18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Generalized terrain-based flow analysis of digital elevation models

Tarboton, D.G. [1], **K.A.T. Schreuders** [1], **D.W. Watson** [1] **and M.E. Baker** [2]

[1] *Utah Water Research Laboratory, Department of Civil and Environmental Engineering, Utah State University, Logan, Utah, USA,*
Email: david.tarboton@usu.edu
[2] *University of Maryland, Baltimore County, Department of Geography & Environmental Systems, , Baltimore, Maryland, USA*

Abstract: Digital Elevation Models (DEMs) of topography are widely used in Geographic Information Systems (GIS) to derive information for the modeling of hydrologic processes. We present a general method for flow analysis that builds upon the representation of downslope flow using the proportioning of flow among one or multiple downslope grid cells, such as with the D8 or D-infinity models for the representation of downslope flow used in hydrologic terrain analysis, to enable the calculation of a new set of flow-based derivative surfaces. This general method which we refer to as *flow algebra*, extends flow accumulation approaches commonly available in GIS through the integration of multiple inputs and a broad class of algebraic rules into the calculation of flow related quantities. It is based on first establishing a flow field through DEM grid cells, or more generally model elements of any spatial discretization of the domain, by removing spurious pits and then proportioning the flow from a grid cell among downslope neighbors based on elevation. The flow field is required to be non-circulating, and as such these methods are applicable for flow problems related to the gradient of any potential field. Once a flow field is defined we show how it may be used to evaluate any mathematical function that incorporates dependence on values of the quantity being evaluated at upslope (or downslope) grid cells as well as other input quantities. The definitions of flow algebra functions are thus recursive and allow the evaluation of concepts that extend the concepts captured by recursive calculations of weighted contributing area available in GIS, to a broad class of flow related hydrologic terrain analysis functionality. Flow algebra encompasses single and multi-directional flow fields, various topographic representations, weighted accumulation algorithms, and enables untapped potential for a host of application-specific functions. While flow algebra expressions are recursively defined and most easily implemented using recursive functions, we describe here both recursive and non-recursive approaches to their evaluation. We have used the non-recursive approach to develop implementations for parallel processing architectures thereby enabling the rapid processing of large datasets.

We illustrate the potential of flow algebra by presenting examples of new functions enabled by this perspective that are useful for hydrologic and environmental modeling. A new weighted flow distance to stream function averages the "weight" moving through multiple flow paths from each point in the domain to the stream or other downslope reference point. This is an example of a general function which can specifically be used to estimate potential for sediment and nutrient filtering by streamside vegetation, by specifying vegetation cover as the weight. This function can also be used to provide a definition for connectivity in the context of multiple flow paths and identify areas that are connected to streams. Other new functions that examine elevations on upslope and downslope flow paths have been developed determine the average rise to ridge and drop to stream which together determine hillslope position useful for soil depth modeling. A new avalanche runout function determines the zone with downslope gradient greater than a critical angle (alpha) and is useful for mapping avalanche or potential landslide hazards.

Software implementing the functions illustrated here is available as part of the TauDEM package: http://www.engineering.usu.edu/dtarb/taudem.

Keywords: *Terrain analysis, Hydrology, Digital elevation models*

1. INTRODUCTION

Terrain analysis based on digital elevation models (DEM) is being increasingly used in hydrology (e.g. Wilson and Gallant, 2000). There is rapid ongoing improvement of DEM data available due to the evolution of DEM acquisition techniques such as LIDAR (light detection and ranging) that are producing high-resolution DEM data. We stand at a threshold of improvement in surface topography precision that provides both opportunities and computing challenges. High resolution data and increased computer power is leading to increased incorporation of terrain derivatives into analysis, in many fields, including hydrology. This paper contributes to methods for development of flow-related terrain derivatives that might enhance such analyses.

This paper examines models for the representation of flow over terrain and presents new formalism for deriving flow-based information useful for hydrologic modeling. A basic underlying assumption is that water and its constituents move downhill. Terrain-based flow models enrich the information available from a DEM by deriving a structured digital representation of the flow field, which serves as the foundation for calculation of a wide range of flow-related quantities. The established procedure (Wilson and Gallant, 2000) of (1) removing pits, (2) computing flow direction, and (3) computing the contributing area draining to each grid cell establishes the basic information model. Contributing area is one of the first and simplest flow related quantities computed from a DEM. The earliest method for delineating channel networks used a support area threshold applied to the grid of contributing areas (O'Callaghan and Mark, 1984), and this is still widely used today. In this paper the algorithm for calculating contributing area is generalized to include additional information and rules, and to produce additional spatial fields of interest. This generalized approach for the incorporation of rules into flow-related calculations is what we call flow algebra.

