G-GCS – A Geographical Aware Graph Coupling Structure for GIS Water Flow Representation

Sergio Rosim[1], Antônio Miguel V. Monteiro[1], Camilo Daleles Rennó[1], João Ricardo de F. Oliveira[1]

[1]Instituto Nacional de Pesquisas Espaciais (INPE) Caixa Postal 515 – 12.227-010 – São José dos Campos – SP – Brazil

{sergio,miguel,camilo,joao}@dpi.inpe.br

Abstract. *A new computational representation for dealing with water flow in GIS applications is proposed. Different computer structures have been studied proposed and implemented and have gained a place in most of the commercial systems. Each particular local water flow data structure needs its own set of operators in order to provide support for Distributed Hydrologic Modeling. On this paper a new data structure based on Graph theory and preserving geographical awareness called G-GCS – Geographical Aware-Graph Coupling Structure (G-GCS) is presented. It is as a basis for a unified computer local water flow representation independent of the data structures used for representing the terrain topography.*

1. Introduction

The local flow distribution in a water basin is the most important item to develop distributed hydrologic modeling oriented to hydrological resources. The underlying premise is that terrain main landscape contributor in settling these local flows [Kiss, 2004; Soille, 1994]. The basis for terrain representation in GIS is to partition the region extent. A *Cell* is the unit of this partition set and the *local flow* is the water flow for each cell considering the status of its neighbor cells according to a specific neighborhood rule chosen.

The most common data structures found in GIS libraries and systems for terrain representation dedicated to hydrologic modelling are the DEM (Digital Elevation Models- DEM) [Burrough, 1998] with regular grids, Irregular Triangular Networks – TIN [Chew, 1989], Contour Lines based representation [Dawes, 1988] and Irregular Polygons Tessellations [Tucker, 2001]. Each surface representation chosen comes with its own local water flow extraction functions and its own local flow data structure attached to it. The local flow representation is dependent on the data structure used to represent the terrain topography. For instance, DEM local flow extraction uses the 8-neighbor idea, creating a local flow representation, called *Local Drain Direction* (LDD) [Burrough, 1998].

Distributed hydrologic modelling environments normally assume a unique data structure for terrain representation. This may simplify software development but it can't make use of the properties of the other terrain data structures. The model proposed here helps to simplify software development and at the same time it allows the use of diverse terrain data structure. In this way, decoupling terrain data structure and local flow data structure eliminates the need to codify a given operator for each terrain data structure used.

The main idea is to decouple the local flow representation from the set of functions needed for its generation. The solution proposed consists in having the local flow mapped into a graph-based structure, the G-GCS, which is then coupled to the particular terrain data structure on which its generation has been based. Figure 1 presents the mapping local flow got from DEM, TIN, Contour Lines or from other possible surface data structure representation into the proposed G-GCS.

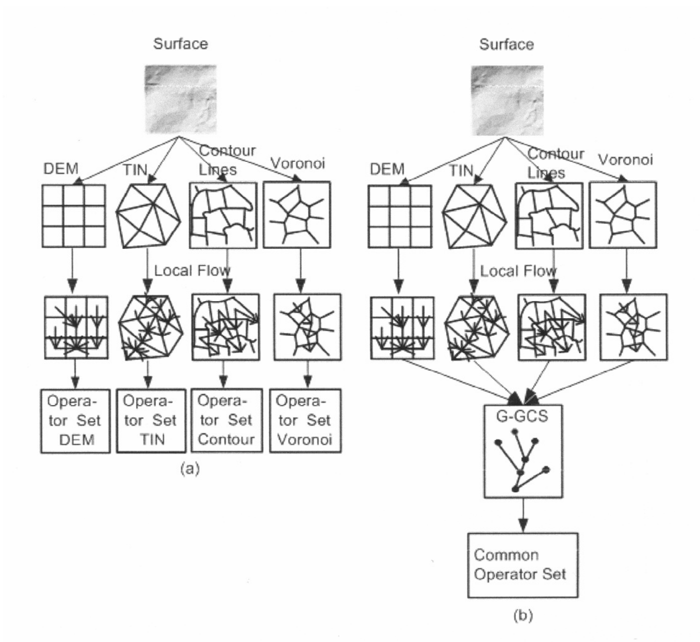

Figure 1. (a) **Present state in GIS Hydrologic Modelling;** (b) **New approach for building GIS Distributed Hydrologic Modelling Applications using the G-GCS idea.**

