



Water Resources Research

RESEARCH ARTICLE

10.1029/2019WR026507

Key Points:

- A new nondispersive eight-direction algorithm to simulate drainage directions is developed
- The algorithm approximates theoretical drainage directions through flexible triangular facets
- A flow aggregation technique iterating global deviations of local drainage directions is purposed

Correspondence to:

J. Liu,
jliu@hhu.edu.cn

Citation:

Wu, P., Liu, J., Han, X., Liang, Z., Liu, Y., & Fei, J. (2020). Nondispersive drainage direction simulation based on flexible triangular facets. *Water Resources Research*, 55, e2019WR026507. <https://doi.org/10.1029/2019WR026507>

Received 8 OCT 2019

Accepted 26 MAR 2020

Accepted article online 4 APR 2020

Nondispersive Drainage Direction Simulation Based on Flexible Triangular Facets

Pengfei Wu^{1,2} Jintao Liu^{1,2} Xiaole Han^{1,3}, Zhongmin Liang² Yangyang Liu^{1,2}, and Junyuan Fei^{1,2}

¹State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering, Hohai University, Nanjing, China,

²College of Hydrology and Water Resources, Hohai University, Nanjing, China, ³School of Earth Sciences and Engineering, Hohai University, Nanjing, China

Abstract A new eight-direction algorithm (iFAD8) for the simulation of drainage directions on grid digital elevation models (DEMs) is presented. In this algorithm, a flexible triangular facet construction technique ($ND\infty$) is developed to provide local drainage directions ranging continuously from 0° to 360° . Subsequently, a flow aggregation technique using global deviations of local drainage directions is proposed to simplify the flow paths into a nondispersive format. Another algorithm (FAD8) accompanying the iFAD8 is also presented, which uses the $D\infty$ directions as the local drainage directions. Then FAD8, iFAD8, and three existing algorithms are compared on 10 abstract terrains including nine basic terrains and a Himmelblau terrain. The algorithms are tested to reproduce exact slope lines and specific catchment areas derived from the terrain functions. The results show that iFAD8 has better performance than FAD8. Both iFAD8 and FAD8 outperform existing algorithms in most cases. The flow aggregation technique is shown to be an excellent choice for nondispersive drainage direction simulation based on infinite directions. The $ND\infty$ direction based on flexible triangular facets is also an improvement to the local infinite direction of $D\infty$. We conclude that the iFAD8 algorithm can provide better definitions of drainage directions.

Plain-language Summary Overland movement of water and soil plays an important role in hydrological cycle and geomorphologic evolution. Analysis and modeling of these processes rely on drainage direction information. Through digitalizing and discretizing a continuous terrain into a mass of grid cells, drainage direction can be assigned to each cell. The accuracy of drainage directions may have a tremendous impact on the hydrologic and geomorphologic modeling. So an advanced algorithm for determination of drainage direction is required. Many algorithms have been proposed for this purpose. But there are some shortcomings in the existing algorithms. Here we present a new algorithm named iFAD8 to provide single drainage direction for each topographic cell. Two newly developed techniques are adopted in iFAD8 to improve both local drainage directions and global flow paths. iFAD8 partly optimizes the shortcomings of existing algorithms in theory and has a better performance on various terrains. Finally, iFAD8 will benefit hydrologic and geomorphologic modeling.

1. Introduction

Drainage direction for overland flow or soil creep routing is an important hydrological or geomorphological parameter. Many mathematical models, such as distributed surface runoff models (e.g., Camporese et al., 2010; Ciarapica & Todini, 2002) and soil evolution models (e.g., Dietrich et al., 1995; Liu et al., 2013), employ a fixed drainage network to describe the downstream movement trajectories on the grid digital elevation models (DEMs). Several necessary parameters of the models mentioned above, such as the catchment boundary, the flow distance, and the specific contributing area (SCA), can be obtained from the drainage network (Tarboton, 1997; Yamazaki et al., 2019).

Drainage direction can be approximated by two types of approaches, namely, the dispersive or nondispersive algorithms (Orlandini & Moretti, 2009a; Pilesjö & Hasan, 2014; Zhou et al., 2011). The dispersive algorithms are designed to represent flow paths driven by gravity as well as turbulence or diffusional effects (Orlandini et al., 2012) and allow a DEM cell to drain to multiple downslope cells (e.g., Costa-Cabral & Burges, 1994; Freeman, 1991; Pilesjö & Hasan, 2014; Quinn et al., 1991; Seibert & McGlynn, 2007; Tarboton, 1997). The nondispersive algorithms approximate flow paths purely driven by gravity with other forces ignored and

assign every DEM cell only one drainage direction (Orlandini et al., 2012). Here the gravity-driven nondispersive flow paths are always coincident with slope lines (Maxwell, 1870), which follow the direction of steepest descent across the land surface and are perpendicular to contour lines everywhere (Bonetti et al., 2018; Gallant & Hutchinson, 2011).

In existing studies, the dispersive algorithms can enhance the simulation of certain variables, such as the SCA (Pilesjö & Hasan, 2014; Zhou & Liu, 2002) and the topographic wetness index (Han et al., 2018; Quinn et al., 1991, 1995). However, the physical dispersion may not obey the same laws as artificial dispersion by dispersive methods because it is affected by a combination of atmospheric forcing, land surface topography, and surface-subsurface flow interaction. (Tarboton, 1997; Orlandini & Moretti, 2009a; Orlandini et al., 2012). In addition, difficulties and inefficiencies will arise by using dispersive algorithms when outlining the catchment boundary (Ariza-Villaverde et al., 2013), estimating the flow distance (Liu et al., 2012), and modeling sheet or rivulet flow (Orlandini et al., 2012). That is to say, dispersive techniques make it difficult to define catchment scale flow paths for runoff modeling. Hence, a nondispersive single drainage direction algorithm is preferred when specific adequate flow path lines for all cells inside the region are needed (Paik, 2008).

O'Callaghan and Mark (1984) proposed the first approach named the eight-direction (D8) algorithm for the judgment of nondispersive drainage directions over grid DEMs. The D8 algorithm employs the direction of the steepest downward slope within eight possible directions toward surrounding cells. This algorithm has been included in many popular GIS packages and used widely for its simplicity (Ariza-Villaverde et al., 2013). However, there remains a disadvantage limiting the application of D8; that is, the final direction is limited to eight possible directions, while the possible deviation between the real steepest direction and the final direction is not considered (Fairfield & Leymarie, 1991; Lea, 1992; Paik, 2008).

To improve the performance of traditional nondispersive algorithms such as D8, Fairfield and Leymarie (1991) introduced randomness into the drainage direction definition. However, the randomness would produce unreproducible results (Costa-Cabral & Burges, 1994). Lea (1992) took the aspects of fitted planes as the final directions and routed the flow as though it was a ball released from the cell center and rolling across the topographic surface. Tarboton (1997) proved that fitted planes by Lea (1992) may give adjacent cells opposing directions. Moreover, modeling the rolling path for every cell may result in a downstream cell crossed by multiple nonoverlapping paths. Zhou et al. (2011) adopted this rolling procedure of Lea (1992) into their own algorithm and declared that substantial computing and storage resources are required for the abundant nonoverlapping rolling paths.