This paper is organized as follows. Section 2 describes the terrain based flow data model that gives a digital representation of the terrain flow field as the foundation for flow algebra which supports the calculation of derivative flow related surfaces, presented in section 3. In section 4 we describe how this methodology, which although recursively defined, may be implemented in a non-recursive fashion for parallel evaluation. Section 5 gives examples that illustrate the capability of flow algebra and section 6 gives discussion and conclusions. This paper builds on our earlier work (Tarboton and Baker, 2008) where the concept of flow algebra was introduced. In the present paper we indicate how flow algebra constructs can be implemented using parallel computing and can be used to derive examples of additional hydrologic quantities useful in the modeling of soil depth.

2. TERRAIN BASED FLOW DATA MODEL

The terrain-based flow data model (Figure 1) comprises a digital representation of terrain and a representation of the flow field that connects adjacent model elements enabling the routing of flow over a terrain surface and providing the basis for terrain-based flow calculations.

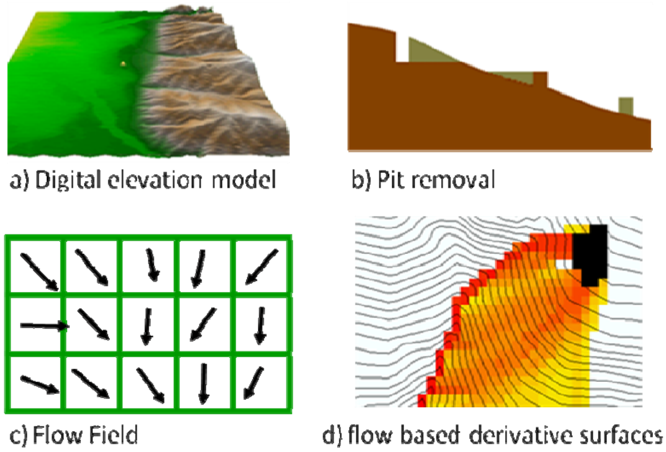

Figure 1. Terrain based flow data model comprised of a) Digital elevation model, b) procedures for removal of pits (e.g. filling or carving), c) definition of a flow field specifying the connectivity between model elements; and d) flow based derivative surfaces.

In grid DEMs, pits comprised of grid cells surrounded by higher-elevation neighbors occur due to deficiencies in DEM production processes (Jenson and Domingue, 1988; Jenson, 1991). Pits are generally interpreted to be spurious and drainage correction that removes pits is the first step in most analyses. Drainage correction alters the DEM to remove pits. Methods for drainage correction include pit filling (Jenson and Domingue, 1988; Jenson, 1991), breaching or carving (Garbrecht and Martz, 1997; Soille et al., 2003) and optimal pit removal (Soille, 2004). In an accompanying paper we present a parallel implementation of a pit filling algorithm (Wallis et al., 2009a).

The most common procedure for routing flow over a terrain surface represented by a grid DEM is the eight-direction method (D8) where the direction of steepest descent towards one of the eight (side and diagonal) neighboring grid cells is used to represent the flow field (O'Callaghan and Mark, 1984). Multiple flow direction methods that proportion the outflow from each element between one or more downslope elements have been suggested as an attempt to resolve the limitations of D8 (Quinn et al., 1991; Tarboton, 1997; Seibert and McGlynn, 2007). The D-infinity (D∞) multiple flow direction model (Tarboton, 1997) represents flow direction as a vector along the direction of steepest downward slope on eight triangular facets centered at each grid cell. Flow from a grid cell is shared between the two downslope grid cells closest to the vector flow angle based on angle proportioning.

All flow field methods assign or proportion flow from each grid cell to one or more of its adjacent neighbors. In grid DEMs the basic model element is a grid cell, but the same concepts can be applied to any set of topologically connected model elements. The flow proportions assigned to each downslope element are positive and should satisfy the conservation constraint:

$$\sum_j P_{ij} = 1 \tag{1}$$

where P_{ij} is the proportion of flow going from element i to a neighboring element j and the sum is over all the neighboring elements. For the D8 grid model these proportions are either 1 (connected) or 0 (not connected). For the multiple flow direction models these proportions fall between 0 and 1 for each neighboring element. There is also a requirement that flow is non-circulating such that no portion of flow leaving one element ever returns to the same element after passing through one or more of its neighbors.