A feature of this new approach is that any new data structure for terrain representation can be incorporated into the framework with no impact at all on existing running models. The set of basic operators and functions can be extended and more complex operation can be built on top of the basic operator set. In order to demonstrate the G-GCS proposal at work we have defined a set of basic operations and have embedded them into a Distributed Hydrologic Modelling Toolkit prototyping environment based in Haskell and the TerraLib GIS free library [Costa, 2006]. Haskell was used because IT is a language that allows easy and fast implementations of prototypes. Same model was implemented using the PCRaster system that has its local flow representation bound to a grid structure. A comparative analysis based on the outcomes from these two implementations of the same model was conducted and the results are discussed over the sections (5) and (6) of this paper.

2. Previous Works

Methods to extract local water flow from the surface representation structures were developed to DEM, as the D8 unidirectional algorithm where the flow follows to the steepest descent 8-neighbour [O'Callaghan, 1984]. The Rho8 is a stochastic version algorithm of the D8 algorithm [Fairfield, 1991] FD8 and FRho8 are the changes of the previous algorithms, allowing dispersion flowing [Freeman, 1991; Quinn, 1991]. Some improvement has been achieved with methods that remove false pits and plane areas [Soille, 1994; Jenson, 1988].

Hydrological models using these methods of local flow extraction have been embedded in GIS systems as a computerized add-on for GIS hydrological modelling. The ArcGis Hydro Module [Maidment, 2002], The Grass GIS [GRASS, 1993], the Topographic Parameterization (TOPAZ) [Garbrecht, 1997], the Topography based Hydrological Model (TopModel), the MIKE SHE [DHI, 1998] and the PCRaster [Deursen, 1995] are examples of systems using DEM terrain representation based on a regular grid data structure for hydrologic modelling. The Watershed Modelling System (WMS) [Nelson, 1994] uses DEM and TIN data structures. TIN-based Real-Time Integrated Basin Simulator (tRIBS) uses TIN data structure and the TOPOG [Dawes, 1988] e SASHI [Rennó, 2003] that use Contour Lines based representations.

3. Local Flow Extractions from a Set of Different Terrain Data Structures Representation

The extraction of local flow is entirely dependent on the data structure chosen for terrain representation. The consequence of this is that each representation requires its own specific extraction algorithms, as well as different formats to store its associated local flow.

3.1. Extracting local flow from DEM

DEM is a rectangular regular elevation grid formed by cells. The use of some algorithm that approximates the local flow direction by the direction of steepest downhill slope for each cell, results in a new grid overlay called the set of Local Drain Directions – LDD. Each cell in this new grid contains an integer code representing the local flow direction at that cell. Figure 2(a) shows possible local flows in a given cell (center); Figure 2(b) shows a codification mask that supports unidirectional or multidirectional local flows; Figure 2(c) shows the resulting code of the cell representing the sum of the corresponding directional codes.

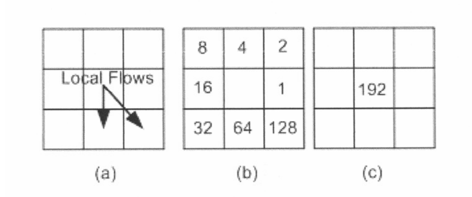

Figure 2. LDD grid creation, (a) **Local flows;** (b) **Codification mask;** (c) **LDD grid.**

3.2. Creating local flow from TIN

TIN structure normally created by Delaunay triangulation, stores information about these triangles and its sides and vertices, preserving the topological neighborhood of the triangle set. Triangles store information about its vertices and neighbor triangles. Sides store information about the two common triangles and about its two defining vertices. Vertices have three coordinates defining their position in space. TIN local flow has two kinds of propagation geometry: either the local flow crosses a triangle, flowing from a side to another, or it flows down along a side. Figure 3 presents an example of these two types of TIN local flow.