Tarboton (1997) proposed the dispersive D ∞ algorithm, which uses the steepest downslope direction of eight planar triangular facets within a 3×3 window to specify the drainage direction. The D ∞ algorithm provides a reasonable compromise between the simplicity of D8 and the sophistication introduced in other existing algorithms to improve the precision of drainage directions (Orlandini et al., 2003). To accommodate the advantage of D ∞ into the nondispersive algorithm, some nondispersive algorithms employ the D ∞ algorithm to provide the local drainage directions (LDDs) ranging continuously from 0° to 360° for every cell (e.g., Orlandini et al., 2003; Shin & Paik, 2017). The LDDs are assumed to be equal to the theoretical drainage directions (TDDs), which are perpendicular to contours. Thus, these algorithms adjust the final directions relying on LDDs. For example, Orlandini et al. (2003) published a path-based algorithm called D8-LTD that selects the final drainage direction from eight permitted directions. In the algorithm, the least transversal deviations (LTDs) between the draining cells and the D ∞ directions are accumulated to adjust the selected directions along the flow path. Paik (2008) introduced a global search algorithm (GD8) to provide reasonable flow paths. Then an improved version of GD8, namely, iGD8, was demonstrated by Shin and Paik (2017). However, both D8-LTD and iGD8 correct its final drainage direction by taking account of the bias between the D ∞ directions of local cells and their eight possible directions along only one uniquely designated path. For instance, while adjusting the direction for a cell where multiple paths converge, only the bias of the pathway with the largest upstream drainage area for the given cell is adopted by D8-LTD (Orlandini et al., 2003), while iGD8 merely considers the path of the source cell with the highest altitude (Shin & Paik, 2017). That is to say, only partial information from upstream can be used in the determination of drainage directions. Furthermore, in Shin and Paik (2017), a few oversteered or understeered flow paths extracted by D8-LTD and iGD8 can be observed from the concave or convex plate applications, because D ∞ tends to cause biases on curved terrains (Hooshyar et al., 2016).

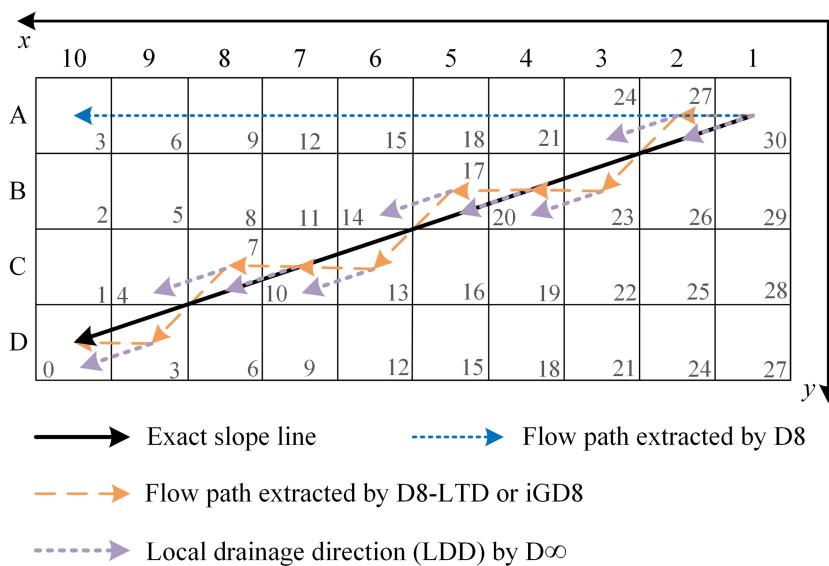


Figure 1. Errors in flow paths extracted by D8. The elevation (z) for every cell on the planar plate, which is displayed as the gray number, is defined by the function $z = 34 - 3x - y$, where $1 \leq x \leq 10$ and $1 \leq y \leq 4$. This plate has a gradient ratio (3:1) between the x and y direction everywhere. Hence, theoretically the D8 direction for every grid is pointed to the x direction. According to the terrain function, the exact slope line out of A1 is the black solid line reaching D10, and D8 provides a wrong straight path from A1 to A10. A more reasonable zigzag flow path (A1 → A2 → B3 → B5 → C6 → C8 → D9 → D10) surrounding the exact slope line can be extracted by D8-LTD or iGD8 that employs the D_{∞} directions as the local drainage directions.

In this paper, a new nondispersive drainage direction algorithm is developed. To improve the performance of D_{∞} on curved terrains, a flexible triangular facet construction technique is taken to determine the LDDs. Then a flow aggregation technique is introduced to assemble the LDDs together and provide continuous nondispersive flow paths. Finally, abstract terrains with different complexities and resolutions are used for a comparison between the proposed algorithm and existing algorithms.

2. The Nondispersive Algorithm

2.1. Background

The D8 algorithm narrows down the search extent to a 3×3 window to guarantee its simplification, but the local search window cannot provide the global optimal results (Paik, 2008). If we just consider the slope of eight directions for the central cell under the D8 rule, false results may rise up. The planar plate in Figure 1 is used to illustrate this problem. The plate has the same aspect everywhere, which is $\arctan(1/3)$ rad (i.e., 18.4°) inclined from the x direction. The exact slope line (e.g., A1 → D10) is a smooth line along the aspect, and it cannot be perfectly represented by D8 directions as limited by the grid structure of DEMs. For example, the D8 algorithm approximates the aspect to 0° and assigns every cell a drainage direction pointing to the x direction in Figure 1. So the D8 directions do not always reproduce the TDDs. More importantly, when the drainage direction (A1 → A2) diverges from the theoretical path, the D8 algorithm would never attempt to force the flow to return to the theoretical path and the flow drains farther and farther away from the theoretical path. Hence, we should determine the drainage direction on the basis of the global flow path.

Some existing nondispersive algorithms, including D8-LTD and iGD8, determine the flow path based on global information (Orlandini et al., 2003; Shin & Paik, 2017). These algorithms accumulate the deviations between the final directions chosen from eight permitted directions and the LDDs along the path. In Figure 1, if the flow drains out of A1 along the LDD, it may cross a neighboring cell (A2) but cannot pass its cell center. Here A1 is treated as the highest cell within the DEM and the source cell of the flow path with the largest upstream drainage area. Thus, both D8-LTD and iGD8 record the deviation between the LDD of A1 and the center of A2 with different deviation compute modes. If the outflow of A2 drains from the A2

center along its LDD, it will reach A3. However, D8-LTD or iGD8 will consider the deviation inherited from A1 and choose B3 instead of A3 as the downstream cell of A2. Finally, both D8-LTD and iGD8 can provide the same zigzag flow path (the yellow flow path in Figure 1) surrounding the exact slope line from A1 in this example.

In fact, there are two kinds of flow for a downstream cell (Lea, 1992). One of them originates from the cell itself and locates at the cell center. The others come from different upstream cells along LDDs and may not reach the cell center. Limited by the nondispersing principle, all the flow (including the flow originating from the cell itself or coming from upstream) must be aggregated together and drained to downstream. As shown in Figure 1, for a downstream cell such as B3, both D8-LTD and iGD8 consider the deviation of flow from only one neighboring upstream cell (e.g., A2). It is obviously that the effects of other inflows from different upstream cells have been ignored by these existed methods. However, a combination of deviations from different inflows may provide extra information for the aggregation. Orlandini et al. (2003) attempted to employ a constant dampening factor to weaken the deviation from the largest upstream area, but the result showed that the best choice was to completely inherit the deviation. In this study, our algorithm uses a technique that takes into account all the inflows in the determination of the deviations for the aggregated flow. The technique can be seen as an enhancement of D8-LTD, which uses a dynamic dampening factor instead of the constant. Thereupon, a synthetical deviation is used to determine the beginning position of the aggregated flow. Then the aggregated flow is released at the beginning position along the LDD of the given cell. The aggregated flow may also miss the downstream cell center, so the flow in the downstream cell needs the aggregation once more.