Many measures useful in hydrologic and environmental modeling have been derived from this flow model (e.g. wetness index, Beven and Kirkby, 1979; erosion, Roering et al., 1999; contaminant transport, Endreny and Wood, 2003; and riparian buffers, Baker et al., 2006). Typically these measures have involved combining existing fields such as slope with outputs of an accumulation operation such as contributing area.

3. FLOW ALGEBRA

Once a flow field comprising a set of flow proportions for each model element is defined, it may be used to evaluate contributing area and other accumulation derivatives across a DEM domain. In the most general sense, the flow field derived from a DEM defines the surface connectivity between any two parts of a landscape. Given a flow field, the general accumulation function is defined as an integral of a weight or loading field, w, over a contributing area, CA.

$$A = \int_{CA} w \, da \tag{2}$$

For a direct contributing area calculation, the weighting field, w, is set equal to 1. In an example calculation of streamflow from excess rainfall, the weighting field would be set equal to rainfall minus infiltration.

Mark (1988) presented a recursive algorithm for evaluation of accumulation in the D8 case that was extended to multiple flow direction methods by Tarboton (1997) (Table 1). Numerically flow accumulation is evaluated recursively for each element as:

$$A_i = w_i \Delta + \sum_{\{k:P_{ki}>0\}} P_{ki} A_k \tag{3}$$

where i is a location in the field represented numerically by a model element such as grid cell in a DEM and A_i represents the accumulation at that element. The model element area is Δ and the notation $\{k:P_{ki}>0\}$ denotes that summation is over the set of k values such that $P_{ki}>0$ (i.e., summing the contribution from neighboring elements k to element i). In other words, accumulated flow at any model element is the sum of flow arising from that element and flow arising from all contributing neighboring elements, each weighted according to the proportion of flow it contributes. This is a recursive definition because the accumulated flow for any model element depends upon the accumulated flow of adjacent upslope elements. Recursive definition includes a requirement that in tracing each path upstream, one must eventually arrive at a source element that has no other elements draining into it. This "termination requirement" is satisfied as long as the flow field is non-circular. Contributing area, as we have defined it in (2) and (3) above, is ill-posed for any flow field that includes looping.

Table 1. Pseudo-code for recursive accumulation algorithm.

```
Global P, w, A, Δ
FlowAccumulation(i)
for all k neighbors of i
        if Pki>0
                FlowAccumulation(k)
next k
Ai = wiΔ + Σ{k:Pki>0} Pki Ak
return
```

There is a generality to the recursive functionality described above: 1) multiple direction accumulations rely on weighted flow proportioning, whereas single direction accumulations are a special case where all flow follows one pathway; (2) flow proportioning can occur to any number of neighboring model elements, so long as it conforms to the conservation constraint; (3) recursion can occur in both upslope and downslope directions; and (4) accumulations can be weighted by additional field(s) (e.g., rainfall minus infiltration). This capability, at least for upslope recursions, is available in flow accumulation functions in general purpose GIS software. However, we suggest here that recursive flow analysis need not be limited to the incorporation of additional weight fields into flow accumulation. Rather, what is needed is the ability to involve one or more additional fields in the accumulation functions that operate during the recursion according to a set of logical rules. We call these general rules for flow related calculations *flow algebra*. By generalizing the functionality of flow-field related calculations, we hope enable examination and use of additional flow-derived measures in the analysis of hydrologic processes.

Flow algebra logic exploits the recursive evaluation methodology illustrated in equation (3). Recursion serves to simplify the evaluation of a flow algebra function from its global or zonal integral definition, such as in equation (2), to a local evaluation where the function value at an element depends only on variables at that element and at either elements immediately upstream or downstream in the flow network, but not both at the same time. Flow algebra also generalizes the capability of zonal integral functions, enabling the evaluation of quantities that could not be defined in terms of a zonal integral because the result depends on both the flow field as well as local rules or additional value fields. We distinguish within flow algebra between simple input variables and variables with recursive dependence. Simple input variables or fields, denoted $\underline{\gamma}$, are fully quantified before the evaluation of a flow algebra expression. Variables that have recursive dependence on the flow field, denoted $\underline{\theta}$, are quantified during the course of evaluating a flow algebra expression.