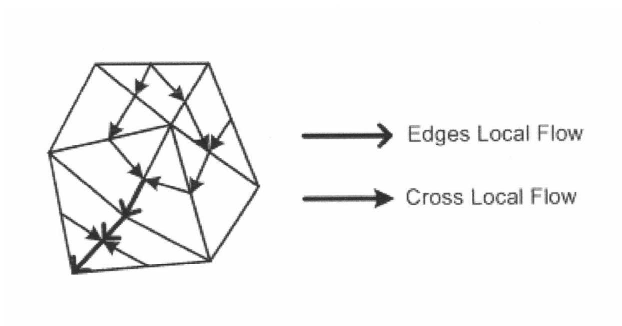

Figure 3. Cross flows and edges local flows

3.3. Creating local flow from Contour Lines

It is necessary to define an uniform flow unit between two neighbor contour lines forming a four-sided irregular polygon where two flow lines link two neighbor contour lines. In this case, local flow inside each flow unit is uniform, and its direction goes from the contour line with higher elevation value to the contour line with lower elevation value. Figure 4 presents local flow from contour lines.

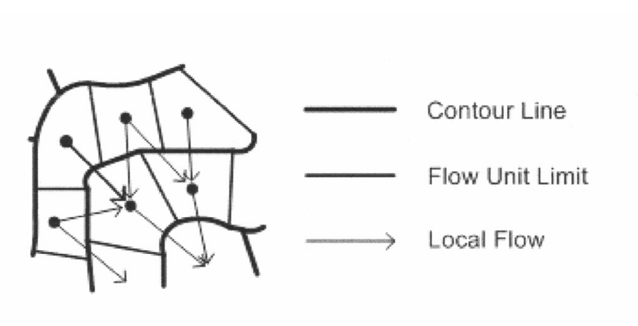

Figure 4. Contour Lines used to create local flows

3.4. Creating local flow from Voronoi diagram

The Voronoi diagram, a TIN dual structure, is a partition of space into cells, each one consists of points closer to a Delaunay triangle vertex than any other vertex. The local flow goes from a Voronoi cell to another through the triangles sides, beginning and ending at the center of neighbor cells, following the steepest downslope criterion [Tucker, 2001]. Figure 5 shows a Voronoi diagram with the local flow triangle sides.

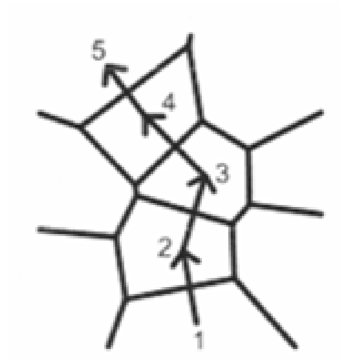

Figure 5. Voronoi used to generate local flows

4. Graph-Based Coupling Structure (GCS)

Labeled acyclic directed graph is a natural way to represent connected local flows. A graph G(V,E) is a vertex set V and an edge set E, where $e \in E, e = (v_i, v_j) \mid v_i \in V, v_j \in V \text{ and } i \neq j$ [Ore, 1962]. A directed graph is a graph with oriented edges. An acyclic graph is a graph without loops. A labeled graph is a graph where nodes and edges can store values. The main advantage of a graph is that it can store data in a structured way. This leads to very useful graph properties comparing to the others structures used to store local flow. Routines accessing graph data usually are more efficient than routines accessing local flow data structures.

Graph is a structure containing ready-to-use information about node links. Applications using graphs use this property to travel through the graph; local flow data structures do not have this property. So routines accessing graph data are usually more efficient than routines accessing local flow data structures.

4.1. LDD to Graph Map

Each LDD grid cell represents a graph node and the flow from a given cell to a neighbor cell defines a graph edge that links these two cells. Figure 6 shows the local flow from cell ‘4’ to cell ‘8’ and its graph representation.

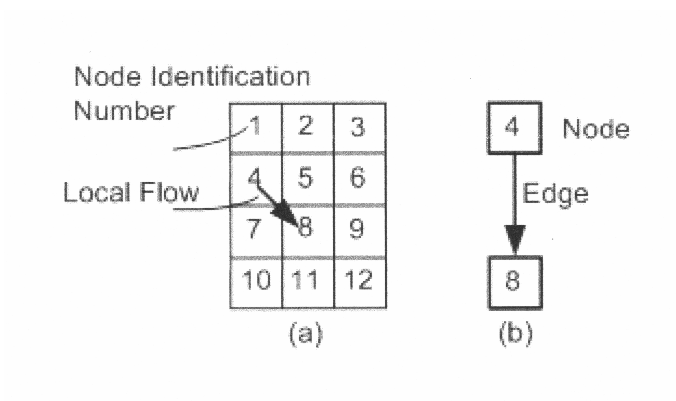

Figure 6. (a) Local flow; (b) Graph representation

4.2. LDD to Graph Map

As TIN local flow has two types of propagation geometry, each type needs a different approach to map TIN local flow to graph.