This algorithm is developed for DEMs without depressions and flats, so a preprocessing is needed to remove these errors in the original DEMs. According to the conception above, the drainage directions of DEM cells are estimated from upstream cells to downstream cells in turn. All cells in a given DEM should be sorted into a queue with elevations from high to low. Then the LDDs as well as the final directions are determined for cells in the queue from upstream to downstream successively. As is known, D8-LTD and iGD8 employ the D_∞ method to determine the LDDs. Here a new method, namely, the new infinite direction algorithm (ND ∞), is developed for derivations of LDDs. ND ∞ is based on the flexible facet construction and improves the performance of D_∞ on curved terrains (see for details in section 2.2). Our algorithm is an eight-direction method because only one final direction is chosen through a flow aggregation procedure. A D_∞ -based algorithm using the same aggregation method is also proposed for verifications and comparisons in this study. We named our D_∞ -based eight-direction algorithm relying on the flow aggregation procedure as FAD8. The ND ∞ -based algorithm is an improvement version to FAD8, and we called it iFAD8. The flow aggregation, nondispersing drainage direction determination, and deviation calculation methods are introduced in section 2.3.

2.2. New Infinite Direction Method Based on Flexible Facets

It is common to generalize the local curved terrains into planes and calculate their aspects to determine the LDDs (e.g., Costa-Cabral & Burges, 1994; Lea, 1992; Seibert & McGlynn, 2007; Tarboton, 1997; Zhou et al., 2011). Triangular facet technique is always adopted for derivation of aspects as three points are enough to determine a plane (Tarboton, 1997). For example, through determining the aspect of a planar triangular facet that is formed by two downslope cell centers as well as itself center, the LDD of a given cell can be approximated in the D_∞ algorithm. Subsequently, Zhou et al. (2011) proposed a technique to calculate the aspect and slope of a facet with three random cell centers. In fact, different algorithms for the triangular facet construction can lead to different definitions of drainage directions (e.g., Pilesjö & Hasan, 2014; Tarboton, 1997; Zhou et al., 2011), and there is no widely accepted method to stipulate how to choose the three cells to form a facet. The deviations in aspects of facets formed by different cells are highly related with the response of vertical concavity or planar divergence. It is hard to evaluate which two downslope cells are the best choices for the facet construction. So we take a common mathematical scheme to select more cells to form more facets in aspect calculations. And then an average value of these aspects is adopted as the final direction, namely, the ND ∞ direction. However, the grid structure and distances between DEM cells have limited the number of possible selectable downslope cells. Moreover, it is not that the more cells adopted, the more accurate the average aspect. As more topographic information and local incised channel features will be ignored if larger facet formed by cells far away from each other is

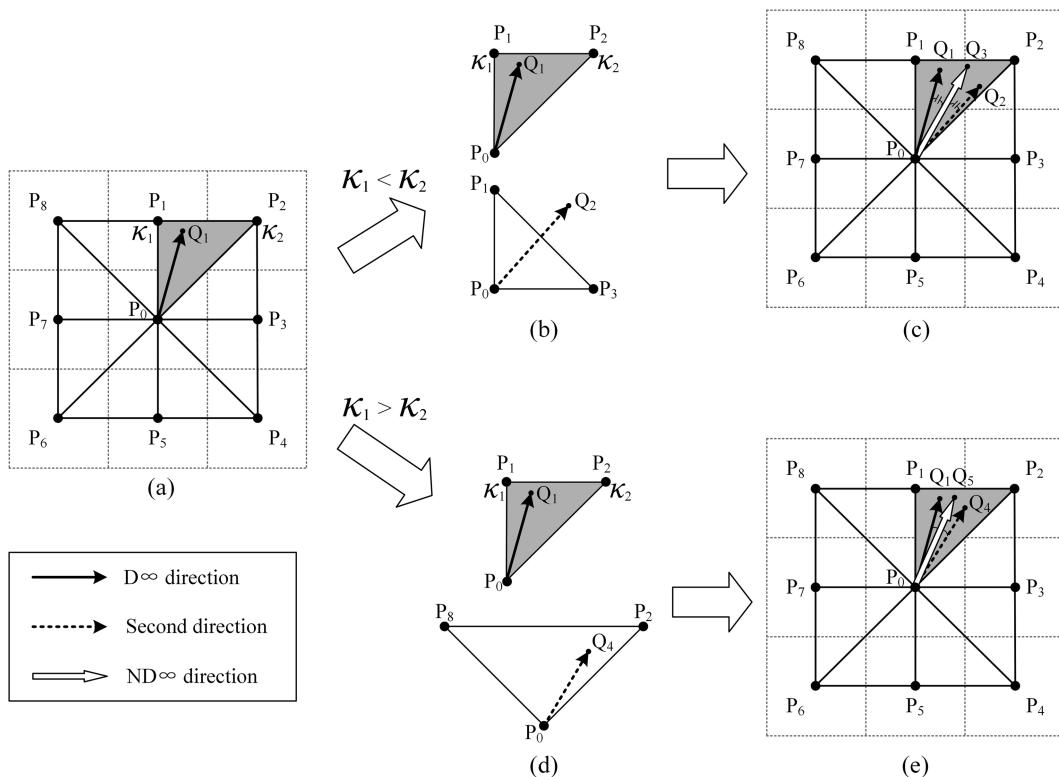


Figure 2. (a) The D_∞ direction ($\overrightarrow{P_0Q_1}$) is used to determine the possible range of the ND_∞ direction, which is marked in gray. Tangential curvatures (κ_1 and κ_2) of two downslope cells are calculated. (b) If $\kappa_1 < \kappa_2$, P_3 is chosen to form a new triangular facet with P_0 and P_1 , and the aspect of the facet is chosen as the second direction ($\overrightarrow{P_0Q_2}$). (c) The ND_∞ direction ($\overrightarrow{P_0Q_3}$) is determined as the average direction of the D_∞ direction and the second direction. (d) If $\kappa_1 > \kappa_2$, P_8 is chosen to form a new triangular facet with P_0 and P_2 , then the second direction ($\overrightarrow{P_0Q_4}$), and (e) the ND_∞ direction ($\overrightarrow{P_0Q_5}$) is determined as mentioned above.

adopted (Kienzle, 2004; Quinn et al., 1991), that is, local terrain relief is flattened. Thus, the aspect of the facet may diverge farther from the TDD. Hence, only three neighboring cells of a central cell closest to the potential TDD were chosen to form two triangular facets in determining the LDD for the central cell in ND_∞ .

