Digital Elevation Model Networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas

Article *in* Water Resources Research · June 1994

DOI: 10.1029/93WR03512

CITATIONS
369

READS
1,416

2 authors:

Mariza Costa-Cabral
Northwest Hydraulic Consultants
29 PUBLICATIONS **661** CITATIONS

SEE PROFILE

Stephen J. Burges
University of Washington Seattle
140 PUBLICATIONS **5,659** CITATIONS

SEE PROFILE

WATER RESOURCES RESEARCH, VOL. 30, NO. 6, PAGES 1681–1692, JUNE 1994

Digital elevation model networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas

Mariza C. Costa-Cabral and Stephen J. Burges
Department of Civil Engineering, University of Washington, Seattle

Abstract. Current algorithms for computing contributing areas from a rectangular grid digital elevation model (DEM) use the flow-routing model of O'Callaghan and Mark (1984), which has two major restrictions: (1) flow which originates over a two-dimensional pixel is treated as a point source (nondimensional) and is projected downslope by a line (one dimensional) (Moore and Grayson, 1991), and (2) the flow direction in each pixel is restricted to eight possibilities. We show that large errors in the computed contributing areas result for any terrain topography: divergent, convergent, or planar. We present a new model, called digital elevation model networks (DEMON), which avoids the above problems by representing flow in two dimensions and directed by aspect. DEMON allows computation of both contributing and dispersal areas. DEMON offers the main advantage of contour-based models (e.g., Moore et al., 1988), the representation of varying flow width over nonplanar topography, while having the convenience of using rectangular grid DEMs.

1. Introduction

The specific contributing area (SCA), and the specific dispersal area (SDA) (Figure 1), are distributed variables with important hydrological, geomorphological, and geological significance. The total contributing area (TCA) of a contour segment is the plan area of terrain that contributes flow to the contour segment. The SCA (designated often by "a" in the literature) of a contour segment is the TCA divided by the contour segment length. Because it is a plan area, the concept of SCA relies on the assumption that the plan view projection of flow directions does not change with depth below the land surface and is determined by surface topography. This assumption provides a valid approximation where the terrain permeability is small relative to the rainfall rate or where subsurface flow lines are approximately parallel in plan view to surface flow lines. The SCA is the plan area located topographically upstream from the unit contour length of interest. The SCA may be interpreted as an equivalent flow path length, because when upslope flow lines are parallel in plan view, the SCA equals the upslope flow path length.

In hydrology and geomorphology, the SCA is used extensively as an indicator of discharge. This assumes that the rate of flow generation **r** is uniform spatially and that the discharge rate is steady, in which case the specific discharge equals SCA times **r**. While these conditions of uniformity and equilibrium seldom are met in nature, they are assumed often to approximate natural conditions. Uses of the SCA as an indicator of discharge include studies of hillslope hydrologic response, channel location, long-term basin evolution, landslide risk, soil water content, and vulnerability to pollution, among others. Indexes that combine the SCA as a discharge indicator with other variables are used widely in hydrology and geomorphology. Examples of such indexes are ln(SCA/S), where S is slope, used to predict the soil moisture deficit [*Beven and Kirkby*, 1979], and S^2SCA, used to predict channel initiation by overland flow [*Montgomery and Dietrich*, 1989, 1992; *Montgomery and Foufoula-Georgiou*, 1993]. Use of the SCA as an indicator of discharge for prediction of the location and extent of channel networks is widespread. Reviews of channel network location based on SCA values obtained from rectangular digital elevation models (DEMs) include those by *Mark* [1988] and *Tarboton et al.* [1989, 1991].

The total dispersal area (TDA) of a contour segment is the plan area of terrain that drains flow from that contour segment. The SDA of a contour segment is the TDA divided by the contour segment length. The underlying assumption is the same as for the SCA definition, and the SDA is then the plan area topographically downstream from the unit length of contour. The SDA extends over the hillslope and may terminate at a location of topographic convergence or at a receiving water body. The SDA indicates the area of influence of flow generated at the given location and may be used, for example, to predict the influence zone of upslope pollution sources or of any development affecting terrain permeability. *Speight* [1974, 1980] interpreted the SDA as an indicator of soil drainage rate. Present uses of the SDA are more limited than those of the SCA.

The SCA and the SDA were defined originally for a segment of a contour line, and *Speight* [1974] pioneered the computation of these variables for landform classification, using a terrain partition based on contours and flow lines. Given the convenience of representing distributed variables for grid cells, or pixels, of rectangular digital elevation models (DEMs), the concept of the SCA was transferred from contour segments to DEM pixels and has joined the list of geomorphometric parameters that are computed routinely in the analysis of a DEM. Various models for flow routing in rectangular DEMs have been proposed for the computation

Paper number 93WR03512.
0043-1397/94/93WR-03512$05.00

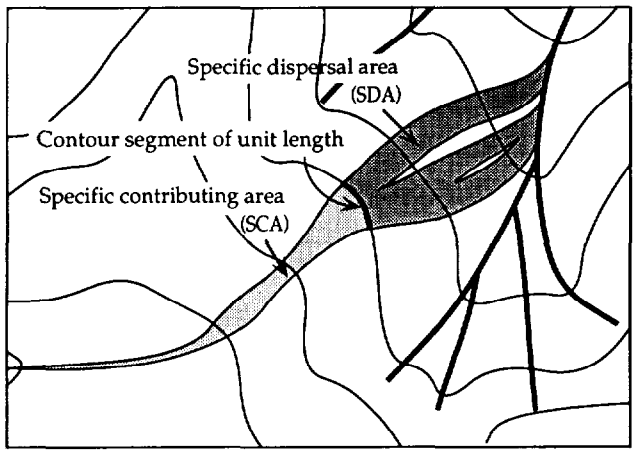


Figure 1. Illustration of the concepts of specific contributing area and specific dispersal area of a contour segment of unit length. The channel network drains to the top right of the figure.

of SCAs. These models have not been used, however, to compute SDAs, given their inappropriate (one-dimensional) downslope projection of flow. The same limitations that render current grid-based models unsuitable for computing SDAs make them equally inappropriate for computing SCAs. The major fault of most current models is their point source representation of flow generation and the resulting one-dimensional representation of flow paths [*Moore and Grayson*, 1991].

In digital elevation model networks (DEMON), flow is generated areally, not at point sources. Flow generated over a pixel is projected downslope over a two-dimensional flow strip, analogous to a flow tube. Flow direction is determined by the local aspect angle, in a manner similar to that used by *Lea* [1992]. The computed width of a "flow tube" increases over divergent topography, decreases over convergent topography, and remains constant over plane surfaces. DEMON offers the main advantage of contour-based models [e.g., *Moore et al.*, 1988], the representation of flow width variation as a function of local topography, and the benefits of rectangular grid DEMs. In addition, it permits computation of both SCAs and SDAs.

2. Review of Existing Methods and Their Limitations

Several methods exist for computing the SCA of a DEM pixel. We are unaware of methods for computing the SDA of a pixel. Computation of SDA values was possible previously only for the areal segments of a contour-based terrain partition [*Moore and Grayson*, 1991]. A review of the principal methods for computing SCAs for DEM pixels and what in our view are their limitations is presented below.

2.1. D8 [*O'Callaghan and Mark*, 1984]

The methods used most for SCA computation for DEM pixels are based on the flow-routing model introduced by *O'Callaghan and Mark* [1984]. We refer to them collectively as method D8. In method D8, pixels are centered on the DEM grid points, and each pixel discharges, or "spills," into one of its eight neighbors: the one located in the direction of steepest descent. The total contributing area (TCA) of a pixel is the number of pixels whose flow reaches the pixel of interest following a path of steepest descents, multiplied by one pixel area, $\Delta x \Delta y$. The SCA of a pixel is given by the TCA divided by the length of the segment orthogonal to flow direction in the pixel, either a pixel side, Δx or Δy, or the diagonal, $(\Delta x^2 + \Delta y^2)^{1/2}$.

Efficient algorithms for computing TCAs search upslope from a pixel and involve recursion. Examples include, among others, the algorithm of *Mark* [1988] and those of *Tarboton et al.* [1989, 1991].