In general, a flow algebra expression may be written as:

$$\underline{\theta}_i = \mathrm{FA}(\underline{\gamma}_i, \underline{P}_{ki}, \underline{\theta}_k, \underline{\gamma}_k) \tag{4}$$

The function FA may include any mathematical operators. In this expression $\underline{\theta}_i$ is a vector that lists the recursive variables being evaluated at location i. $\underline{\gamma}_i$ is a vector of all simple input variables. $\underline{P}_{ki}$ is a vector listing the proportion of flow from the first subscript element to the second subscript element, defined over all k for which $\underline{P}_{ki}$ is non-zero. $\underline{P}_{ki}$ represents the connections to neighboring elements. $\underline{\theta}_k$ is a list of all recursive variables evaluated at each neighbor location k. $\underline{\gamma}_k$ is a list of simple input variables at each neighbor element k.

For example, consider the problem of calculating runoff generation, where the input precipitation r, either infiltrates or runs off depending on a spatial infiltration or retention capacity, c. This could be accommodated in the present standard GIS flow accumulation method using a weight field, w=min(r-c,0) as input. However, now consider the effect of run-on. Runoff from upstream grid cells may have the opportunity to infiltrate downslope if the downslope infiltration capacity has not been exceeded. This cannot be evaluated using w as a weight for a standard flow accumulation. Rather, a rule is required where the runoff from each grid cell, i, is $q_i=\min(q_{in}+r_i-c_i,0)$ where q_{in} is the sum of runoff inflows from upslope. Using flow algebra this is evaluated by setting the vector $\underline{\theta}_i=q_i$. In this case $\underline{\theta}$ is a scalar at each grid cell, i. The vector of simple input variables $\underline{\gamma}$ includes r and c at each grid cell. The flow algebra function FA (4) is then defined as

$$\mathrm{FA}(r_i, c_i, \underline{P}_{ki}, q_k) = \min\Big(\sum_{\{k:P_{ki}>0\}} P_{ki} q_k + r_i - c_i, 0\Big) \tag{5}$$

A general recursive flow algebra algorithm is given in Table 2. In this pseudocode, FlowAlgebra is the recursive wrapper function, while FA is the functional flow algebra expression (4), of which (5) is an example, that is evaluated at each grid cell to implement the particular function being evaluated. FA may be as simple, or as complicated, as is needed, as long as it can be expressed in terms of simple and recursive input variables at a cell and its neighbors and that dependence on the recursive variables is in one direction only, i.e. either upstream or downstream. These are the general rules for this algebra formalism. The recursive variables, $\underline{\theta}_k$, appear on the right-hand side of the expression because evaluation of the expression at location i depends on the values for these variables at adjacent model elements (grid cells), either upstream or downstream. With this structure, both γ and θ fields can be used during the calculation of any quantity with recursive dependence. A flow algebra expression is either of type "upstream" or "downstream" depending on whether the functional dependence is on upstream or downstream quantities. Recursive dependence upon both upstream and downstream variability in the same expression is not allowed because such recursions would

Table 2. Pseudocode for general recursive flow algebra algorithm.

```
Global P, γ, θ
FlowAlgebra(i)
for all k neighbors of i
        if Pki>0
                FlowAlgebra(k)
next k
θi = FA(γi, Pki, θk, γk)
return
```

not terminate. Downstream flow algebra is obtained by reversing P_{ki} to P_{ik}, but is otherwise similar to upstream flow algebra in all respects.

Flow algebra expands upon the concept of map algebra available in popular GIS systems by including flow field operations. Map algebra involves point-by-point (cell-by-cell) mathematical operations between spatial fields. Flow algebra adds to this capability by incorporating operations based on the flow field and algebraic or functional descriptions of how the quantity being modeled is related to, and involved with, the flow field. Nothing in the formulation of flow algebra limits it to grids; it is applicable to any numerical representation of a flow field, including Voronoi polygons based upon a TIN discretization, or flow net model elements based upon a contour and flow line discretization as long as the underlying logical network connectivity between elements is non circular and can be represented by a set of proportions that satisfy (1).

