

# A Comparison of the Performance of Digital Elevation Model Pit Filling Algorithms for Hydrology

N. Senevirathne<sup>a</sup> and G. Willgoose<sup>a</sup>

<sup>a</sup> School of Engineering, Faculty of Engineering and Built Environment  
The University of Newcastle, NSW, Australia  
Email: [c3152807@uon.edu.au](mailto:c3152807@uon.edu.au)

**Abstract:** Digital elevation models (DEM) are widely used in hydrological applications for computing useful topographic parameters such as slope, flow direction, flow accumulation area and stream channel network. However, DEMs generally contain numerous topographic depressions which are real and/or artifactual. These depressions can take the form of single cells (pits) or contiguous areas in DEM (depressions). The problem with these pits and depressions is that they interrupt continuous flow paths in DEMs. To avoid these problems, all pits have to be rectified and create a depressionless DEM before calculating flow directions or any related topographic parameters.

Agency provided DEMs may be pit filled, but pits can also be generated while interpolating DEMs for changing grid spacing (e.g. LiDAR data). Therefore pit filling is an essential requirement for any hydrological study. A number of algorithms have been developed over the past few decades to treat pits in DEMs. With the availability of high resolution data, DEMs typically contain millions of cells which increase file sizes and computational effort. Therefore, efficiency of pit filling algorithms has to be taken into account when using such DEMs in hydrological applications.

In this paper, two of most widely used pit filling algorithms (Jenson and Domingue, 1988 and Planchon, 2001) are compared in terms of their performance and ability to extract topographical parameters. One second DEM derived from Shuttle Radar Topography Mission (SRTM) data was used in the study. Two study areas were used in the comparison. The first study area is comprised of three catchments located in Eastern Australia and they were used for evaluating topographic changes made by pit filling algorithms. The second study area is located in East coast and it was used to compare the performance of the two algorithms across a regional extent. According to the results, both algorithms behave similarly in modifying existing topography, but calculated flow accumulations and drainage networks were slightly different from each other. When the filled area was relatively small, both algorithms have resulted in similar flow paths. But in relatively large filled areas, they have resulted in unrealistic parallel flow paths significantly different from each other.

Both algorithms were implemented in the Python programming language to provide a common platform for comparison. Python is an interpreted language and the Cython tool has been recently developed to convert Python code to C code and allow it to be compiled. Cython was used to convert Python code to C extensions and the performance of both Python and Cython versions were evaluated. Time taken to execute pit filling algorithms on different sizes of DEMs was measured. The execution time of the Planchon algorithm showed a linear relationship with the size of the DEM while execution time of Jenson algorithm increased exponentially. Moreover, performance of both algorithms was evaluated for different resolutions, on a constant grid extent. The execution time of the Jenson algorithm increased with DEM resolution and showed a direct relationship to the total number of pits. However, execution time of Planchon algorithm remained almost constant regardless of the total number of pits. In Cython, both Jenson and Planchon algorithms showed significant improvement in execution time, relative to implementation in Python.

**Keywords:** Pit filling, Digital elevation model, Python, Cython

## 1. INTRODUCTION

Digital Elevation Models (DEMs) are widely used in hydrological applications. They are used to extract topographic parameters such as slope, aspect, flow directions (O'Callaghan and Mark, 1984, Tarboton, 1997), contribution areas (Tarboton, 1997) and channel networks. A variety of algorithms have been developed to extract and determine the parameters above. Most of them require an initial flow direction grid that is determined based on elevation gradient measured outwards from a grid cell. A major problem in determining flow directions is presence of closed depressions within DEMs where the maximum outward elevation gradient is zero or negative (Pan et al., 2012). These take the form of single cells lower than all their neighbours (referred to as pits), or contiguous areas within the DEM which again are lower than all of their neighbours (referred to as depressions) (Arnold, 2010). For convenience, both pits and depressions will be referred to as 'pits' for the rest of the paper.

The problem with these pits is that they interrupt continuous flow paths in DEMs. Water modelled as flowing into a pit will be trapped inside the pit, resulting in underestimates of the contribution area for downslope pixels. If pits lie inside a channel network, the extracted channel network will be discontinuous (Pan et al., 2012). To avoid these problems, all pits have to be rectified to create a depressionless DEM before calculating flow directions or any other topographic parameters.