The D8 method allows flow from each pixel to discharge to only one receiving neighbor. This amounts to treating flow which originated over a two-dimensional pixel as a point source (nondimensional) and projecting it downslope by a line (one dimensional) instead of a flow tube (two dimensional). This limitation has been pointed out by *Moore and Grayson* [1991]. The single receiving neighbor also imposes restrictions on possible flow path configurations because flow can occur only in either a cardinal or diagonal direction. The errors resulting from these limitations are different for areas where flow is divergent (where flow tube width increases downslope), convergent (flow tube width decreases downslope), or parallel (flow tube width is constant).

For parallel flow the true SCA is equal numerically to the flow path length. Method D8 computes the SCA correctly for parallel flow only when flow is in the x or y direction. When flow is at an angle to the principal grid orientation, two kinds of errors arise: errors that affect flow path direction and errors of SCA underestimation for a given flow path. The first source of error results when flow is at an angle different from a multiple of 45°. For example, if flow is at an angle of 30° (measured counterclockwise from east), then the steepest descent direction given by the D8 method will be toward the NE, and each pixel will discharge into its NE neighbor. This amounts to approximating the angle of flow to 45°, and modeled flow is diverted from its true path by 15°.

The second source of error results from the one-dimensional projection of flow. Consider parallel flow at an angle of 315° (a multiple of 45°), shown in Figure 2. The computed TCA is underestimated by a factor of 2 when using method D8. When using D8, the computed SCAs for parallel flow are correct for flow that is entirely in a cardinal direction (0°, 90°, 180°, or 270°), are underestimated by a factor of 2 for flow that is entirely in a diagonal direction (45°, 135°, 225°, or 315°), and are less than the correct value by a factor between 1 and 2 for flow that changes along its path between cardinal and diagonal directions. A simple correction by a factor of 2 is possibly only if the entire TCA of the

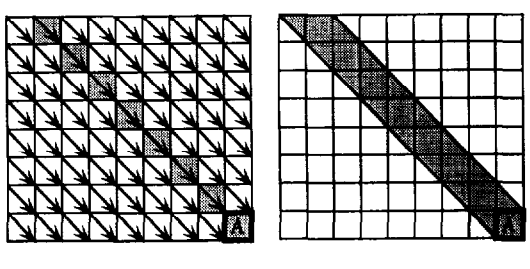


Figure 2. TCA of a pixel on a planar slope with aspect angle 315° (a) predicted by method D8 and (b) true TCA. Method D8 underestimates the TCA of pixel A by a factor of 2.

pixel is in a diagonal direction. In cases where flow paths have both cardinal and diagonal directions a correction factor between 1 and 2 must be determined.

For divergent flow over a right circular cone mountain (Figure 3), the true TCA of any pixel includes the pixel itself and a triangle with one vertex at the cone center and two vertices at pixel corners. The SCA of a pixel at a distance r from the cone center is numerically close to $\frac{1}{2}r$, and the SCA has radial symmetry (Figure 3d). Divergent flow cannot be represented by a linear path of steepest descents, hence the SCA values computed by method D8 do not have radial symmetry (Figure 3c). Modeled flow is concentrated arbitrarily on eight preferential paths oriented in the cardinal and diagonal directions, while avoiding other paths. Consequently, at any given distance r from the cone center, some pixels drain all pixels between themselves and the cone center and have an SCA close to r, an overestimation by a factor of 2; while other pixels drain only themselves and have an SCA equal to a pixel side (Δx or Δy), an underestimation that approaches 100% at large r. Similar problems arise for convergent flow, such as over a right circular cone crater (Figure 4). Again, modeled flow is concentrated along eight preferential paths (Figure 4c), although overestimation and underestimation of the SCA magnitudes are not as large as for a cone mountain.

2.2. Rho8 [*Fairfield and Leymarie*, 1991]

Method Rho8 attempts to solve one of the problems of method D8: the diversion of modeled flow paths toward a cardinal or diagonal direction, resulting arbitrarily from grid orientation. Method Rho8 introduces a stochastic component into method D8, yielding flow paths that reflect more closely the true aspect of hillslopes. As in method D8, each pixel discharges into one of its eight neighbors. (A variation, called method Rho4, considers only the four cardinal neighbors.) The choice of the receiving pixel among the neighbors is made stochastically. One of the neighbors is assigned a probability p of being chosen and another neighbor is assigned a probability $1 - p$.

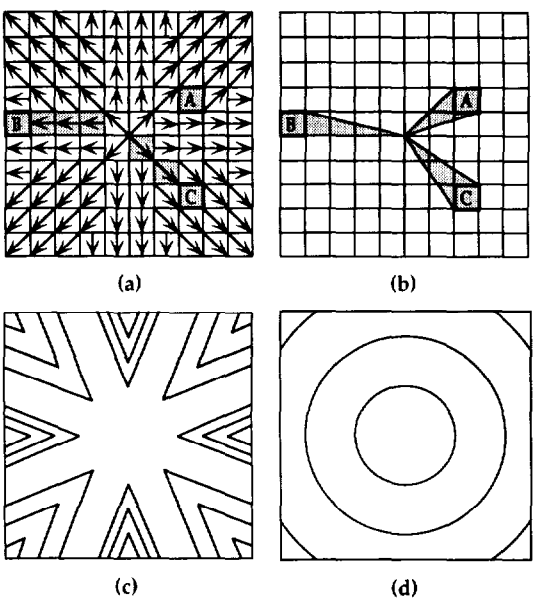


Figure 3. TCA of pixels on a right circular cone mountain surface (or TDA of pixels on a right circular cone crater surface): (a) TCA of pixels A, B, and C predicted by method D8, (b) true TCA of pixels A, B, and C, (c) SCA contours for method D8, and (d) true SCA contours. Radial symmetry is not conserved with method D8. With D8, pixel A drains only itself.

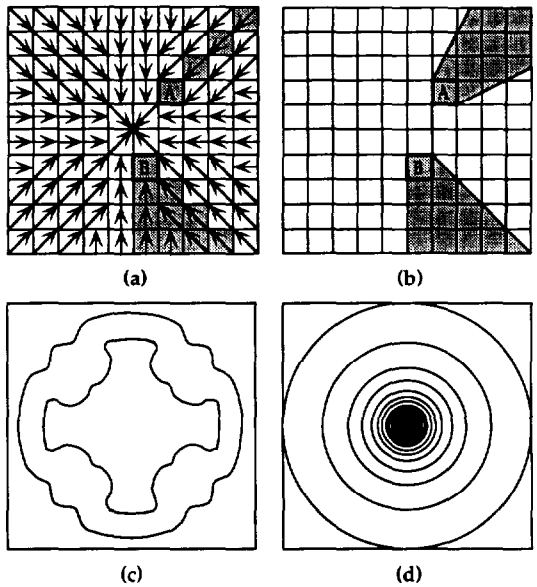


Figure 4. TCA of pixels on a right circular cone crater surface (or TDA of pixels on a right circular cone mountain surface): (a) TCA of pixels A and B predicted by method D8, (b) true TCA of pixels A and B, (c) SCA contours for method D8, and (d) true SCA contours. Radial symmetry is not conserved with method D8. The TCA of pixel A is underestimated greatly.

The scheme for assigning probabilities and the objective of this method are illustrated in the following example. Consider a plane surface with an aspect angle of 30° (measured counterclockwise from east). Method D8 makes every pixel discharge into its NE neighbor, resulting in a path direction that is wrong by 15°. Method Rho8 assigns a probability p to any given pixel discharging to its NE neighbor and a probability $(1 - p)$ to it discharging to its eastern neighbor. Therefore some pixels will discharge to their NE neighbor, and the remainder will discharge to their eastern neighbor. If the number of pixels discharging to the NE versus the eastern neighbor is in the right proportion (the expected proportion is $p/(1 - p)$), then the resulting flow lines will have an overall direction of 30°. The value of p should be such that the expected value of the flow path direction is equal to the aspect angle.