4.2.1. Triangle crosses local flow to graph

Each triangle side starting or ending as a local flow represents a graph node. Local flow goes from one side to another side of a triangle, passing through their middle points. The graph nodes identifiers are the same associates to the triangles sides during the triangulation process. This enables the remap from graph to triangulation, if desired. Figure 7 shows this local flow type.

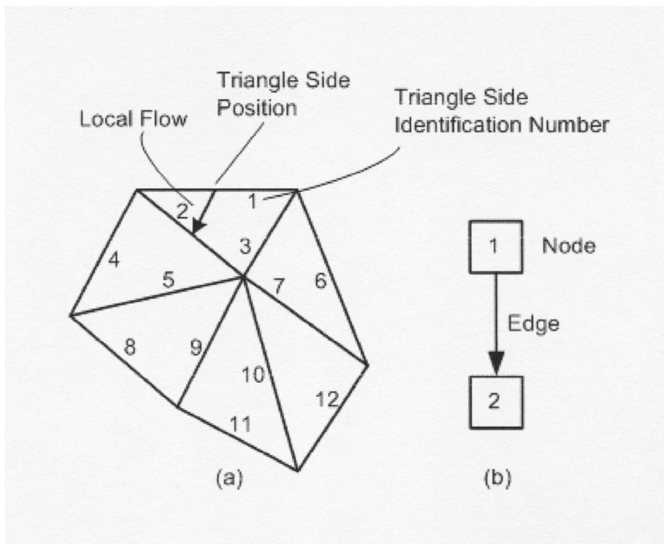

Figure 7. (a) Triangle crosses local flow; (b) Graph representation

4.2.2. Common triangles edge local flow map to graph

When a local flow goes along a triangle side, the vertices of that side represent graph nodes. The graph node identifier corresponding to triangle vertex is computed adding the total number of triangles sides to the vertex identifier from triangulation. Figure 8 shows this graph edge type and the corresponding detailed graph structure.

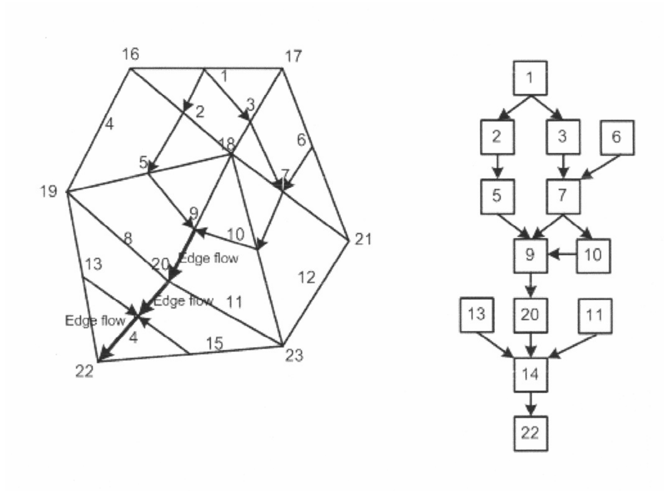

Figure 8. Mapping triangle local flows with edge flows to graph edges

4.2.3. Contour line to graph map

The local flow goes from each cell to one or more neighbors, passing through their centers. The graph node stores the cell identification number and a graph edge is a link between two cells. Multi flow issue is intrinsic in the contour lines data model. Figure 9 presents an example of the mapping from contour lines to graph.