Usually all DEMs contain pits. They could be either real depressions in the topography or artifact depressions that do not represent actual features of the landscape. However, there is no way to determine from the DEM which pits are real and which are artifacts (Arnold, 2010). Artifact pits may occur in DEMs because of: (1) the limited horizontal and vertical resolution of elevation data; (2) data errors; (3) error resulting from the interpolation of elevation data to generate a DEM (Lindsay and Creed, 2005). Although there are DEMs with finer resolution and greater accuracy, artifact depressions will always be present in DEMs because they are an inherent characteristic of the tessellation of a continuous surface (Mark, 1988).

Though pit filled DEMs are readily available nowadays, pit filling methods may be useful in later stages of DEM processing. For example, pits can also be generated while interpolating DEMs for changing grid spacing (e.g. LiDAR data). In fact, resampling a DEM to increase the resolution (downscaling) leads to more pits while decreasing the resolution (upscaling) reduces the total number of pits but the total percentage area covered by pits increases (Grimaldi et al., 2007).

DEMs are derived from the actual ground surface using a variety of methods (e.g. photogrammetry, LiDAR). As these methods have increased in precision and accuracy, finer resolution DEMs become available. As a result of the increased precision and file sizes, many of the hydrologic preprocessing and analysis techniques for coarser resolutions and smaller DEMs become time consuming when applied to high resolution data (Tarboton et al., 2009). With such DEMs, runtime efficiency of pit filling algorithms is becoming a more important issue.

## 2. PIT FILLING METHODS

A number of algorithms are available to deal with pits in DEMs. They can be categorized into three main classes according to their approach of rectifying pits (Poggio and Soille, 2012).

1. Incremental methods: fill pits by increasing their elevation value until their lowest pour point is reached. (Jenson and Domingue, 1988, Planchon and Darboux, 2001)
2. Decremental methods: where values along a path starting from the bottom of the pit and reaching a pixel of lower elevation value are decremented by setting their elevation value to that of the bottom of the pit. The two main decremental methods are known as the breaching (Martz and Garbrecht, 1998) and carving (Soille et al., 2003).
3. Hybrid methods: combining incremental and decremental methods. (Lindsay and Creed, 2005)

Two widely used incremental filling algorithms (Jenson and Domingue, 1988, Planchon and Darboux, 2001) are compared in this paper in terms of their performance and ability to extract topographical parameters.

The pit filling algorithm of Jenson and Domingue (1988) is possibly the most widely used (Arnold, 2010). The algorithm works in two stages. In the first stage, cells contained in depressions are raised to the lowest elevation value on the rim of the depression. This procedure converts every depression into a flat area where at least one external cell has a lower elevation than the flat (Figure 1a). The second stage assigns flow directions to the cells in flat areas based on the shortest flow path to its pour point. Jenson and Domingue algorithm is implemented in many GIS and hydrological software (for example ArcGIS (ESRI, 2008) and GRASS (Neteler and Mitasova, 2008)).

Planchon and Darboux (2001) used a different approach in their algorithm. Instead of gradually filling the depressions, it first inundates the surface with a thick layer of water and then removes the excess water, working inwards from the edges (figure 1b). The algorithm has a time complexity of  $N^{1.2}$  (where  $N$  is the number of grid points). Therefore, it can process large DEMs with an acceptable time. Moreover, this algorithm provides two options; pits can be filled with a surface either strictly horizontal or slightly sloping. The first option is used for the calculation of depression storage capacity and the second one for drainage network extraction (Planchon and Darboux, 2001).

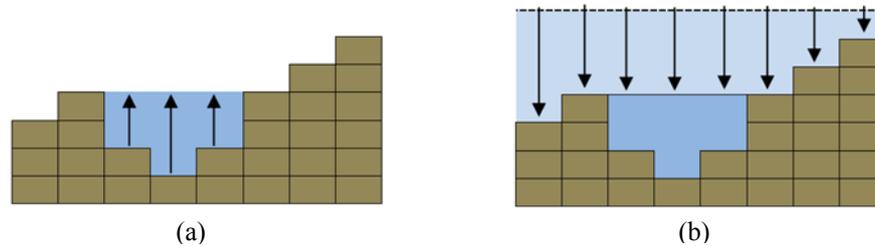


Figure 1. Filling approaches of the algorithms a) Jenson and Domingue b) Planchon and Darboux

For convenience, algorithms of Jenson and Domingue (1988) and Planchon and Darboux (2001) will be referred to as ‘Jenson algorithm’ and ‘Planchon algorithm’ respectively for the rest of the paper.