While this method provides, in mathematical expectation, appropriate flow path directions, all other problems identified above remain. Method Rho8 introduces problems of its own: randomness does not ensure reproducible results; and in locations of parallel flow, adjacent flow paths are not parallel but wiggle randomly and therefore often converge laterally with one another. Lateral convergence of flow paths on plane surfaces, where flow should be parallel, concentrates upslope flow on only some pixels. Once two flow paths have merged due to their random wiggling, there is no mechanism that can make them diverge again, hence errors increase downslope as flow becomes more and more con-

centrated. Some pixels will have overestimated TCA values, while others that were missed by the wiggling flow paths have underestimated TCAs.

2.3. *Lea* [1992]

This method solves the problem attempted by *Fairfield and Leymarie* [1991] by routing flow according to local aspect angle. The surface of each pixel is approximated by a best fit plane, and the direction of steepest slope (or the aspect angle) is computed (section 3.2 below). Flow is routed downslope along a path comprising straight segments with different directions, according to the aspect angle of each pixel. The algorithm models the entry and exit points of flow on the perimeter of each pixel along the flow path. Flow is modeled as a "rolling ball" (point source assumption) moving across the topographic surface in the direction of local aspect. Hence the major limitation of method D8, the one-dimensional representation of flow, remains.

2.4. Multiple-Direction Methods

Multiple-direction methods attempt to solve the major limitation of method D8, the one-dimensional representation of flow, by distributing flow from a pixel among all of its lower-elevation neighbor pixels, according to some specified rule. *Quinn et al.* [1991] proposed that the fraction of flow allocated to each lower neighbor i be determined by

$$f_i = \frac{S_i L_i}{\sum S_j L_j} \tag{1}$$

where the summation is for all lower neighbors, S is the directional slope, and L is an "effective contour length" that acts as a weighting factor. The values of L used by *Quinn et al.* [1991] were $\frac{1}{2}$ of the pixel side for cardinal neighbors and a fraction 0.354 of the pixel diagonal for diagonal neighbors.

Freeman [1991] proposed that the fraction of flow allocated to each lower neighbor i be determined by

$$f_i = \frac{S_i^p}{\sum S_j^p} \tag{2}$$

where S is the directional slope and p is a nonphysical parameter. This method was tested for flow over a right circular cone mountain, for different values of parameter p. Since $p = 1.1$ provided drainage contours which approached circles, this value was recommended for use. Freeman applied the method exclusively to divergent topography and used a different algorithm for convergent topography.

Figure 5 represents schematically the contributing area of a pixel, labeled "A," located on three different surfaces: a plane, a right circular cone mountain, and a right circular cone crater. The contributing areas shown are those that result from use of any multiple-direction algorithm, such as those of *Quinn et al.* [1991] or *Freeman* [1991]. For an inclined plane surface each pixel discharges to three or four other pixels (depending on plane aspect), and only a fraction of the discharge pixel surface area belongs to the contributing area of a receiving pixel. Therefore the contributing area of a pixel does not include any full pixel but instead is composed of portions of different pixels and is discontiguous. The flow direction over the plane in Figure 5a is top to bottom, and the true TCA of pixel A is the full area of the four pixels immediately above pixel A. Multiple-direction algorithms predict that the contributing area is composed of portions of pixels inside a triangular region. The computed TCA value of a pixel is correct if it is located far enough from the plane's lateral edges, otherwise the triangular region is incomplete and the TCA is underestimated. For the plane represented in Figure 5a, 70 of the 100 pixels represented have an underestimated TCA value (e.g., the TCA of pixels labeled "B" and "C"). Thus results are affected by boundary proximity. For a right circular cone mountain (Figure 5b) and a cone crater (Figure 5c) the TCA of pixel A is, again, discontiguous and includes portions of pixels located far outside the true (triangular) contributing area. Parameter calibration may yield predicted TCA values that are correct (e.g., $p = 1.1$ in (2) for a right circular cone mountain). The approximation relies on surface geometric symmetry and will suffer to the extent that a natural terrain surface will diverge from the symmetric geometric surfaces used for parameter calibration. Due to their misplacement of contributing areas, multiple-direction algorithms are not appropriate for contaminant tracing nor can they represent distributed runoff rates.

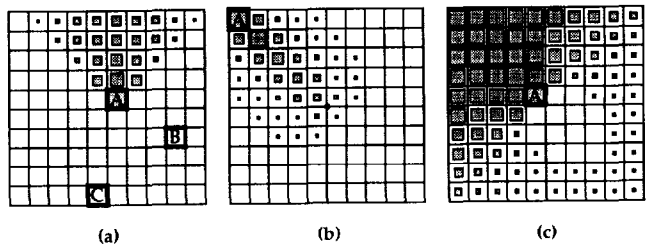


Figure 5. Schematic representation of the TCA of a pixel that is estimated by any multiple-direction algorithm. The size of the shaded square inside each pixel is proportional to the size of that pixel's portion that drains to pixel A. (a) Predicted TCA of pixel A located on a plane surface where flow is from top to bottom. (b) Predicted TCA of pixel A located on a right circular cone mountain. (c) Predicted TCA of pixel A located on a right circular cone crater.

2.5. Summary

The D8 family of methods, used to compute SCAs for DEM pixels, contains limitations which can result in large errors for any terrain configuration, including planar, divergent, and convergent topographies. More recent methods attempt to overcome some of the problems of method D8. *Lea* [1992] provides a sound scheme for aspect-driven routing which is an improvement over D8 routing. However, no satisfactory solution has been presented for the most important limitation of method D8, the point source assumption and one-dimensional routing. This problem has been addressed with partial success by multiple-direction models. In our view, the most important limitations of multiple-direction models are that the computed contributing areas are discontiguous, and the quality of the approximation of the computed values relies on geometric symmetry and is affected by boundary proximity.

We present an alternative approach which attempts to overcome the problems identified in this section by modeling downslope flow in two dimensions and in well-defined flow tubes.

3. DEMON

In this section we define the SCA and SDA of a pixel (or, in general, any terrain discretization element), discuss our representation of the flow field, and describe the DEMON algorithms used to compute the SCA and SDA matrices.

3.1. Definitions

The total contributing area (TCA) of a DEM pixel is the plan-view area draining to that pixel. Implicit in this definition is the assumption that the plan-view projection of the flow field does not change with depth below the land surface and is determined by surface topography. The TCA of a pixel is then the plan-view area of the collection of all points located upstream topographically from the pixel. A point is located upstream topographically from a pixel if the surface flow line passing through that point enters the pixel downstream. The SCA of a DEM pixel (or any terrain discretization element) is defined as the average value of contributing area per unit flow width as flow exits the pixel and is obtained by division of the pixel's TCA by the total exiting flow width, ω, that is,

$$\mathrm{SCA} = \frac{\mathrm{TCA}}{\omega} \tag{3}$$

The TDA of a DEM pixel is the plan-view area draining flow from the pixel. The implicit assumption is the same as in the TCA definition. The TDA of a pixel is the plan-view area of the collection of all points located topographically downstream from the pixel. A point is located topographically downstream from a pixel if there is a flow line that passes through the pixel and also through the point downstream. The SDA is the TDA per unit flow width ω:

$$\mathrm{SDA} = \frac{\mathrm{TDA}}{\omega} \tag{4}$$

The total flow width ω in (3) and (4) is the flow width orthogonal to the flow direction along the portion of the pixel boundary through which flow exits the pixel. For a DEM grid with x and y axes pointing east and north, respectively, and pixel dimensions Δx and Δy,

$$\begin{aligned}\omega &= \int_{l_N} \frac{\partial \omega}{\partial l}\, dl + \int_{l_E} \frac{\partial \omega}{\partial l}\, dl + \int_{l_S} \frac{\partial \omega}{\partial l}\, dl + \int_{l_W} \frac{\partial \omega}{\partial l}\, dl \\ &= \int_{l_N} \sin \alpha(x,\, y = \Delta y)\, dx + \int_{l_E} \cos \alpha(x = \Delta x,\, y)\, dy \\ &\quad - \int_{l_S} \sin \alpha(x,\, y = 0)\, dx - \int_{l_W} \cos \alpha(x = 0,\, y)\, dy \end{aligned} \tag{5}$$

where l_N, l_E, l_S, and l_W designate the lengths of the exit segments lying on the northern, eastern, southern, and western boundary segments, respectively, and $\alpha(x, y)$ is the flow direction angle, measured counterclockwise from east. Here l_N and l_S take values between zero and Δx, and l_E and l_W take values between zero and Δy.