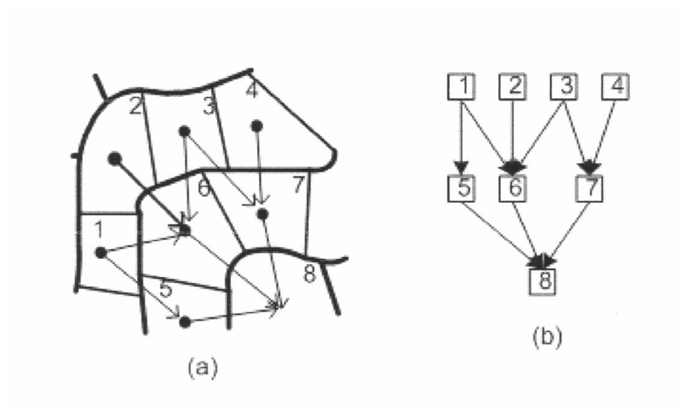

Figure 9. (a) Uniform flow unit local flows; (b) Graph representation

4.2.4. Voronoi to graph map

Each Voronoi polygon is a graph node and each graph edge represents a link between two neighbor polygons. Graph nodes store the Voronoi identification numbers existing in the Voronoi data structure and this approach is similar to grid cell approach. Figure 10 shows Voronoi to graph map.

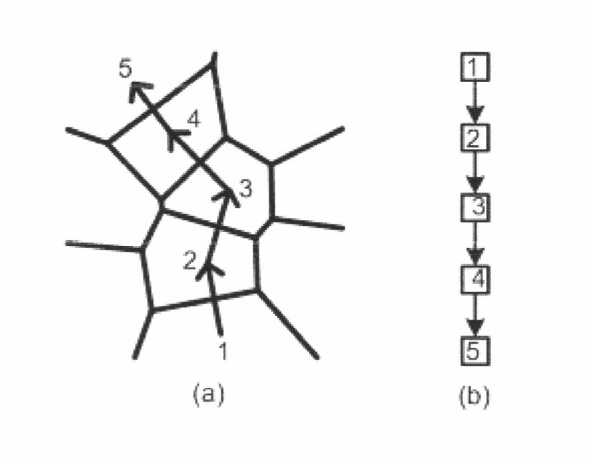

Figure 10. (a) Voronoi local flows; (b) Graph representation

5. A Water Flow implementation Problem with G-GCS

The computational implementation of G-GCS is not dependent of any programming language. Here, the implementation has used the Haskell functional language [Peyton Jones, 2002; Hudak, 2007] and the Functional Graph Library (FGL) [Erwig, 2001], which has several functions to create and manipulate graphs. Haskell language allows quick codification, depuration and prototype testes. Other decisive factor in choosing Haskell was the ease of use of many graph functions provided by FGL.

The example using PCRaster system "Simplified Hydrological Runoff Model" was chosen [Karssenberg, 1997] to compare with the same implementation with the G-GCS proposed. This application uses a LDD grid, a soil infiltration grid and a rainfall temporal series with same grids obtained from three rainfall stations to calculate flow accumulation for each grid cell.

This application aim is to determine the flow accumulation for each grid cell that has an infiltration capacity value. If the sum of the water arriving in a cell, coming from its neighbor cells, plus the rainfall at the same cell, exceeds its infiltration capacity, the water excess will flow out of this cell. Otherwise, the whole water quantity will be retained in the considered cell. Flow accumulation can be calculated using the water balance equation [Rennó, 2003].

P is the rainfall, E_{int} is the part of precipitation intercepted by canopy and evaporated afterwards, E_s is the soil evaporation, E_p is the water evaporation from canopy, Q_{out} is the runoff, Q_{in} is the water flow in the system and I is the quantity of infiltration water. The example of the PCRaster system doesn't consider the evaporation, then $E_{\text{int}} = 0,\ E_s = 0,\ E_p = 0$ and it assumes that the system infiltration capacity as infinite. It means that each time step is independent of the previous time steps. Figure 11 shows all components considered by this modelling.

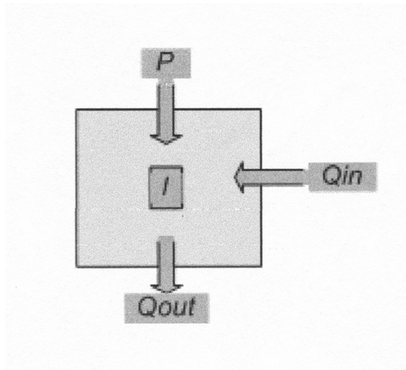

Figure 11. Simplified water flow balance diagram

PCRaster uses the accuthresholdflux and accuthresholdstate functions to accomplish this example. The accuthresholdflux function assigns the water that is transported out of the cell and the accuthresholdstate function calculates the amount of water infiltrated in the cell. Figure 12 presents the grids used in this example that were extracted using the NutShell, a PCRaster graphical user interface.