Assuming that flow moves over a triangular facet, the drainage direction should be inconsistent with the aspect and along the steepest edge of the facet when the aspect angle is outside the facet (Tarboton, 1997). With two facets used in our algorithm, the final ND_∞ direction is forced to be inside the smaller facet. So a possible range is estimated for ND_∞ firstly. We form the facets according to the D_∞ algorithm (see in Figure 2a), and then the D_∞ direction for the central cell (P_0) is estimated ($\overrightarrow{P_0Q_1}$). Here, P_0-P_8 in Figure 2 are the central points of their cells, and these numbers are also used to represent the cells where they are placed in this context. Then the angle of the facet where the D_∞ direction follows on is set as the possible range for the ND_∞ direction, namely, the gray facet in Figure 2a. If the D_∞ direction is coincident with a cardinal or diagonal direction, there must be two neighboring facets containing this direction. In this case, we select the angle of a facet with larger slope as the possible range. Second, a new triangular facet containing the possible range should be formed (Figures 2b and 2d). To ensure this triangular facet small enough, it is formed by the central cell (P_0), one downslope cell within the possible range, and a newly selected cell. The newly selected cell should be neighboring to the central cell and surrounding the possible range. So two neighboring cells of the central cell are alternative, that is, P_3 and P_8 in Figure 2. As the angle between the direction of any alternative cell center ($\overrightarrow{P_0P_3}$ or $\overrightarrow{P_0P_8}$) and its closest edge of the possible range ($\overrightarrow{P_0P_2}$ or $\overrightarrow{P_0P_1}$) is 45° , here we just need to confirm which direction ($\overrightarrow{P_0P_2}$ or $\overrightarrow{P_0P_1}$) in the two edges is closest to the TDD. For this task, the convergence of terrains is considered because the potential channel may be located at more convergent positions. The convergence of terrains can be described in plan curvature or

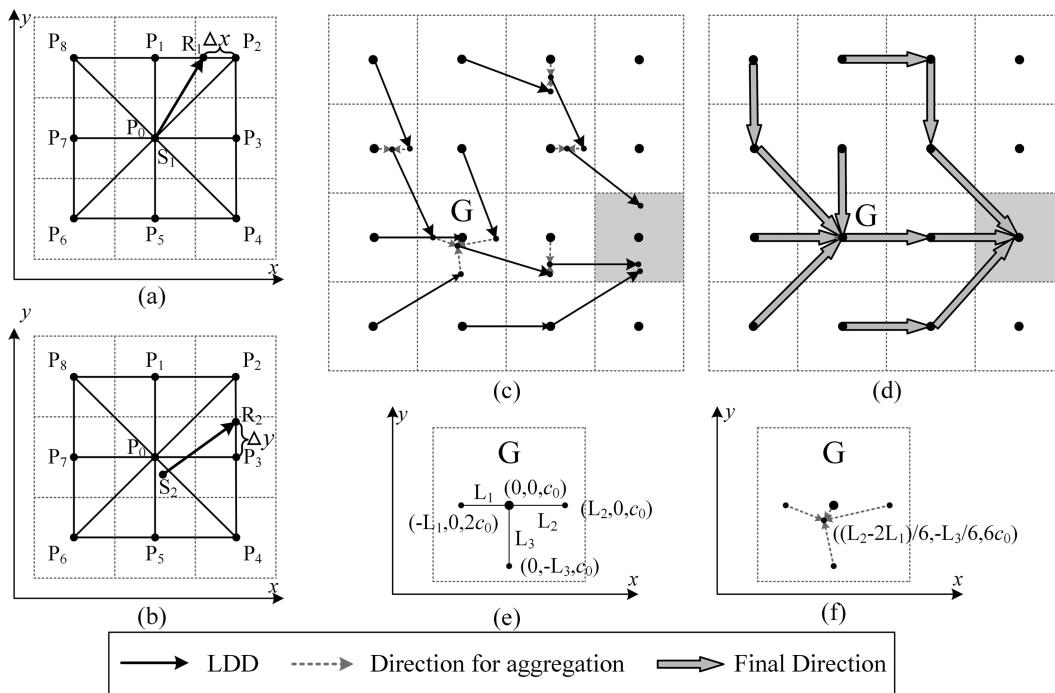


Figure 3. The aggregated flow package (S_1 or S_2) out of (a) the cell center (P_0) or (b) another position (S_2) moves along the LDD ($\overrightarrow{S_1R_1}$ or $\overrightarrow{S_2R_2}$) and reaches the border of the 3×3 window and then transfers the deviation (Δx , Δy) to the downstream cells. (c) The inflows for a downstream cell are aggregated to a new position as a package which continues to move along the LDD. (d) Then the complex paths are simplified into the eight-direction format. The cell G owning four inflow packages is enlarged in (e). L_1 , L_2 , and L_3 are distances between the cell center and three flow packages from upstream. Another inflow package reaches the cell center. Setting the area of one cell as c_0 , deviation (Δx_i , Δy_i), and the upstream drainage areas (c_i) are united into (Δx_i , Δy_i , c_i) and labeled. (f) According to function 1, the flow over G are aggregated to point ($(L_2-2L_1)/6$, $-L_3/6$), with an upstream drainage areas of $6c_0$.

tangential curvature, and the latter is more appropriate for defining the flow divergence/convergence (Krcho, 1991; Mitášová & Hofierka, 1993). So as tangential curvatures increase moving toward the channel centerline (Hooshyar et al., 2016), the edge direction ($\overrightarrow{P_0P_1}$ or $\overrightarrow{P_0P_2}$) of which the downslope cell (P_1 or P_2) is with higher tangential curvature tends to be closer to the TDD. Here the Mitášová and Hofierka (1993) method is applied for calculating tangential curvatures. To calculate the tangential curvature for a cell (P_1 or P_2), another 3×3 window centered at this cell (P_1 or P_2) is needed. Thus, the data in a 5×5 window are required to determine the final $ND\infty$ direction for a cell. After tangential curvatures obtained, the edge direction is used to determine the newly selected cell. For instance, in Figure 2, tangential curvatures of P_1 and P_2 are denoted as κ_1 and κ_2 , respectively. If P_2 has a higher tangential curvature than P_1 , namely, $\kappa_1 < \kappa_2$, P_3 is selected (Figure 2b). Otherwise, if $\kappa_1 > \kappa_2$, P_8 is selected (Figure 2d). Then a new triangular facet is formed by the newly selected cells, with the central cell (P_0 in Figure 2) and the lower-curvature cell (P_1 in Figure 2b and P_2 in Figure 2d), and its aspect is estimated as the second direction ($\overrightarrow{P_0Q_2}$ in Figure 2b or $\overrightarrow{P_0Q_4}$ in Figure 2d). Finally, the $ND\infty$ direction is calculated as the average direction of the unmodified $D\infty$ direction and the second direction ($\overrightarrow{P_0Q_3}$ in Figure 2c or $\overrightarrow{P_0Q_5}$ in Figure 2e). If the $ND\infty$ direction is not in the possible range, the same procedure used in $D\infty$ (Tarboton, 1997) is adopted to adjust it to the closest edge of the possible range. This procedure will only act at the partial V-shape terrain forms to force flow to drain along valley lines.