3.2. Surface Fitting to the DEM and Determination of Flow Direction Angles

The SCA and SDA variables are determined entirely by the flow field, that is, the field of flow direction angles, $\alpha(x, y)$. Flow at a point is in the direction of maximal surface slope and is obtained from the reverse direction of the elevation tensor at that point. Thus determination of the flow angle field requires prior knowledge of the elevation field. Given that the DEM provides only a point sampling of elevations, it is necessary to fit an elevation surface to the DEM. This may be thought of as an attempt to reconstruct the topographic surface after it had been stored in the image-compressed form of a DEM. How well the reconstructed surface represents the true topographic surface will depend on the size of the DEM grid relative to the roughness of the terrain and on the surface-fitting method.

The simplest surface-fitting method approximates the surface of each pixel by a best fit plane using local interpolation. In this case the fitted terrain surface consists of planar mosaics and, in general, will be discontinuous (i.e., with finite jumps from one pixel to the next). The approximation is best for Δx and Δy small relative to topographic roughness.

DEMON uses planar surfaces because a single-flow direction within each pixel is simpler computationally. Pixels are defined by grid lines, having grid points at the corners. The elevation tensor in each pixel is given by vector $a\mathbf{i} + b\mathbf{j}$ ($\mathbf{i}$ and $\mathbf{j}$ are unit vectors in the x and y directions, respectively), with

$$\begin{aligned} a &= \frac{1}{2\Delta x}(-z_1 + z_2 + z_3 - z_4) \\ b &= \frac{1}{2\Delta y}(z_1 + z_2 - z_3 - z_4) \end{aligned} \tag{6}$$

where z_1, z_2, z_3, and z_4 are the elevations at the upper left, upper right, lower right, and lower left pixel corners, respectively. Flow is in the direction of vector $-(a\mathbf{i} + b\mathbf{j})$, indicated by the aspect angle α.

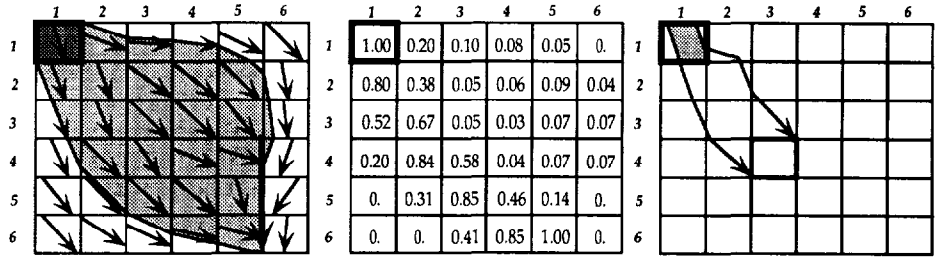
For planar pixels, if the flow direction is parallel to the grid orientation, the exit portion of the boundary is a single full boundary segment. If the flow direction is not parallel to the grid orientation, the exit portion of the boundary consists of two full adjacent boundary segments. The general expression (5) for the flow width becomes

$$\omega = |\sin \alpha|\Delta x + |\cos \alpha|\Delta y \tag{7}$$

3.3. DEMON-Downslope and DEMON-Upslope Algorithms for Computing the SCA and SDA Matrices

DEMON-downslope is a particle-tracking algorithm that projects flow downslope using the matrix of flow angles and allows computing both the SCA and SDA matrices. DEMON-upslope traces the boundary of a pixel's contributing area and calculates the size of the area enclosed by that line. DEMON-upslope computes the SCA matrix faster than DEMON-downslope but does not calculate the SDA matrix because the TDA cannot always be bounded by a single connected line (Figure 1). While different, the two algorithms, DEMON-downslope and DEMON-upslope, provide the same SCA values. The information required by either algorithm is the matrix of flow angles.

3.3.1. DEMON-downslope: SCA computation. If a unit flow depth is generated uniformly everywhere over the area covered by the DEM, the total flow volume drained by any given pixel equals the pixel's TCA. DEMON-downslope



	1	2	3	4	5	6
1	1.00	0.20	0.10	0.08	0.05	0.
2	0.80	0.38	0.05	0.06	0.09	0.04
3	0.52	0.67	0.05	0.03	0.07	0.07
4	0.20	0.84	0.58	0.04	0.07	0.07
5	0.	0.31	0.85	0.46	0.14	0.
6	0.	0.	0.41	0.85	1.00	0.

(a) (b) (c)

Figure 6. (a) Example of two-dimensional, aspect-driven (arrow directions) flow movement. Flow originates over pixel (1,1). The area shaded is the TDA of pixel (1,1). (b) Influence matrix of pixel (1,1). Each entry in the influence matrix represents the fraction of the area within pixel (1,1) that is drained by a pixel. (c) Illustration of the physical meaning of the value 0.58 of pixel (4,3) in Figure 6b.

considers that a unit flow depth is generated uniformly over the surface of every pixel in the DEM. The flow generated over each pixel is tracked downslope in two dimensions (i.e., as a flow tube) according to local terrain aspect, until the flow either leaves the DEM or enters a sink pixel (a "pit"). The total flow drained by a given pixel equals that pixel's TCA.

DEMON-downslope takes one pixel at a time and makes it a source pixel. A unit flow depth is considered to be generated uniformly over the source pixel and is followed downslope. Figure 6a shows two-dimensional aspect-driven flow movement from source pixel (1,1) (we use (i, j) to indicate the pixel corresponding to row i and column j of the DEM). As we track the flow downslope, we compute the flow volume drained by each pixel that the flow traverses. The influence matrix of a source pixel contains the flow volume from the source pixel that is drained by each pixel in the DEM. If a pixel receives no flow from the source pixel, its entry in the influence matrix is zero; if it drains all of the flow, its entry is $\Delta x \Delta y$; and if it drains part of the flow, its entry is a fraction of $\Delta x \Delta y$. Figure 6b is the influence matrix of pixel (1,1) (in this example, $\Delta x = \Delta y = 1$). The entries on any cross diagonal of the influence matrix, before flow convergence to a line occurs, sum to unity. Figure 6c shows the physical location of the portion of the source pixel (shaded area) associated with the entry of pixel (4,3) in the influence matrix of pixel (1,1). Pixel (4,3) drains 58% of the area of pixel (1,1).

The TCA matrix is computed by successive addition of the influence matrix of every pixel in the DEM. In the example DEM of Figure 6, 36 influence matrices, one for each source pixel, are computed and added. The SCA matrix is computed by division of the TCA matrix by the flow width matrix, which is obtained from the matrix of flow angles using (7).

We use Figure 6 to illustrate the three-step procedure for computing the influence matrix of a pixel.

Step 1: Flow generated over the source pixel flows over the pixel surface in the direction indicated by α. If α is a multiple of 90°, all flow goes to a single neighbor. For example, if α is equal to 180°, all flow will enter the western neighbor. If α is not a multiple of 90°, then flow will be split between two cardinal neighbors (S and E in Figure 7). There can be no flow to a diagonal neighbor because contact with diagonal neighbors is through a point (width of flow is infinitesimally small), not a segment. For flow to reach a diagonal neighbor it must cross through a cardinal neighbor. Therefore a diagonal neighbor cannot be a direct receiving pixel. Figure 7 illustrates how flow is split among the eastern and southern neighbors for $\alpha = 292°$ (the flow angle of pixel (1,1) in Figure 6). A flow particle generated in the upper right shaded triangular area, moving at an angle of 292°, must cross the eastern border and enter the eastern neighbor. The area of the triangular section and the fractions f_E and f_S are, for $\alpha = 292°$,

$$f_E = \frac{A_\Delta}{A} = \frac{1}{2}\frac{\Delta y}{\Delta x}\frac{1}{\tan(292°)} = 0.20; \quad f_S = 1 - f_E = 0.80 \tag{8}$$

where A_Δ represents the triangular area in Figure 7, and A represents the pixel area ($\Delta x \Delta y$). Twenty percent of the flow generated over pixel (1,1) is delivered to the E neighbor and 80% to the S neighbor.