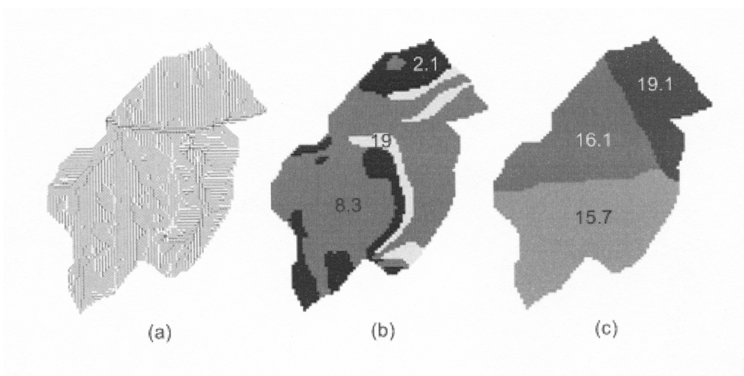

Figure 12. Grids used in this application. Numbers at the Figure are infiltration and rainfall values; (a) **LDD grid;** (b) **Soil infiltration grid;** (c) **Rainfall grid.**

The implementation here devised uses data stored in TerraLib [Câmara, 2000] open source geographical library implemented in C++ language. Terraview, a TerraLib based software, is used to visualize and to manipulate vector and raster data preserved in geodatabases to read, write and visualize the grids. The grids used in this application are in TerraLib format and they can't be directly read in Haskell. A binding in C language called Terra-HS [Costa, 2006] was used to access the TerraLib grids.

Any developing application needs to execute the same steps independent of the particular application. Initially, it is necessary to have in separated files the grids that will be used. The computer execution sequence is: (1) read the LDD, infiltration and rainfall grids of TerraLib; (2) create a graph using LDD grid, assigning the infiltration and rainfall values at the respective graph nodes; (3) execute the accumulate flow function; (4) convert accumulation nodes values to a grid; (5) write this grid in Terralib format. Figure 13 presents the general steps and the particular implementation used.

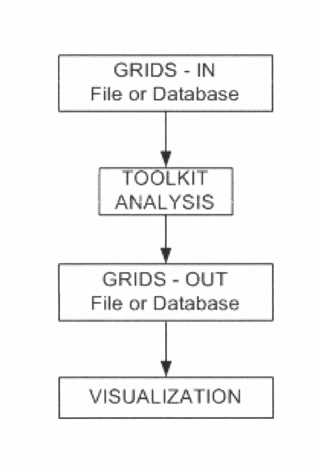

Figure 13. Developing steps to implement an application with G-GCS.

The Haskell accumulation flow function result is converted from graph to grid (4) and is saved in the TerraLib format (5) where it can be seen using TerraView software. Figure 14 shows G-GCS result.

Figure 14. Result obtained with G-GCS approach

The results obtained using the model proposed here is exactly the same obtained using PCRaster, validating the proposed model. This was expected because in both cases the local flow was expected from the same LDD grid. The next step in this work will be the creation of the graph and the flow accumulation calculus for local flows extracted from TIN. The Very Important Points (VIP) [Chen, 1987] algorithm will be used to reduce DEM data points to generate TIN sample points.

To illustrate the graph usage in the drainage network implementation, the drainage definition operator (DefDrain) will be presented. This operator defines a subset of the accumulated area network using a threshold parameter. All network nodes with values greater than the threshold will belong to the network drainage. This operator is easily done by using the graph structure.

6. Conclusions

It is possible to use a single coupling structure to represent local flows generated from different surface representation structures. This was illustrated for the accumulation flow function using DEM structure and the PCRaster system, comparing their results, showing they are similar.

The surface representation structure is decoupled from the operator set linked to the G-GCS structure. This guarantees that the operator set does not need to be duplicated and, at the same time it can be utilized the best characteristics of each surface representation structure.