### 3. STUDY AREA AND DATA USED

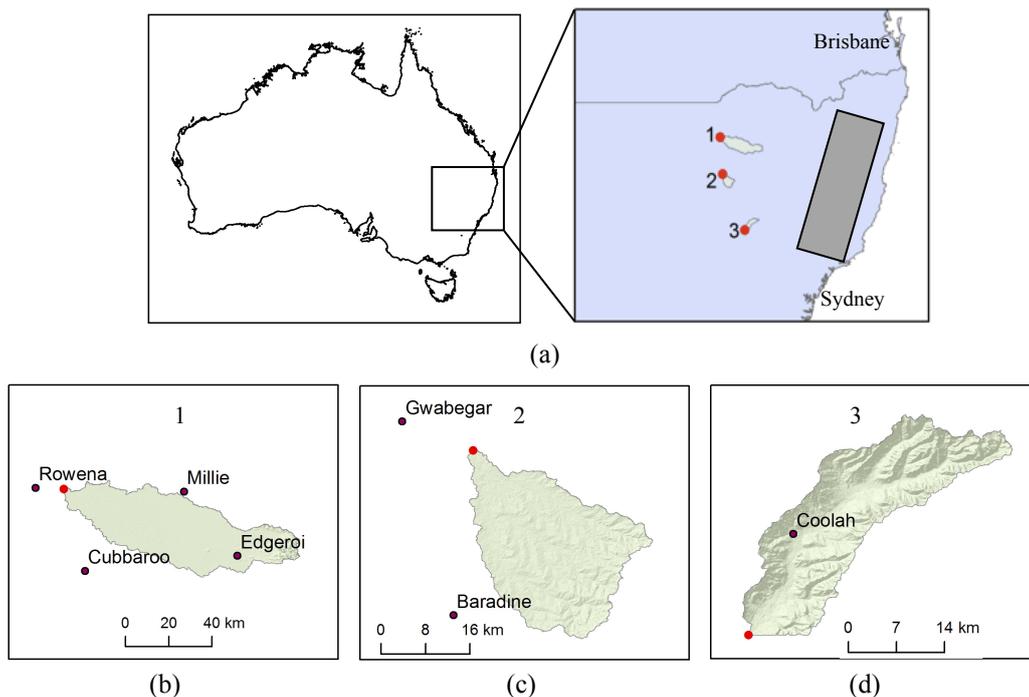


Figure 2. Study area. (a) Overview of three catchments and the region (shaded rectangle) used to compare performance of the algorithms, (b) Catchment 1, (c) Catchment 2, (d) Catchment 3.

One arc second DEM derived from Shuttle Radar Topography Mission (SRTM) data (DEM-S) were used in the study (Gallant et al., 2011). The DEM represents ground surface topography, and where possible, excludes other features such as vegetation and man-made structures. They have been adaptively smoothed to reduce random noise typically associated with the SRTM data in low relief areas.

Two study areas were used in the comparison. One was for evaluating topographic changes made by pit filling algorithms (catchments 1,2 and 3 in the Figure 2) and the other one was for comparing the

performance of the two algorithms (shaded rectangle in the Figure 2a). Choosing two study areas from inland and coastal area provides a representative sample of the DEM data set.

The first study area is comprised of three watersheds located in Eastern Australia (Figure 2b, 2c and 2d). These were used for evaluating topographic changes (described in section 4.1). Basic statistics of these catchments are provided in the Table 1.

Table 1. Statistics of the catchments

	Catchment 1	Catchment 2	Catchment 3
Area (km <sup>2</sup> )	2744.1	640.6	455.6
Elevation range: Min - Maximum (m)	155.7 – 1306.0	234.9 – 471.1	429.1 – 1100.3
Mean Elevation (m)	217.6	331.4	651.3
Slope range: Min - Maximum (degrees)	0 – 65.3	0 – 9.7	0 – 48.9
Mean Slope (degrees)	0.8	1.4	7.7

The second study area is located on the East coast (shaded rectangle in Figure 2a) and several DEM tiles from that area were used to compare the performance of two algorithms (described in section 4.2 and 4.3). The area comprises a narrow strip of land along the eastern coast and complex of hills toward the western side.

Basic analysis has been done in order to identify pits in both study areas and it has been found that the DEM data contain 0.4%-0.5% of pits by area for these particular study areas.

#### 4. METHODS AND RESULTS

##### 4.1. Changes in Topography

All three catchments have been filled separately using the two pit filling algorithms. A small minimum gradient (0.001) was used in Planchon algorithm. Since both algorithms are incremental methods (fill up to the lowest elevation value on the rim of the depression), final results were almost identical. Using a small minimum gradient (0.001) in Planchon algorithm has resulted in a slight slope over the filled area. However, it does not affect the total amount of pixels changed during the pit filling. The only modifications occurred in elevation and slope.