Step 2: In Figure 6a, source the S neighbor. has two receiving pixels. To follow flow downslope we must step to one of these two receiving pixels, while the other one is put on a waiting list for later consideration. Consider pixel (2,1), with pixel (1,2) on the waiting list.

Step 3: Figure 8 illustrates the definition of the eight

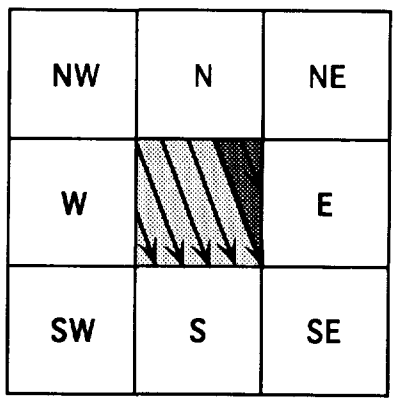


Figure 7. Illustration of how flow from the source pixel is partitioned between two cardinal neighbors. In this case the flow angle in the source pixel is 292°, and flow is partitioned between the eastern and southern neighbor pixels. The fraction of flow that enters the eastern neighbor is the fraction of the area of the source pixel represented by the darker shaded triangular area.

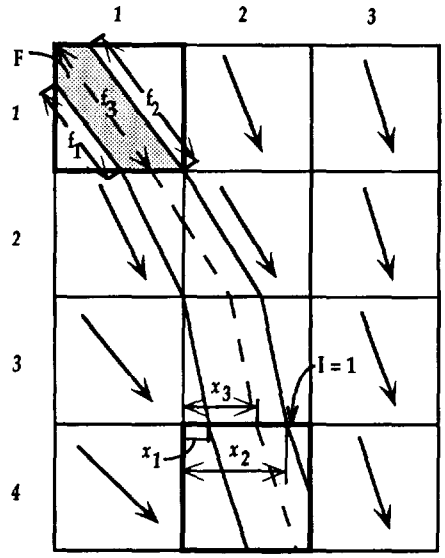


Figure 8. Illustration of the definition of variables F, f_1, f_2, f_3, x_1, x_2, x_3, and I, needed for projecting flow across pixel (4,2). F is the flow entering pixel (4,2) (the shaded area); f_1 and f_2 are the lengths drained by the flow lines bounding the flow tube; I = 1, 2, 3, 4 indicates flow entering from the north, east, south, and west, respectively; x_1 and x_2 indicate the points of entry of the bounding flow lines; x_3 indicates the point of entry of the flow line generated at the corner of the source pixel (dashed curve); $f_3 = f_2$ indicates that all points inside the flow tube to the right of the dashed curve drain the same amount of flow, while the flow drained by any point to the left of the dashed curve varies linearly across the flow tube.

variables (F, f_1, f_2, f_3, I, x_1, x_2, and x_3) required to describe the geometry of flow movement across a pixel (pixel (4,2) in Figure 8). (The DEM in Figure 8 is different from that in the previous figures to facilitate displaying the eight variables.) The source pixel being considered in Figure 8 is (1,1). The flow tube shown is the flow tube that has the path (1,1)-(2,1)-(2,2)-(3,2)-(4,2). Part of the flow carried by this flow tube will enter pixel (4,3) and the remainder will enter pixel (5,2), that is, this flow tube will be split into two flow tubes. The fractions entering (4,3) and (5,2) are computed from geometry, using f_1, f_2, and f_3.

Topographic convergence converts a flow strip (two dimensional) into a line (one dimensional). The vertical bold arrow in the lower right of Figure 6a is an example of one-dimensional flow. One-dimensional flow need not be along pixel boundaries but may cross through pixels. Projecting a line according to flow angle requires only the variables I and x_1 to keep track of the point of entry of the flow line in each pixel [*Lea*, 1992].

The algorithm used to compute the influence matrix of a pixel can follow one flow path at a time, that is, the dispersal area of the pixel is dissected into several flow tubes so that the path of each flow tube can be written as a path from one pixel to the next. The flow tube in Figure 8 can be written as (1,1)-(2,1)-(2,2)-(3,2)-(4,2), which is a directed graph. Figure 9a represents all the (directed graph) flow tube paths that can be written for the example DEM shown in Figure 6. Figure 9b represents these paths schematically; each branch in Figure 9b corresponds to one flow tube in Figure 9a.

3.3.2. DEMON-downslope: SDA computation. The SDA is computed simultaneously with the SCA. As we step to each pixel downslope and project flow through it, the size of the area within the pixel occupied by the flow tube is computed. The summation of all these areas is the TDA of the source pixel (the shaded area in Figure 6a). The SCA matrix is obtained from the TCA matrix through division by the flow width matrix.

3.3.3. DEMON-upslope. The DEMON-upslope algorithm computes the TCA directly for each pixel by tracing the boundary of the pixel's contributing area and calculating the size of the area enclosed by that line. Most pixels have a single source area, whose boundary intercepts two corners of the pixel of interest. It is not possible to start at one pixel corner and trace the entire boundary line until the second pixel corner is reached. This is because for part of the way the tracing direction is against the flow (going upslope), while for the second part, movement is with the flow (going downslope), and it is not known a priori which is the right location to switch from upslope to downslope. Hence it is necessary to trace the boundary line starting at each of its two ends at the pixel corners and always move upslope, until

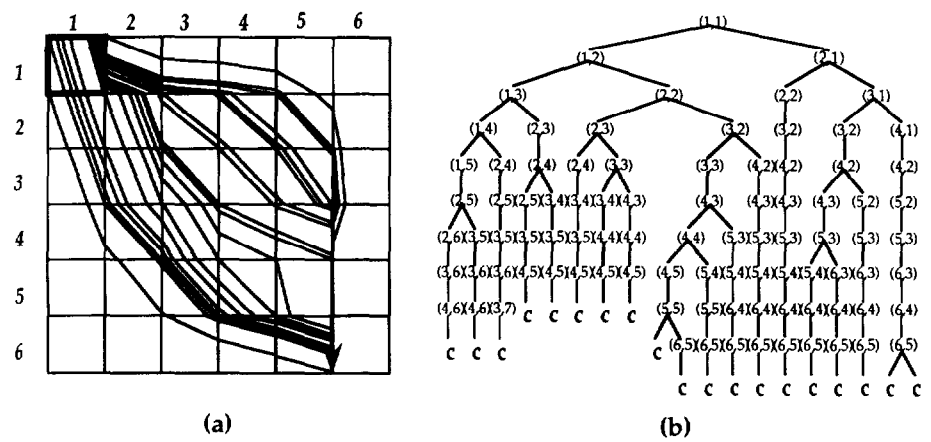


Figure 9. (a) Flow tubes that must be considered to compute the influence matrix of pixel (1,1) in Figure 6. (b) Schematic representation of the flow tubes defined in Figure 9a. Branching corresponds to flow being split between two neighbor pixels. All paths exit the DEM in concentrated form, indicated by "c."

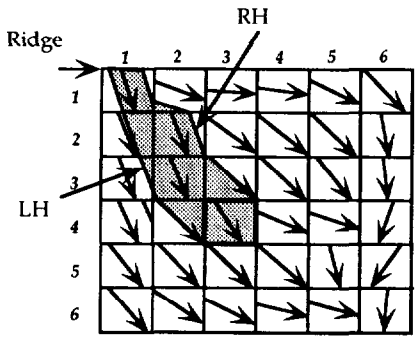


Figure 10. Illustration of the right-hand side (RH) and left-hand side (LH) flow lines that must be traced uphill in the DEMON-upslope algorithm.

the point where the two traced lines meet (Figure 10). Some pixels may receive flow from two source areas, in which case four lines must be traced. The three-step procedure for computing the TCA of a pixel is as follows:

Step 1: Determine which of the four cardinal neighbors discharge some flow to the pixel of interest. If a cardinal neighbor drains into the pixel of interest, an indicator variable for that neighbor is set to 1 (0 otherwise). Next, the indicator variables of the four neighbors are checked in clockwise order, and if a switch from 0 to 1 or from 1 to 0 from one neighbor to the next is found, then there is a boundary line that starts at the corner point shared by those two neighbors.