2.3. Flow Aggregation

As we take the $D\infty$ direction or the $ND\infty$ direction as the LDD, the flow package out of the cell center will miss the center of neighboring cell if the LDD mismatches the cardinal or diagonal directions (e.g., Figure 3a). Thus, we need to aggregate the flow packages on the downstream cell together into a new package and permit the new flow package to drain out along the LDD. The following part of this section

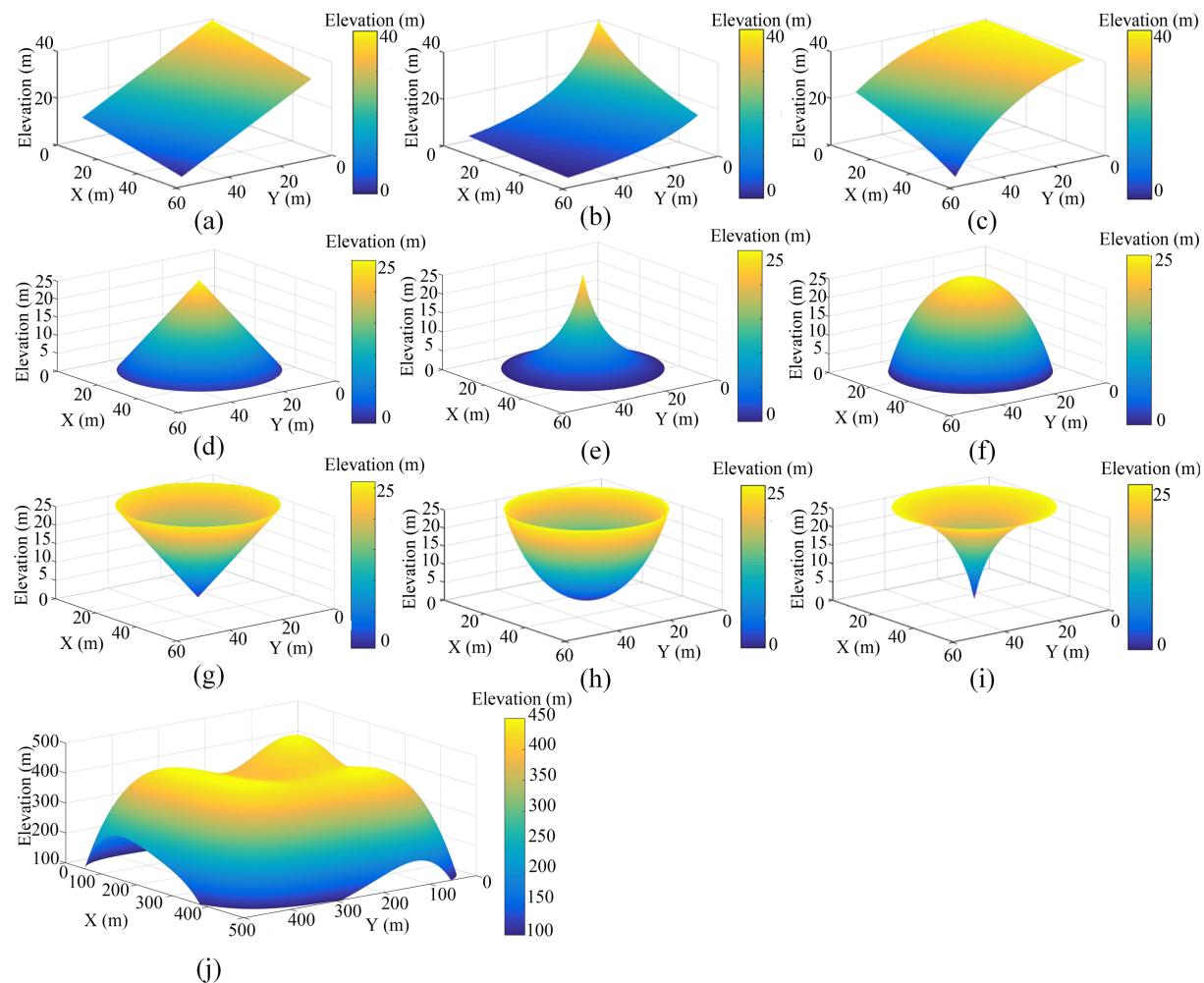


Figure 4. Ten mathematical terrains are used for the comparison between different algorithms: (a) planar plate; (b) concave plate; (c) convex plate; (d) planar cone; (e) concave cone; (f) convex cone; (g) planar inward cone; (h) concave inward cone; (i) convex inward cone; and (j) Himmelblau terrain.

Table 1
Functions of Mathematical Terrains Used for Evaluating Drainage Direction Algorithms

Mathematical terrains		Function	Limits
Plates	Planar	$z_1(x, y) = 40 - \frac{1}{5}x - \frac{3}{5}y$	$0 \leq x \leq 50$
	Concave	$z_2(x, y) = \frac{500}{50 - z_1(x, y)} - 10$	$0 \leq y \leq 50$
	Convex	$z_3(x, y) = \sqrt{6400 - 4800 \times \left(1 - \frac{z_1(x, y)}{40}\right)^2} - 40$	$0 \leq z_i \leq 40 \quad (i = 1, 2, 3)$
Cones	Planar	$z_4(x, y) = 25 - \sqrt{(x-25)^2 + (y-25)^2}$	$0 < x < 50$
	Concave	$z_5(x, y) = \frac{25z_4(x, y)}{100 - 3z_4(x, y)}$	$0 < y < 50$
	Convex	$z_6(x, y) = \sqrt{2500 - 1875 \times \left(1 - \frac{z_4(x, y)}{25}\right)^2} - 25$	$0 < z_i \leq 25 \quad (i = 4, 5, 6)$
Inward cones	Planar	$z_7(x, y) = 25 - z_4(x, y)$	
	Concave	$z_8(x, y) = 25 - z_6(x, y)$	

Table 1
(continued)

Mathematical terrains	Function	Limits
Himmelblau Convex	$z_9(x,y) = 25 - z_5(x,y)$ $z_{10} = 450 - 0.75 \left\{ \left[\left(\frac{x}{50} - 5 \right)^2 + \left(\frac{y}{50} - 5 \right)^2 - 4 \right]^2 + \left[\left(\frac{y}{50} - 5 \right)^2 + \left(\frac{x}{50} - 5 \right) - 7 \right]^2 \right\}$	$0 < x < 50$ $0 < y < 50$ $0 \leq z_i < 25 \quad (i = 7, 8, 9)$ $0 \leq x \leq 500$ $0 \leq y \leq 500$ $100 \leq z_{10} \leq 450$

Note. Unit is in meters.

illustrates the process to determine the aggregated package location, the final direction, and the deviations transferred to the downstream cell.

Whether the flow package of a cell is located at the cell center (e.g., S_1/P_0 in Figure 3a) or other locations (e.g., S_2 in Figure 3b), it is allowed to move along the LDD within the 3×3 window (e.g., the square window linked by P_2, P_4, P_6 , and P_8 in Figures 3a and 3b). Once the package reaches the window border at an intersection (e.g., R_1 in Figure 3a or R_2 in Figure 3b), the cell owning the intersection (e.g., P_2 in Figure 3a or P_3 in Figure 3b) is determined as the downstream cell. The 2-D deviation of the intersection from the downstream cell center is recorded as $(\Delta x, \Delta y)$. As shown in Figure 3, only Δx (Figure 3a) or Δy (Figure 3b) within a cell requires calculation, while another one is equal to zero. These deviations can provide assistance to determine the aggregated flow position for a downstream cell.

It is worth noting that every cell has its one unit of native flow package (viz., flow originating from the cell itself) located at its center, and this flow package has no deviation with the cell center. In addition, there is only one flow package for a source cell as it has no upstream cell. Hence, source cells do not need flow aggregation, and their flow packages can be released from their centers. For a downstream cell, there are several packages from upstream at different locations as well as its own one unit of flow package at the center. These packages need to be aggregated to a point, whose location is determined by the locations and upstream drainage areas of all the inflow packages, to form a new flow package. The relative coordinates (X, Y) for the point from the cell center are derived as

$$\begin{cases} X = \sum_{i=1}^n c_i \Delta x_i / \left(\sum_{i=1}^n c_i + c_0 \right) \\ Y = \sum_{i=1}^n c_i \Delta y_i / \left(\sum_{i=1}^n c_i + c_0 \right) \end{cases}, \quad (1)$$

where n is the number of upstream neighboring cells, Δx_i and Δy_i are the deviations with respect to the x and y direction inherited from the i th upstream neighboring cell, c_i is the upstream drainage area of the i th upstream cell, and c_0 is the area of the study cell. Then the aggregated package moves out of the new position along the LDD of the cell and repeats the process above.