4. PARALLEL IMPLEMENTATION

In order to evaluate a flow algebra expression for large grids in parallel, a method must be devised to partition the data across multiple processes. We have developed such a method for contributing area that is readily generalized to flow algebra (Wallis et al., 2009b). A striped partitioning scheme where the grid is divided horizontally into p equal parts and mapped to p processes, with any portion of the grid remaining being attached to the last divided portion, is used. Each process reads in their assigned portion of the DEM from a file, along with a row of cells directly above and below the assigned portions. This allows each process to have quick access to all neighboring cells without the need for any extra communication between processes. Before the function FA can be evaluated for a cell i, all of the cells in the region that drain into i must first be calculated. So that different cells can be considered simultaneously in different processes, a queue-based algorithm is used. Each parallel process independently evaluates its own partition as much as possible, accompanied by sharing of data between processes when needed. To achieve this, a dependency grid is created. The dependency grid contains, at each cell i, the number of immediate neighbors that drain into i. If there are no neighboring cells that drain into i, that cell is considered a peak of the DEM (and thus is not dependent on any other cell), so it is placed on the queue, allowing its flow algebra expression to be evaluated. If there is a neighbor that drains into i, the number of neighbors that drain into i is recorded in the dependency grid. This number is used later to determine when cell i is ready to be calculated. Table 3 gives the pseudo-code for building the dependency grid. Each process completes this phase in parallel with all other processes; no communication is necessary.

Table3. Pseudocode for building the dependency grid.

```
Executed by every process with grid flow
field P, grid dependencies D initialized to 0
and an empty queue Q.
FindDependencies(P,Q,D)
for all i
        for all k neighbors of i
              if Pki>0
                    D(i)=D(i)+1
        next k
        if D(i)=0
              add i to Q
next
```

Once the initial step has completed, each process contains a queue containing cells that are ready for evaluation and a grid filled with number of cell dependencies. Each process begins popping cells off the queue and evaluating the flow algebra expression. Then the dependency grid is decremented by one at downslope neighbors, n. If the dependency grid becomes zero at n, all cells upslope have been evaluated and the cell n is put on the queue. It is possible however that n may not pertain to a cell in the partition of that process, but rather to a neighboring one. In this case a dependency buffer that is used to facilitate message passing across processes is decremented. Upon completion of process queues, communication between processes is performed to swap dependency information and the procedure is iterated until all cells have been evaluated. Table 4 gives pseudo-code for the parallel flow algebra function.

Table 4. Pseudocode for parallel flow algebra function.

```
Executed by every process with D and Q
initialized from FindDependencies.
FlowAlgebra(P,Q,D,θ,γ)
while Q isn't empty
     get i from Q
     θi = FA(γi, Pki, θk, γk)
     for each downslope neighbor n of i
          if Pin>0
               D(n)=D(n)-1
               if D(n)=0
                    add n to Q
     next n
end while
swap process buffers and repeat
```

5. EXAMPLES

Figure 2 illustrates the evaluation of infiltration capacity limited runoff generation with run on according to (5). Inputs are the r and c values indicated, with the proportional flow directions indicated by the arrows. Grid cells A and B on the left have no dependencies so may be immediately evaluated and using min(r-c,0) result in q=3 and 0 for grid cells A and B respectively. With these evaluated, the unevaluated dependencies of grid cell C become 0, so equation (5) can be evaluated for grid cell C. The inflow to C is q_{in}=0.6 x 3 = 1.8 from grid cell B. Then at C q=min(5-6+1.8,0)=0.8. Now all dependencies of D have been evaluated.

Inflows from cells A, B, and C, accounting for the proportions are 3 x 0.4 + 0.8 = 2, so the runoff from D is q=min(4-5+2,0)=2.

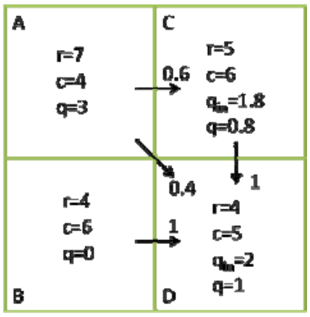

Figure 2. Example of infiltration capacity limited runoff generation accounting for run-on.