Step 2: Trace the TCA boundary lines uphill. If a pixel has more than one contributing area, this procedure is repeated for each. The boundary line consists of two lines that join uphill; the left-hand (LH) line bounds the contributing area from the left, and the right-hand (RH) line bounds the contributing area from the right as paths are followed upslope from the pixel. The RH line is traced from the pixel of interest by moving continuously against the flow direction. The coordinates of the points at which the line intercepts the boundary of each pixel are stored. The RH line is traced for some large, preestablished, number of pixels n_p (e.g., equal to 3 times the typical hillslope length for the study area). The LH line is traced uphill in the same way, until the RH line is met. While tracing the RH or LH line, often a local elevation maximum is found. In such cases it is not possible to move further upslope, so the path follows a line of divergence, that is, a line where flow directions oppose each other on each side.

Step 3: After the RH and LH lines have been mapped, the TCA boundary is known. The TCA is calculated from the area enclosed by the boundary lines which is determined from the stored coordinates that describe their paths.

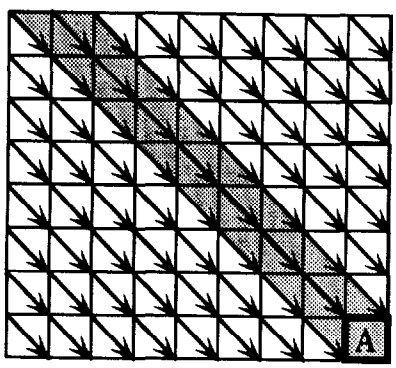


Figure 11. TCA of a pixel on a planar slope with aspect angle 315° predicted by DEMON.

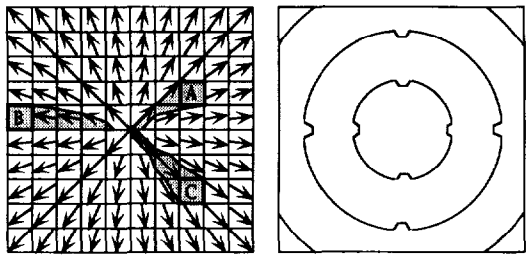


Figure 12. TCA of pixels on a right circular cone mountain surface (or TDA of pixels on a right circular cone crater surface) predicted by DEMON. (a) TCA of pixels A, B, and C. The prediction is poorest for pixel B, which is overestimated as a result of allowing for a single flow direction in each pixel. (b) SCA contours.

4. Results

We present results from use of DEMON for a plane, a right circular cone mountain, and a right circular cone crater (Figures 11–13, respectively) to illustrate model performance for parallel, convergent, and divergent flow conditions. Figures 11–13 may be compared with Figures 2–4, which show the corresponding results using D8; and Figures 11, 12a, and 13a may be compared with Figures 5a–5c, which show the corresponding results using multiple-direction algorithms. For the plane, computed SCAs correspond correctly to the distance to the top of the plane. True SDAs for a right circular cone mountain are the same as the SCAs for a right circular cone crater and vice versa. For both cones the predicted SCA and SDA isolines are approximately circular with indentations to the north, east, south, and west. These indentations are larger for the crater cone than for the mountain cone, and in both cases they become more pronounced with distance to the cone center. The indentations result from the approximation of the conical surface by a mosaic of planes, which generates a bias toward the N-S and E-W directions. The indentations can be avoided only if a

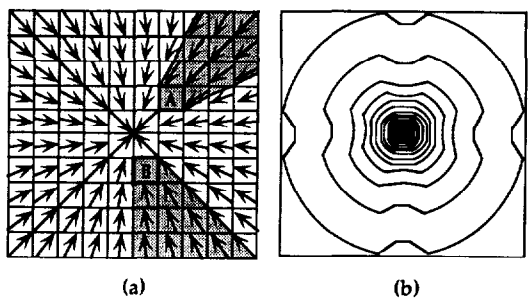


Figure 13. TCA of pixels on a right circular cone crater surface (or TCA of pixels on a right circular cone mountain surface). (a) TCA of pixels A and B predicted by DEMON. (b) SCA contours.

curved rather than a planar surface is fitted to each pixel; using DEMON, the computational expense for nonplanar surface fitting would be large. Significant errors may result on concave or convex hillslopes that are long relative to grid size, that is, comprising a large number of pixels. For a conical surface the magnitude of the error a distance of 40 pixels away from the cone center is approximately 15%. For a 10-m grid DEM, 40 pixels represents a physical distance of 400 m.

To illustrate model performance for natural topography, we present results of computed SCAs and SDAs for the pixels of a DEM for Mettman Ridge, in southern Oregon [*Zhang and Montgomery*, 1994]. This DEM has a 2 × 2 m grid covering an area of 720 × 900 m^2 (360 × 450 pixels) and was obtained by interpolation of a 1:4800 scale topographic map [*Montgomery*, 1991]. Field mapping of the channel network revealed that the land surface differed locally from the DEM [*Montgomery and Dietrich*, 1994, Figure 4(b)]. The SCA and SDA matrices obtained with DEMON are represented as images in Figures 14 and 15. Figure 16 shows the SCAs computed using D8. DEMON represents hillslope aspect better, while in D8 the erroneous tendency toward preferred directions is obvious. Also, curved flow paths are represented by DEMON but not D8.

In Figure 17 we plot the computed SCA values for each pixel using DEMON and D8. Only pixels belonging to basins that lie fully inside the DEM are represented, to eliminate any DEM boundary effects. The disagreement between computed SCAs spans 5 orders of magnitude at the hillslope scale. The two methods agree in many pixels with a high SCA value because no flow is lost in either routing model, so

Figure 15. Image of the SDA matrix for the Mettman Ridge study area obtained with DEMON. A logarithmic scale of gray shades is used; lighter shades correspond to higher values.

Figure 14. Image of the SCA matrix for the Mettman Ridge study area obtained with DEMON. A logarithmic scale of gray shades is used; lighter shades correspond to higher values.

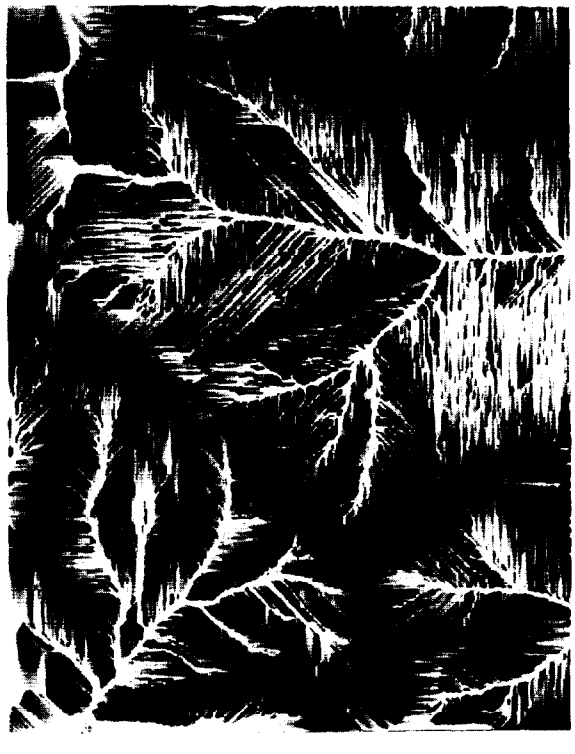


Figure 16. Image of the SCA matrix for the Mettman Ridge study area obtained with D8. A logarithmic scale of gray shades is used; lighter shades correspond to higher values.

that topographic convergence eventually leads to the same value at the outlet of each basin. The large differences found at the hillslope scale are attributed principally to D8 errors similar to those illustrated in Figures 2–4.