Once all cells are calculated, the aggregated flow paths based on our theory are shown in Figure 3c. We just use these lines to show the process of the aggregation, and only the downstream cell, as well as the deviation between the arriving position of the flow and the downstream cell center, should be stored. Finally, due to every cell having only one downstream cell, the final direction for this cell is pointing from its center to the

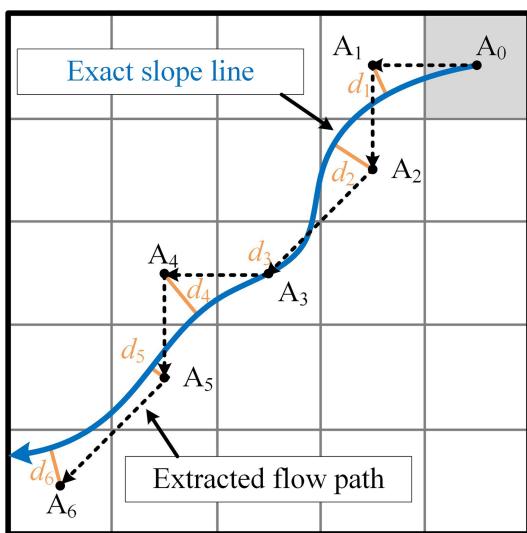


Figure 5. The extracted flow path souring from A_0 passes through six downstream cell centers (A_1 – A_6). The yellow lines represent the shortest distances between the downstream cell centers and the exact slope line of A_0 . The lengths of these yellow lines (d_1 – d_6) are treated as point lateral deviations of downstream cells (A_1 – A_6). Finally, the average path lateral deviation (APLD) of A_0 is calculated as the average of these point lateral deviations (d_1 – d_6).

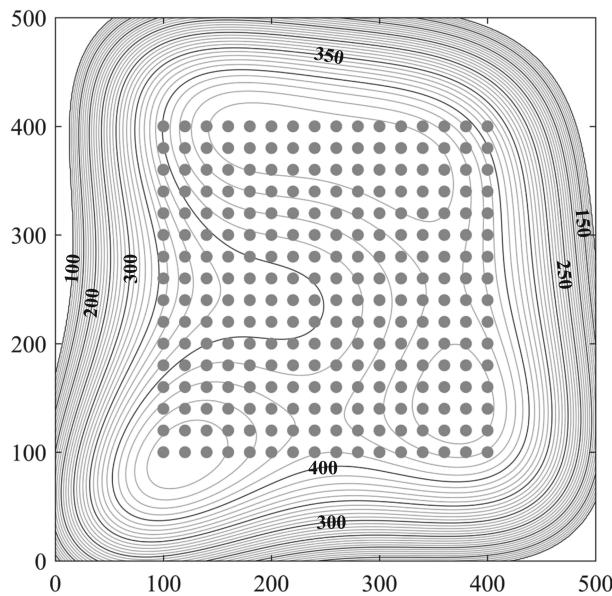


Figure 6. Contour map of the Himmelblau terrain. The dark gray contour lines with 50 m intervals are labeled with elevation values. A total of 256 evenly distributed cells (gray points) are selected for assessment. The 2-D coordinates (x, y) of these cells are defined as $x = 20p+100$ and $y = 20q+100$, where the integer p and q are ranging from 0 to 15.

discretized into $1 \text{ m} \times 1 \text{ m}$ DEMs. All the plates have the same ratio of 1:3 between the lateral and longitudinal gradient anywhere according to their definitions. So exact slope lines derived from the definition function for all cells are straight lines with $\arctan(1/3)$ rad (i.e., 18.4°) inclined from the longitudinal direction when projected to a plane. Moreover, exact slope lines derived from the function for cells in any inward cone are straight lines pointing to the tip and are straight lines pointing back to the tip for cells in any cone (Paik, 2008).

In order to provide more reliable verification results, assessment using a complex terrain is necessary. Early researches applied nondispersive algorithms to real-world DEMs and attempted to find visible errors from extracted flow paths (e.g., Orlandini et al., 2003; Paik, 2008). However, due to the complex microtopographical features, quantitative assessment is always lacked because exact slope lines over real-world DEMs are hard to be determined. Subsequently, Orlandini et al. (2014) provided an abstract terrain created with a Himmelblau's function for flow path assessment (Figure 4j). This Himmelblau terrain is almost as complex as real-world terrains and was also adopted by others, for example, Shin and Paik (2017). Exact slope lines on this terrain can be obtained from its differentiable function with finite difference methods. Here we adopt the Himmelblau terrain (see in Table 1 for its terrain function) with seven different resolutions (0.1, 0.5, 1, 2, 5, 10, and 20 m). In the present study, the ode23tb solver contained in Matlab™ used also by Orlandini et al. (2014) is employed for numerical integrations to provide exact slope lines.

3.2. Assessment Criterion

Performances of different nondispersive algorithms, including D8, D8-LTD, iGD8, FAD8, and iFAD8 are compared on the abstract terrains. These algorithms are assessed to reproduce two hydrological attributes derived from terrain functions, that is, the exact slope line and the theoretical SCA.

3.2.1. Slope Line Assessment

Deviations between extracted flow paths and exact slope lines, parallelism of paths for the plates, and symmetry of paths for the cones are used for visual assessment. Furthermore, a numerical criterion named global lateral deviation is also adopted for quantitative assessment.

As shown in Figure 5, for an extracted flow path beginning from a given cell (e.g., A_0), some downstream cells' centers are not located on the exact slope line. The point lateral deviation can be calculated for every

downstream cell center for a simple storage format; that is, only the cardinal or diagonal direction is used (Figure 3d). Hence, our algorithm (either FAD8 or iFAD8) is an eight-direction method, which considers more global information than D8-LTD and iGD8. Here a detailed example to calculate the relative coordinates of the aggregated position for the point G in Figure 3c is provided and illustrated in Figures 3e and 3f. Furthermore, if a cell has one upstream flow path with a huge drainage area, the relative deviation derived by function 1 may be very close to the deviation from this main path. In this case, FAD8 based on D_∞ may behave in a similar manner as D8-LTD. So FAD8 is an algorithm adopting a dynamic dampening factor to enhance D8-LTD. Drainage networks by FAD8 and D8-LTD are different mainly in the upstream regions.

