. 2010. Master thesis, Université de Nantes.

[10] Sanzana, P., JANKOWFSKY, S., BRANGER, F., BRAUD, I., VARGAS, X., HITSCHFELD, N., GIRONÁS, J. Automatic mesh optimization based on Hydrological Response Units for object-oriented distributed hydrological modeling, 2012, submitted to Computers and Geosciences.

[11] SHEWCHUCK, J. Applied Computational Geometry: Towards Geometric Engineering, volume 1148, chapter Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, 1996, pages 203–222.

[12] BROSSARD, F. Automatisation du pré-traitement des données spatiales pour la modélisation hydrologique distribuée en zone périurbaine. 2011. Rapport de stage, EPMI Cergy.

[13] 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, Université de Grenoble, 2011.