Figure 18 shows the histograms of the SCAs smaller than 100 m computed with D8 and DEMON. The D8 histogram is discontinuous, while the DEMON histogram is continuous and smooth, as is to be expected over natural hillslopes. The discontinuous D8 histogram reflects the fact that D8 TCAs are integers, and flow width has only three possible values (Δx, Δy, or $(\Delta x^2 + \Delta y^2)^{1/2}$).

Figure 19 shows pixels with SCAs smaller than 4 m (i.e., draining themselves and one other pixel), for DEMON and for D8. DEMON gives low values almost exclusively at or near hillslope tops, while D8 gives low values at many locations on the hillslopes. This is because some pixels are missed by the one-dimensional flow paths of method D8 where flow is divergent (such as pixel A in Figure 3a).

Figure 20 shows pixels with SCAs higher than 1000 m, for DEMON and for D8. There are significant differences between the two drainage configurations. Overall, the D8 network has straighter segments. Network structure is different at a location near the lower left-hand corner (see arrow). Two D8 flow paths run parallel at a very short distance from each other at a location in the central upper half of the study area (see arrow). The high-SCA network configuration computed with DEMON is in better agreement with the field-surveyed channel network [*Montgomery and Dietrich*, 1994, Figure 4(b)].

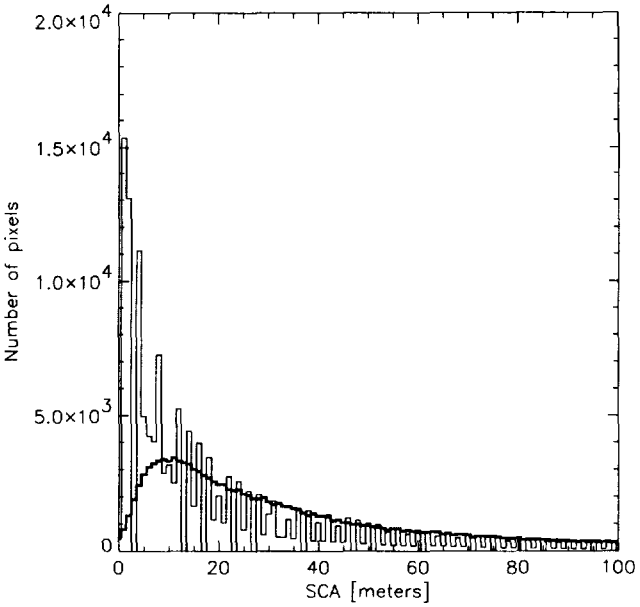


Figure 18. Histogram of SCA values obtained with methods D8 (finer curve) and DEMON (heavier curve) for values lower than 100 m.

5. Conclusions

We have shown that the models presently in widest use for computing the specific contributing area of rectangular grid DEM pixels, those based on flow-routing method D8 [*O'Callaghan and Mark*, 1984], can produce seriously erroneous results for either planar, convergent, or divergent topography. The most important source of errors is the point source assumption and the resulting one-dimensional flow routing (each pixel discharges into only one neighbor pixel). Another source of error is the restriction on the flow direction in each pixel to only eight possibilities (multiples of 45°), established by the orientation of the sampling grid. More recent models reviewed attempt to overcome the limitations of method D8, with partial success. *Lea* [1992] provides a sound scheme for aspect-driven routing which is an improvement over D8 routing. However, no satisfactory solution has been presented for the most important limitation of method D8, the point source assumption and one-dimensional routing. This problem has been addressed with partial success by multiple-direction models. The most important limitations of multiple-direction models are that the computed contributing areas are discontiguous, and the quality of the approximation of the computed values relies on surface geometric symmetry and is affected by boundary proximity.

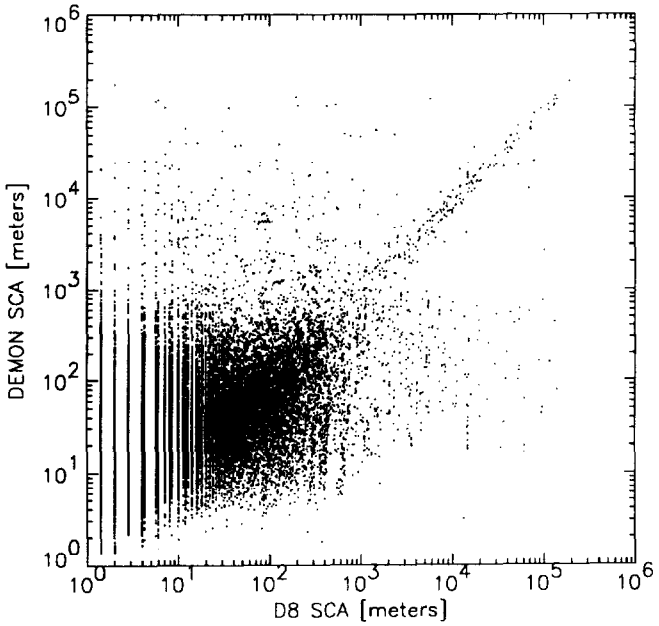


Figure 17. Plot of SCA values for the Mettman Ridge study area: D8, horizontal axis, and DEMON, vertical axis. One fifth (32,400) of all pixels are shown to avoid clutter.

DEMON computes the specific contributing areas and specific dispersal areas of DEM pixels. Assignment of flow directions is according to aspect angle, as given by *Lea* [1992]. Flow path routing is two dimensional, allowing representation of the effect of terrain topography on flow path width. Flow path width remains constant over planar terrain, increases over divergent topography, and decreases over convergent topography. Thus the proposed model has capabilities which at present are offered only by contour-based flow models, while having the convenience of using rectangular grid DEMs.

Results for geometric surfaces for which SCAs can be computed analytically show that DEMON approximates analytic values, while D8 has large errors for either parallel, convergent, or divergent flow. Computed SCAs for the Mettman Ridge DEM using D8 and DEMON differ by 5 orders of magnitude at the hillslope scale. The two methods

have better agreement at high SCA values, because flow conservation leads to similar values after flow is concentrated into linear paths by terrain convergence. The large errors at the hillslope scale indicate that D8 is inappropriate for hillslope applications, including studies of hillslope hydrologic response, channel head location, long-term basin evolution, landslide risk, and soil water content.

We presented two equivalent algorithms for computing SCAs: DEMON-downslope and DEMON-upslope. DEMON-downslope provides more information than DEMON-upslope. The particle-tracking approach of DEMON-downslope may be used for surface sediment or pollutant tracking. While both algorithms can be adapted to accommodate distributed values of runoff generation, only DEMON-downslope can permit reinfiltration. DEMON-downslope can also provide SDA values for individual flow tubes within a pixel, which may constitute useful information for a large grid size. Finally, DEMON-downslope distinguishes between dispersed (two dimensional in plan view) and concentrated (one-dimensional) flow, information with potential utility for models of rainfall-runoff, soil water content, and other applications.

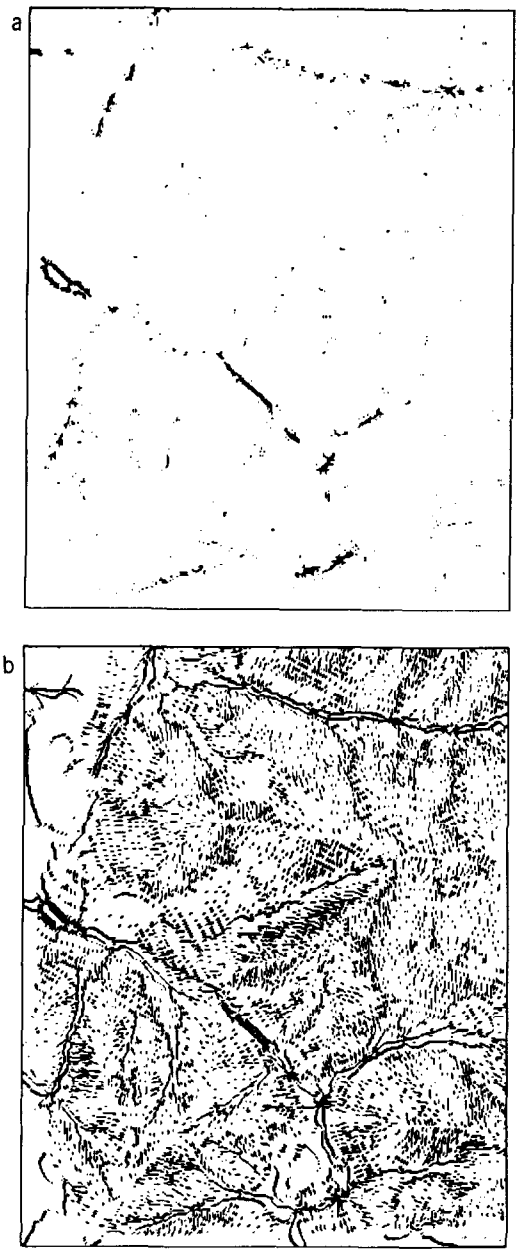


Figure 19. Image of the SCA values less than 4 m, that is, pixels that drain no more than the area of one pixel other than themselves for (a) DEMON and (b) D8. Low values of SCA are located over large ridges for DEMON but are ubiquitous on the hillslopes for D8.