Table2. Mean elevation and mean slope (degrees) of filled catchments

			Whole Catchment			Filled area		
			Original	Filled (Planchon)	Filled (Jenson)	Original	Filled (Planchon)	Filled (Jenson)
Catchment 1	Elevation	Mean	217.6	217.7	217.7	182.3	182.7	182.7
		STD	86.6	86.6	86.6	25.9	25.8	25.8
	Slope	Mean	0.784	0.747	0.746	0.245	0.0002	0.0
		STD	2.956	2.960	2.960	0.302	0.0000	0.0
Catchment 2	Elevation	Mean	331.4	331.4	331.4	293.0	293.5	293.5
		STD	49.9	49.9	49.9	34.7	34.7	34.7
	Slope	Mean	1.377	1.348	1.349	0.656	0.0001	0.0
		STD	0.860	0.875	0.879	0.492	0.0000	0.0
Catchment 3	Elevation	Mean	651.3	651.3	651.3	519.8	520.5	520.5
		STD	147.7	147.7	147.7	115.4	115.4	115.4
	Slope	Mean	7.718	7.696	7.701	1.155	0.0001	0.0
		STD	6.528	6.533	6.541	1.327	0.0001	0.0
STD = Standard Deviation								

According to the results shown in Table 2, the overall topography of the catchments has been slightly changed. Compared to the size of the catchments and range of the elevations, the percentage changed is quite small. For example, the change in the mean slope is less than 0.04 degrees for all catchments.

Neither pit filling algorithm alters existing topography significantly. However, the different filled surfaces produced by the algorithms (Jenson algorithm results in a flat surface while Planchon algorithm results in a slightly slope surface) can affect the extracted topographic parameters. For example, flow direction grids can be somewhat different on DEMs filled by these algorithms. This can lead to different flow accumulations and drainage networks.

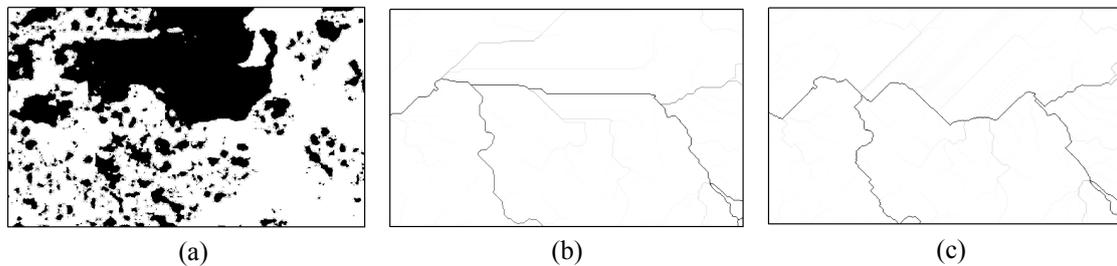


Figure 3. Flow routing over a pit filled region in catchment 1 (a) Filled area (540 x 360 pixels) (b) drainage network of the region filled by Jenson algorithm (c) drainage network of the region filled by Planchon algorithm

Figure 3 shows an example of flow routing over a pit filled region in catchment 1 (540 x 360 pixels). Flow accumulations and drainage network were calculated in ArcGIS software. Both results show unrealistic parallel flow paths in the filled area. Planchon algorithm also creates parallel flow paths, but some of them are perpendicular to each other. In the lower portion of figure 3(a) where the filled area is relatively small, both algorithms have resulted in similar flow paths.

#### 4.2. Performance of algorithms in Python platform

Python is an open source, interpreted, object-oriented, high-level programming language with dynamic semantics. Because code is automatically compiled to byte code and executed, Python is suitable for use as a scripting language.

Both pit filling algorithms are available in different software and different programming languages. The problem is that they do not provide a fair representation of algorithms' original behaviour. The same algorithm might yield different computational times depending on the programming language used. For instance, low level programming languages are typically much faster than interpreted languages. Moreover, one can use additional programming optimization techniques to make the code faster. These biases make it difficult to use available versions of the algorithms for a comparison.

Therefore, both pit filling algorithms were written in Python in order to provide a common platform for comparison. When developing algorithms in Python, considerable effort has been made to follow Jenson's and Planchon's original codes. Matlab code provided by Planchon was used to develop his algorithm and guidelines provided by Jenson and Domingue (1988) were used in developing Jenson algorithm. Hence, it can be assumed that Python algorithms used in this study represent their original code as closely as possible.