3. Performance on Abstract Terrains

3.1. Abstract Terrains

Basic landforms can be divided into nine types according to the combinations of parallel, divergent or convergent platforms, and planar, concave, or convex profiles (Liu et al., 2013). Hence, nine simple mathematical abstract terrains are considered here, including three parallel plates, three divergent cones, and three convergent inward cones with different profiles. Each terrain represents one type of basic landform. Three-dimensional visualizations of these terrains are shown in Figures 4a–4i, and their functions are listed in Table 1. In order to generate grid DEMs for algorithm evaluation, continuous terrains are discretized

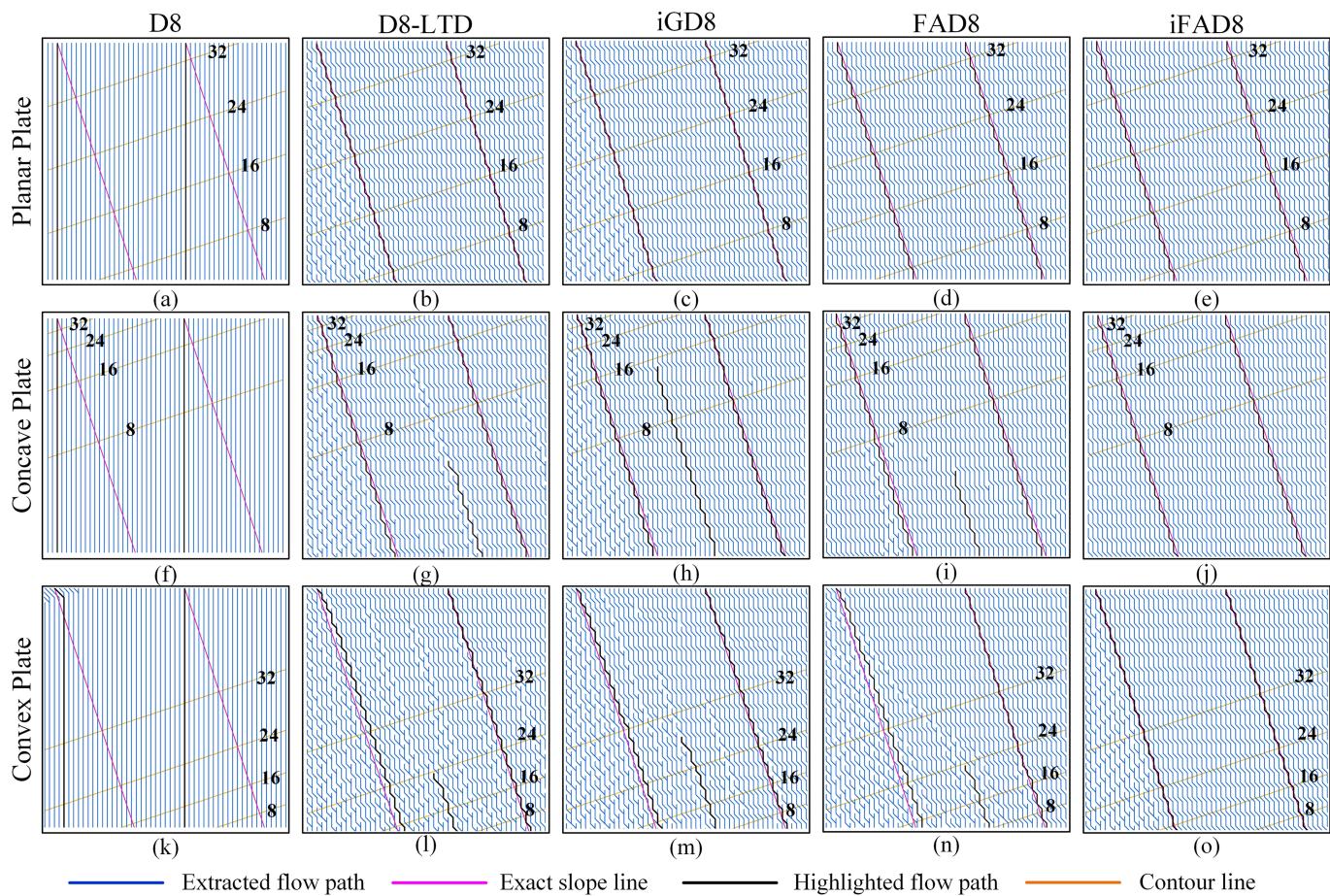


Figure 7. Extracted flow paths for different plates with different methods. For visual evaluation of algorithm performance, exact slope lines and extracted flow paths sourcing from two upstream cells are highlighted in purple and black for each plate. Some short flow paths sourcing from locations where upstream extracted paths diverge are also highlighted in black. The contour lines highlighted in yellow are labeled with intervals of 8 m.

downstream cell as the shortest distance between the cell center and the exact slope line (Paik, 2008), that is, the lengths of the yellow lines in Figure 5. Paik (2008) used the path lateral deviation to represent the deviation between the extracted flow path and the exact slope line of a specific source cell. This deviation is calculated as the sum of the point lateral deviations of all downstream cells along the extracted flow path. Obviously, a larger path lateral deviation can always be observed with a longer extracted flow path. So we adopt the average path lateral deviation (*APLD*) here. It computes the average instead of the sum of the point lateral deviations of all downstream cells along the extracted flow path for a specific source cell. The average path lateral deviation ($APLD_j$) of the j th source cell in a given terrain is denoted as

$$APLD_j = \frac{1}{m_j} \sum_{k=1}^{m_j} d_{k,j}, \quad (2)$$

where $d_{k,j}$ is the point lateral deviations of the k th downstream cell along the extracted flow path of the j th source cell and m_j is the number of downstream cells in the extracted flow path. Thus, for the whole given terrain, the global lateral deviation (*GLD*) is denoted as

$$GLD = \frac{1}{l} \sum_{j=1}^l APLD_j, \quad (3)$$

where l denotes the number of source cells in the selected terrain.