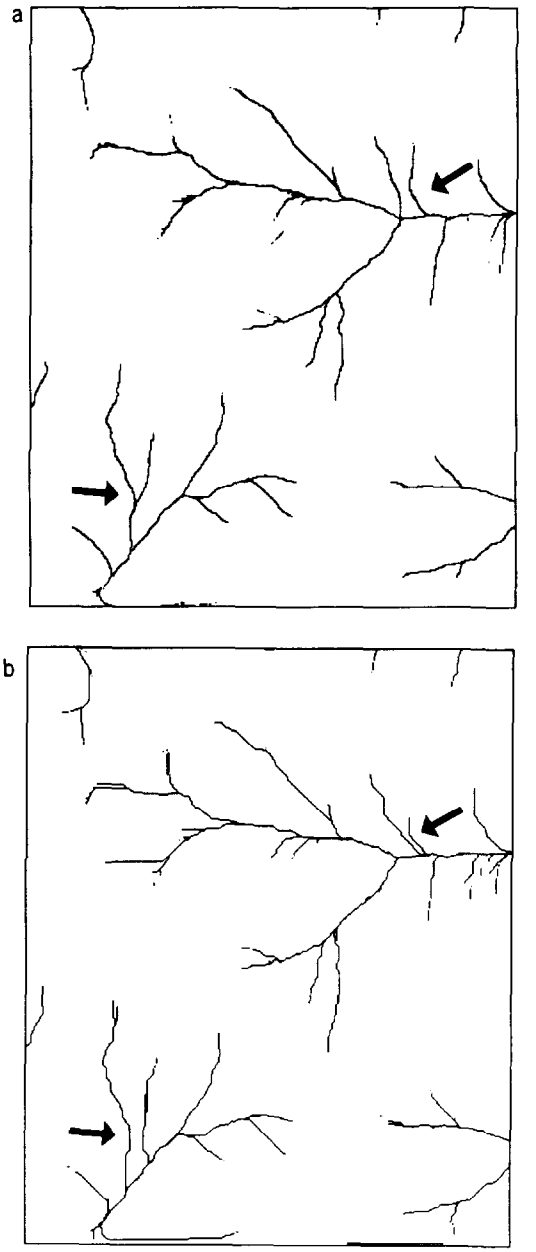


Figure 20. Image of the SCA values larger than 1000 m for (a) DEMON and (b) D8. (In some locations the networks are discontinuous because the pits in the original DEM were not removed.)

Acknowledgments. This work is dedicated to the memory of Ian Moore (1951–1993). We appreciated his generosity and selflessness in the help he provided during preparation of our manuscript. We are all the richer for the many contributions he made during his too

short professional life. We are all the poorer for the loss of his intellectual leadership, energy, enthusiasm, and collegiality. This work has been supported in part by the "Junta Nacional de Investigação Científica e Tecnológica," Portugal, under award BD-1480-91, and by the U.S. Geological Survey (USGS), Department of the Interior, under USGS award 14-08-0001-G1887. The views and conclusions contained in this document are those of the writers and should not be interpreted as necessarily representing the official policies, either expressed or implied, of either the U.S. or Portuguese governments. We thank David Montgomery, of the University of Washington, for helpful discussion and providing the Mettman Ridge DEM. We thank David Tarboton and an anonymous reviewer for their valuable critiques of this manuscript. The DEMON codes are available from the authors upon request to colleagues who provide an e-mail address.

References

Beven, K. J., and M. J. Kirkby, A physically-based variable contributing area model of basin hydrology, *Hydrol. Sci. Bull.*, *24*, 43–69, 1979.

Fairfield, J., and P. Leymarie, Drainage networks from grid digital elevation models, *Water Resour. Res.*, *27*(5), 709–717, 1991. (Correction, *Water Resour. Res.*, *27*(10), 2809, 1991.)

Freeman, T. G., Calculating catchment area with divergent flow based on a regular grid, *Comput. Geosci.*, *17*(3), 413–422, 1991.

Lea, N. L., An aspect driven kinematic routing algorithm, in *Overland Flow: Hydraulics and Erosion Mechanics*, edited by A. J. Parsons and A. D. Abrahams, Chapman and Hall, New York, 1992.

Mark, D. M., Network models in geomorphology, in *Modelling Geomorphological Systems*, edited by M. G. Anderson, pp. 73–97, John Wiley, New York, 1988.

Montgomery, D. R., Channel initiation and landscape evolution, Ph.D. dissertation, 421 pp., Dep. of Geol. and Geophys., Univ. of California, Berkeley, 1991.

Montgomery, D. R., and W. E. Dietrich, Source areas, drainage density, and channel initiation, *Water Resour. Res.*, *25*(8), 1907–1918, 1989.

Montgomery, D. R., and W. E. Dietrich, Channel initiation and the problem of landscape scale, *Science*, *255*, 826–830, 1992.

Montgomery, D. R., and W. E. Dietrich, A physically based model for the topographic control of shallow landsliding, *Water Resour. Res.*, *30*(4), 1153–1171, 1994.

Montgomery 1R., and E. Foufoula-Georgiou, Channel network source representation for digital elevation models, *Water Resour. Res.*, *29*(12), 3925–3934, 1993.

Moore, I. D., and R. B. Grayson, Terrain-based catchment partitioning and runoff prediction using vector elevation data, *Water Resour. Res.*, *27*(6), 1177–1191, 1991.

Moore, I. D., E. M. O'Loughlin, and G. J. Burch, A contour-based topographic model for hydrological and ecological applications, *Earth Surf. Processes Landforms*, *13*, 305–320, 1988.

O'Callaghan, J. F., and D. M. Mark, The extraction of drainage networks from digital elevation data, *Comput. Vision Graphics Image Process.*, *28*, 323–344, 1984.

Quinn, P., K. Beven, P. Chevalier, and O. Planchon, The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models, *Hydrol. Processes*, *5*, 59–79, 1991.

Speight, J. G., A parametric approach to landform regions, *Spec. Publ. Inst. Br. Geogr.*, *7*, 213–230, 1974.

Speight, J. G., The role of topography in controlling throughflow generation: A discussion, *Earth Surf. Processes Landforms*, *5*, 187–191, 1980.

Tarboton, D. G., R. L. Bras, and I. Rodriguez-Iturbe, The analysis of river basins and channel networks using digital terrain data, *Tech. Rep. 326*, 251 pp., Ralph M. Parsons Lab., Dep. Civ. Eng., Mass. Inst. of Technol., Cambridge, 1989.

Tarboton, D. G., R. L. Bras, and I. Rodriguez-Iturbe, On the extraction of channel networks from digital elevation data, *Hydrol. Processes*, *5*, 81–100, 1991.

Zhang, W., and D. R. Montgomery, Digital elevation model grid size, landscape representation, and hydrologic simulations, *Water Resour. Res.*, *30*(4), 1019–1028, 1994.

S. J. Burges and M. C. Costa-Cabral, Department of Civil Engineering, FX-10, University of Washington, Seattle, WA 98195.

(Received May 19, 1993; revised November 30, 1993; accepted December 13, 1993.)