The time taken to execute pit filling algorithms on different size of DEM subsets is given in Figure 4. The total number of pixels in the DEM is given on the x-axis and computational time is given on the y-axis. For example, a DEM subset with 1000 rows x 1000 columns will have 1 million pixels. Total execution time was

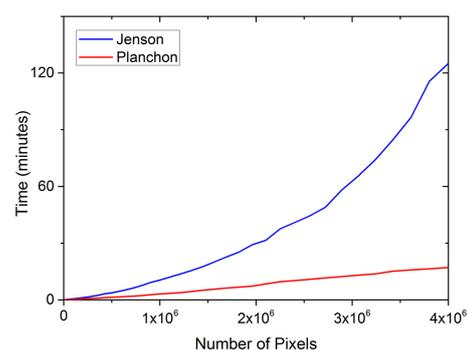


Figure 4. Execution time of pit filling algorithms on different size of DEM subsets

measured for 40 different DEM subsets, ranging from 50x50 (2500 pixels) to 2000x2000 (4 million). This procedure was repeated on multiple DEM sites and the average times were used for analysis.

The Planchon algorithm shows a robust and superior performance for all DEMs. Its execution time linearly increases with the size of the DEM. The longest time is less than 17 minutes for 4 million pixels (fig 4). The execution time of Jenson algorithm shows an exponential increase with the size of the DEM. This is probably caused by the complexity of the pits. In Jenson algorithm, embedded pits are merged because they flow in to each other. This step is the most costly part of the whole procedure, so increases the execution time (Planchon and Darboux, 2001).

The performance of both algorithms was evaluated for different resolutions, ranging from 1 arc second to 5 arc seconds. 1 second original dataset were resampled into 2, 3, 4 and 5 seconds resolutions using nearest neighbor method in ArcGIS software. The execution time of both pit filling algorithms is mostly determined by two variables: 1) the size of the DEM and 2) the number of pits. In order to observe any effect caused by DEM resolution, both these variables need to be constant. Since it is impractical to make both variables constants at the same time, only a constant grid extent was chosen (in this case, 720 rows x 720 cols). Even though the grid size is same (720x720), their spatial extents are different due to different resolutions. Therefore, the total number of pits can vary from one DEM to another. Figure 5a shows the average number of pits at different resolutions. According to the results (Figure 5b), the execution time of Planchon algorithm remains almost a constant regardless of the total number of pits. Its time varies between 69-79 seconds for all of the different DEMs. However, the execution time of the Jenson algorithm increases with the DEM resolution and shows a direct relationship with the total number of pixels. Apparently, total execution time of Jenson algorithm is determined by both total number of pits and DEM size.

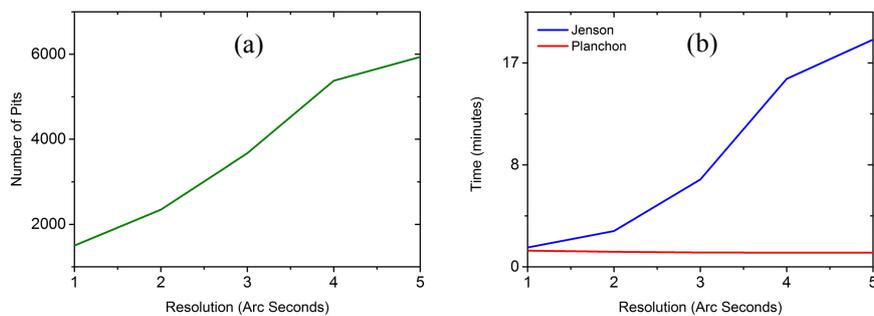


Figure 5. Performance of algorithms for different resolutions (a) average number of pits on different resolutions (b) execution time of pit filling algorithms for different resolutions on a constant grid extent (720 rows x 720 cols)

### 4.3. Performance in Cython

Although Python is fast enough for many tasks, low-level computational code written in Python tends to be slow, largely due to Python’s extremely dynamic nature (Behnel et al., 2011). In order to solve this problem, the Cython tool has been recently developed to convert Python code to C code and allow it to be compiled. Because it uses C types, Cython makes it possible to embed numerical loops, running at C speed, directly in Python code.