References

Burrough PA, McDonnell RA (1998) Principles of Geographical Information Systems. Oxford University Press, New York

Câmara G, Souza RCM, Pedrosa B, Vinhas L, Monteiro AMV, Paiva JA, Carvalho MT, Gattass M (2000) TerraLib: technology in support of GIS innovation. II Brazilian Symposium on GeoInformatics, GeoInfo2000, São Paulo.

Chen ZT, Guevara JA (1987) Systematic selection of very important points (VIP) from digital terrain model for constructing triangular irregular networks. In N. Chrisman, editor, Proc. of Auto-Carto 8 (Eighth Intl. Symp. on Computer-Assisted Cartography), pages 50–56, Baltimore, MD. American Congress of Surveying and Mapping.

Chew LP (1989) Constrained Delaunay triangulations. Algorithmica 4(1):98-108

Costa SS, Câmara G, Palomo P (2006) TerraHS: integration of functional programming and spatial databases for GIS application development. VIII Brazilian Symposium on GeoInformatics, GeoInfo2006, Campos do Jordão, São Paulo, pp 109-127

DHI-Danish Hydraulic Institute (1998) MIKE SHE water movement - user guide and technical reference manual. Edition 1.1.

Dawes WR, Short DL (1988) TOPOG series topographic analysis and catchment drainage modelling package: user manual. Canberra, CSICO.

Deursen WPA (1995) Geographical Information Systems and dynamic models development and application of a prototype spatial modelling language. PhD Thesis, Faculty of Spatial Sciences, University of Utrecht

Erwig M (2001) Inductive graphs and functional graph algorithms. Journal of Functional Programming, 11(5):467-492

Fairfield J, Leymarie P (1991) Drainage networks from grid digital elevation models. Water Resource Research, 30(6):1681–1692

Freeman GT (1991) Calculating catchment area with divergent flow based on a regular grid. Computers and Geosciences, 17:413-422

Garbrecht J, Martz LW (1997) Automated channel ordering and node indexing for raster channel networks. Computers and Geosciences, pp 96–146

Garbrecht J, Ogden FL, DeBarry PA, Maidment DR (2001) GIS and distributed watershed models. I:Data Coverages and Sources, Journal of Hydrologic Engineering, 6(6):506-514

GRASS (1993) 4.1 User's reference manual. U.S. Army Corps of Engineers Construction Engineering Research Laboratory, Champaign

Hudak P, Hughes J, Jones, SP, Wadler P. (2007) A history of Haskell: being lazy with class. Third ACM SIGPLAN History of Programming Languages Conference (HOPL-III) San Diego, California, June 9-10, pp 55

Jenson SK, Dominque JO (1988) Extracting topographic structure from digital elevation data for geographic information systems. Photogrammetric Engineering and Remote Sensing 54(11):1593–1600

Karssenberg D, Wesseling CG, Burrough PA, van Deursen WPA (1997) A simplified hydrological runoff model. Department of Physical Geography, Ultrecht University, http://pcraster.geo.uu.nl/demos.html

Kiss R (2004) Determination of drainage network in digital elevation models, utilities and limitations. Journal of Hungarian Geomathematics, vol 2

Maidment DR (2002) Arc Hydro: GIS for water resources. ESRI Press, Redlands CA, 2002, pp 220

Nelson EJ, Jones NL, Miller AW (1994) Algorithm for precise drainage-basin delineation. Journal of Hydraulic Engineering, 120(3):298-312

O'Callaghan JF, Mark DM (1984) The extraction of drainage networks from digital elevation data. Computer Vision, Graphics, and Image Processing, 28:323–344

Ore O (1962) Theory of graph. American Mathematical Society, Michigan

Peyton Jones S (2002) Haskell 98 language and libraries the revised report

Quinn P, Beven K, Chevalier P, Planchon O (1991) The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. Hydrological Processes, 5:59-79

Rennó C D (2003) Construção de um sistema de análise e simulação hidrológica: aplicação a bacias hidrográficas (in portughese). Ph.D. thesis, National Institute for Space Research, São José dos Campos

Soille P, Gratin C (1994) As efficient algorithm for drainage network extraction on DEMs. Journal of Visual Communication and Image Representation, 5(2):181-189

Tucker GE, Lancaster ST, Gasparini NM, Brás RF, Rybarczyk SM (2001) An object-oriented framework for distributed hydrologic and geomorphic modelling using triangulated irregular networks, Computers & Geosciences 27:959–973