A natural measurement derived from any flow field is that of *distance along a flow pathway*. Specifically, we consider here the distance in a downslope direction from each model element to a target set, such as a stream or catchment outlet, though upslope distances may also be defined using an upslope recursion to obtain quantities such as the distance to a ridge. Average downslope distance may be determined using a flow algebra expression:

$$\mathrm{add}_i = \sum P_{ik}(\mathrm{dist}(i,k) + \mathrm{add}_k) / \sum_{\{k:\mathrm{add}_k \geq 0\}} P_{ik} \qquad (5)$$

In this expression 'add' stands for an average distance downslope that needs to be initialized to 0 at the destination set of points (such as the stream). dist(i,k) is a generalized distance function between grid cells i and k that may quantify distance horizontally, vertically or along the slope (Figure 3).

Similarly, average distance upslope may be determined using a flow algebra expression:

$$\mathrm{adu}_i = \sum P_{ki}(\mathrm{dist}(k,i) + \mathrm{adu}_k) / \sum P_{ki} \;\; \text{if} \;\; \sum P_{ki} > 0, 0 \text{ otherwise} \qquad (7)$$

In these functions dist may include a weighting factor to quantify the effectiveness of a buffer at, for example, preventing or filtering the sediment being carried to a stream. Figure 4 illustrates the buffer weighted potential distance to the stream. Note in this figure how the dark red colors are upslope of the area where the buffer around the stream is wide, while lighter reds occur where the buffer is narrow. These functions have been used in a study involving the modeling of soil depth based on topographic attributes (Tesfa et al., submitted 2008). This study found that average vertical rise to the ridge plus a dimensionless vertical hillslope position quantity defined from a combination of the longest vertical rise to the ridge and drop to the stream improved the explanatory capability of the regression model used to predict soil depth.

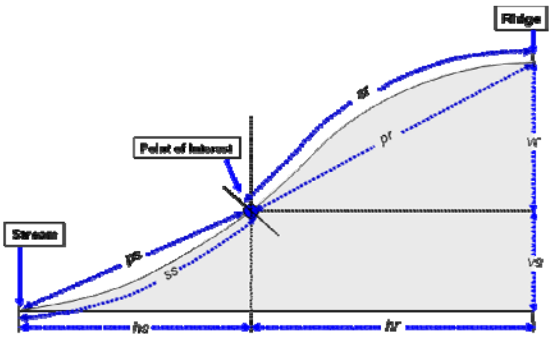

Figure 3. Types of distance measurements possible in add and adu functions.

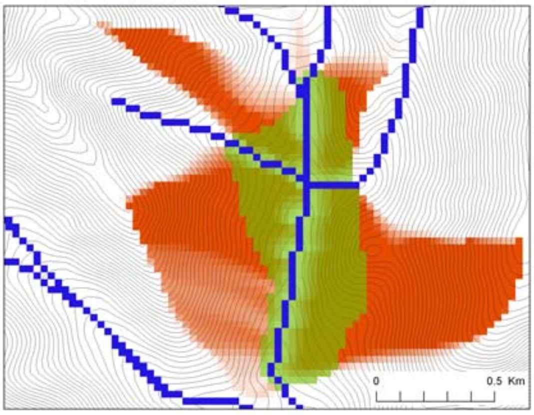

Figure 4. Buffer weighted potential distance to stream. The buffer is indicated in brown/green while the weighted distance is in red.

6. DISCUSSION AND CONCLUSIONS

Flow algebra is based on a flow field established through (1) drainage correction involving the removal of pits, followed by (2) definition of the flow field through a general multidirectional proportioning of flow from a grid cell to downslope neighbors. The flow field is required to be non-circulating, and as such is suitable for representation of flow derived from the gradient of any potential field. Flow proportions arising from any element (grid cell) should sum to one to ensure conservation. Once this flow field is defined, a broad class of upstream and downstream recursive functions may be constructed. We have presented some examples for exploiting this capability including new techniques for addressing the measurement of flow distances and elevation drops. The new techniques have already been utilized in several distinct applications and they serve to illustrate the potential of the flow algebra approach. Although the examples we present have been developed using grid data structures, the logic of flow algebra is applicable for any set of logically connected elements defining flow in a non-circulating flow field. Further work is needed to test flow algebra results, such as buffer weighted distance to the stream and runoff generation with run-on for their efficacy in representing physical processes. It would also be helpful to implement a symbolic system for definition of general flow algebra expressions to further facilitate evaluation of flow algebra functions in GIS.