Both Python algorithms were converted into Cython by adding C declarations in the Python code. Then the performance of the Cython versions was evaluated. Both Jenson and Planchon algorithms showed significant improvement in execution time (Figure 6). Planchon algorithm shows 8.4 times speedup, while Jenson algorithm shows 9.8 times speedup.

During this study, it was found that module level C type declarations still have some Python calls during execution and tend to be slow. Function level C type declarations are much faster than module level declarations. Once they have been passed into the function argument, overall execution time was significantly improved.

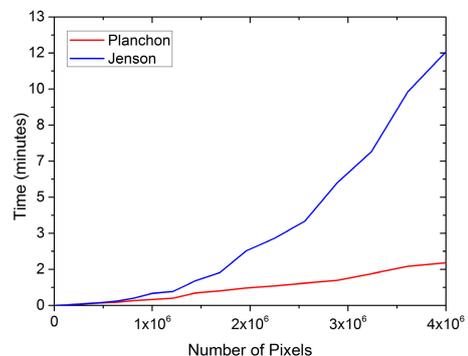


Figure 6. Execution time of pit filling algorithms on different size of DEM subsets in Cython (Compared with Figure 4)

## 5. CONCLUSION

Much improved and optimized versions of Jenson and Domingue (1988) and Planchon and Darboux (2001) algorithms are now available. This study provides insights into their original algorithms and behaviour.

Even though both algorithms perform similarly in terms of altering topography, a clear difference can be seen in computational times. Jenson algorithm is very efficient in small DEMs, but dramatically decreases in its performance when it comes to large DEMs. Compared to the Jenson algorithm, Planchon algorithm is more efficient in all tested cases in the study. Moreover, Planchon algorithm has the added advantage of filling flats with a slight slope which is convenient for calculating flow directions. Furthermore, large array manipulations in Python tend to be slow. Cython can be used to gain a significant speedup using small modifications to the Python code.

Algorithms used in this study can be made available to interested parties on request.

## REFERENCES

- Arnold, N. (2010). A new approach for dealing with depressions in digital elevation models when calculating flow accumulation values. *Progress in Physical Geography*, 34, 781-809.
- Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S. & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering*, 13, 31-39.
- Gallant, J.C., Dowling, T.I., Read, A.M. Wilson, N., Tickle, P. and Inskeep C. (2011) 1 second SRTM derived digital elevation models user guide. Geoscience Australia
- Grimaldi, S., Nardi, F., Benedetto, F. D., Istanbuluoglu, E. & Bras, R. L. (2007). A physically-based method for removing pits in digital elevation models. *Advances in Water Resources*, 30, 2151-2158.
- Environmental Systems Research Institute (ESRI) (2008) ArcGIS 9.2 Desktop Help. Available at: <http://webhelp.esri.com/arcgisdesktop/9.2>
- Jenson, S. & Domingue, J. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, 54, 1593-1600.
- Lindsay, J. B. & Creed, I. F. (2005). Removal of artifact depressions from digital elevation models: towards a minimum impact approach. *Hydrological Processes*, 19, 3113-3126.
- Mark, D. (1988). Network models in geomorphology. *Modelling Geomorphological Systems*. New York: JohnWiley.
- Martz, L. W. & Garbrecht, J. (1998). The treatment of flat areas and depressions in automated drainage analysis of raster digital elevation models. *Hydrological processes*, 12, 843-855.
- Neteler, M. & Mitasova, H. (2008). Open Source GIS: A GRASS GIS Approach, 3rd Edn., *The International Series in Engineering and Computer Science: Vol. 773*, Springer, New York.
- O'Callaghan, J. F. & Mark, D. M. (1984). The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing*, 28, 323-344.
- Pan, F., Stieglitz, M. & Mckane, R. B. (2012). An algorithm for treating flat areas and depressions in digital elevation models using linear interpolation. *Water resources research*, 48.
- Planchon, O. & Darboux, F. (2001). A fast, simple and versatile algorithm to fill the depressions of digital elevation models. *Catena*, 46, 159-176.
- Poggio, L. & Soille, P. (2012). Influence of pit removal methods on river network position. *Hydrological Processes*, 26, 1984-1990.
- Soille, P., Vogt, J. & Colombo, R. (2003). Carving and adaptive drainage enforcement of grid digital elevation models. *Water Resources Research*, 39.
- Tarboton, D., Watson, D., Wallace, R., Schreuders, K. & Tesfa, T., (2009), Hydrologic terrain processing using parallel computing. *AGU Fall Meeting Abstracts*, 2009. 0867.
- Tarboton, D. G. (1997). A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water Resources Research*, 33, 309-319.