Software implementing the functions illustrated here is available as part of the TauDEM package: http://www.engineering.usu.edu/dtarb/taudem.

ACKNOWLEDGMENTS

This research was supported by the US Army Research and Development Center under contract number W9124Z-08-P-0420 and CICEET, the Cooperative Institute for Coastal and Estuarine Environmental Technology.

REFERENCES

Baker, M. E., D. E. Weller and T. E. Jordan, (2006), "Improved methods for quantifying potential nutrient interception by riparian buffers," *Landscape Ecology*, 21(8): 1327-1345.

Beven, K. J. and M. J. Kirkby, (1979), "A Physically Based Variable Contributing Area Model of Basin Hydrology," *Hydrological Sciences Bulletin*, 24(1): 43-69.

Endreny, T. A. and E. F. Wood, (2003), "Watershed weighting of export coefficients to map critical phosphorous loading areas," *Journal Of The American Water Resources Association*, 39(1): 165-181.

Garbrecht, J. and L. W. Martz, (1997), "The Assignment of Drainage Direction Over Flat Surfaces in Raster Digital Elevation Models," *Journal of Hydrology*, 193: 204-213.

Jenson, S. K., (1991), "Applications of Hydrologic Information Automatically Extracted From Digital Elevation Models," *Hydrological Processes*, 5(1): 31-44.

Jenson, S. K. and J. O. Domingue, (1988), "Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis," *Photogrammetric Engineering and Remote Sensing*, 54(11): 1593-1600.

Mark, D. M., (1988), "Network models in geomorphology," in *Modelling in Geomorphological Systems*, Edited by M. G. Anderson, John Wiley., p.73-97.

O'Callaghan, J. F. and D. M. Mark, (1984), "The Extraction of Drainage Networks From Digital Elevation Data," *Computer Vision, Graphics and Image Processing*, 28: 328-344.

Quinn, P., K. Beven, P. Chevallier and O. Planchon, (1991), "The Prediction of Hillslope Flow Paths for Distributed Hydrological Modeling Using Digital Terrain Models," *Hydrological Processes*, 5: 59-80.

Roering, J. J., J. W. Kirchner and W. E. Dietrich, (1999), "Evidence for nonlinear, diffusive sediment transport on hillslopes and implications for landscape morphology," *Water Resources Research*, 35(3): 853-870.

Seibert, J. and B. L. McGlynn, (2007), "A new triangular multiple flow-direction algorithm for computing upslope areas from gridded digital elevation models," *Water Resources Research*, 43: W04501, doi:10.1029/2006WR005128.

Soille, P., (2004), "Optimal removal of spurious pits in grid digital elevation models," *Water Resources Research*, 40(12): W12509, doi: 10.1029/2004WR003060.

Soille, P., J. Vogt and R. Colombo, (2003), "Carving and adaptive drainage enforcement of grid digital elevation models," *Water Resources Research*, 39(12): 1366, doi:10.1029/2002WR001879.

Tarboton, D. G., (1997), "A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models," *Water Resources Research*, 33(2): 309-319.

Tarboton, D. G. and M. E. Baker, (2008), "Towards an Algebra for Terrain-Based Flow Analysis," in *Representing, Modeling and Visualizing the Natural Environment: Innovations in GIS 13*, Edited by N. J. Mount, G. L. Harvey, P. Aplin and G. Priestnall, CRC Press, Florida, p.496.

Tesfa, T. K., D. G. Tarboton, D. G. Chandler and J. P. McNamara, (submitted 2008), "Modeling Soil Depth based upon Topographic and Land Cover Attributes," *Water Resour. Res.*

Wallis, C., R. Wallace, D. G. Tarboton, D. W. Watson, K. A. T. Schreuders and T. K. Tesfa, (2009a), "Hydrologic Terrain Processing Using Parallel Computing," *Accepted for presentation at the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, Cairns, Australia, 13–17th July.

Wallis, C., D. Watson, D. G. Tarboton and R. Wallace, (2009b), "Parallel Flow-Direction and Contributing Area Calculation for Hydrology Analysis in Digital Elevation Models," *Accepted for presentation at PDPTA'09, The 2009 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada, USA, July 13-16.

Wilson, J. P. and J. C. Gallant, (2000), *Terrain Analysis: Principles and Applications*, John Wiley and Sons, New York, 479 p.