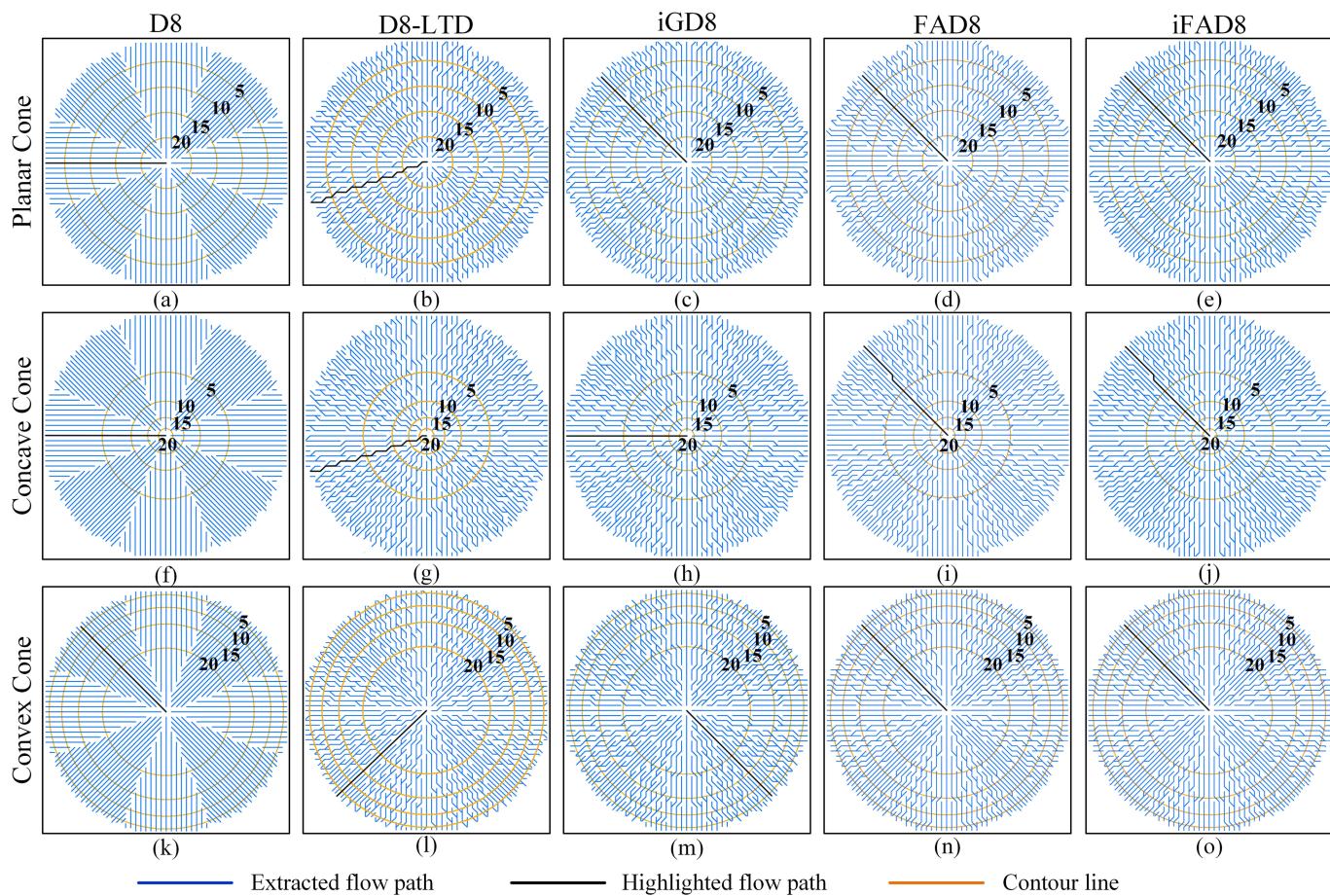


Figure 8. Extracted flow paths for different cones with different methods. The contour lines highlighted in yellow are labeled with intervals of 5 m. The flow paths out from the tip are highlighted in black.

Limited by the complexity of the Himmelblau terrain and the enormous amount of cells, it is difficult to automatically calculate all the exact slope lines for all cells in this terrain at once by Matlab programming. But if we select enough cells for assessment under a fair principle, a convincing result can still be obtained. Previous research adopted generally a small number of cells (10 or even less) with typical flow paths for assessments (Orlandini et al., 2014; Shin & Paik, 2017). In this study, the authors calculate the *APLDs* for more evenly distributed cells (256 cells shown in Figure 6) and take the average of these deviations as the *GLD* for the Himmelblau terrain.

3.2.2. Specific Catchment Area Assessment

The mean absolute error (*MAE*) between SCAs derived from the drainage direction algorithms (abbreviated as the numerical SCA) and the theoretical “true value” of SCA (abbreviated as the theoretical SCA) derived from terrain functions is adopted for further assessments. The numerical SCAs are computed following the Costa-Cabral and Burges (1994) method. The theoretical SCA is determined according to the terrain functions for any point over various terrains based on the Gallant and Hutchinson (2011) method. In particular, for complex terrains like the Himmelblau terrain, Gallant and Hutchinson (2011) introduced a technique to obtain the theoretical SCA through numerical integrations along the exact slope line. The *MAE* is adopted to express the bias between the numerical and theoretical SCAs. The *MAE* for a terrain owning n cells is computed as

$$MAE = \frac{1}{n} \sum_{i=1}^n \frac{|NSCA_i - TSCA_i|}{TSCA_i}, \quad (4)$$

where $NSCA_i$ and $TSCA_i$ denote the numerical SCA and theoretical SCA for the i th cell in the selected terrain, respectively.

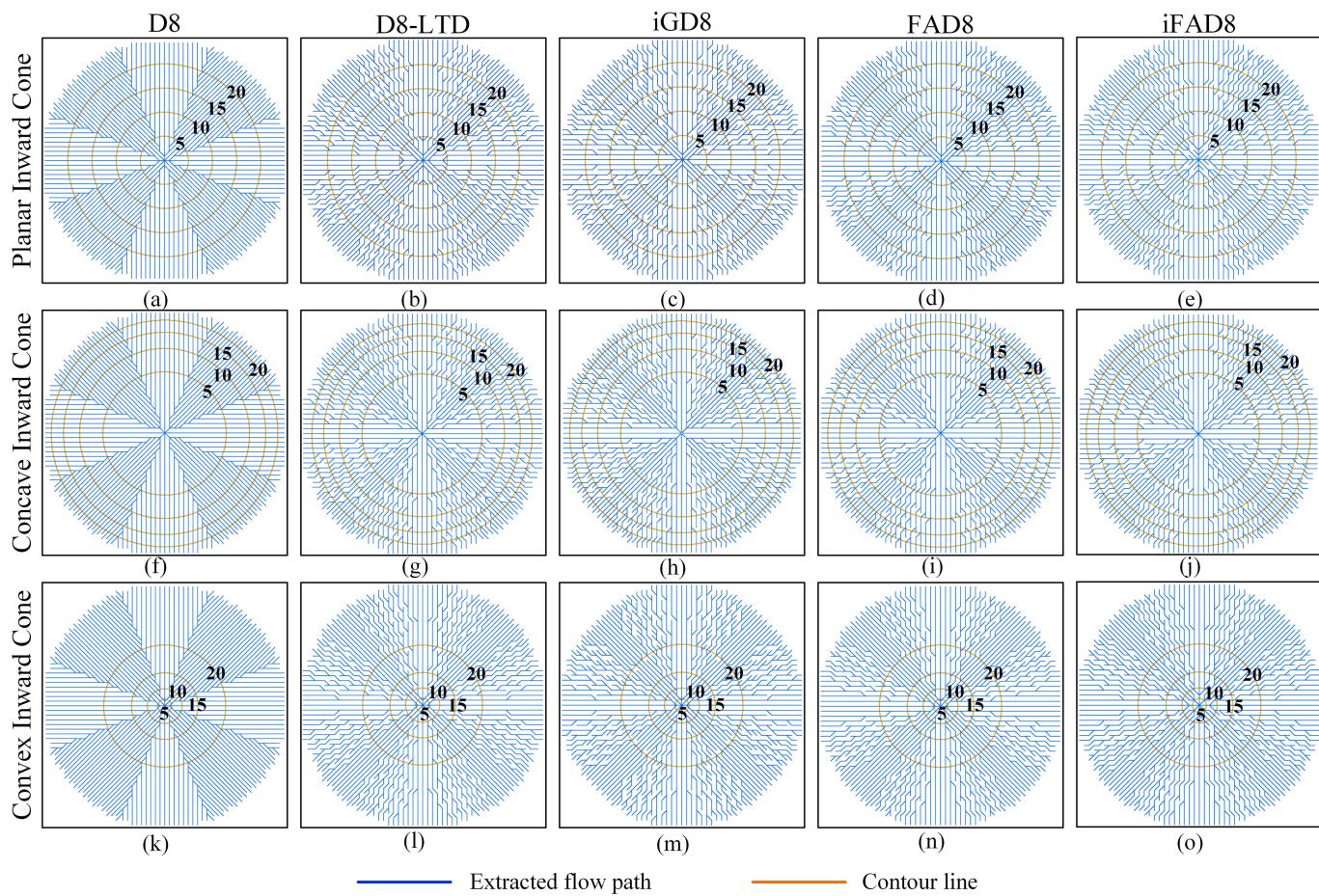


Figure 9. Extracted flow paths for different inward cones with different methods. The contour lines highlighted in yellow are labeled with intervals of 5 m.

3.3. Results and Discussions

3.3.1. Slope Line Assessment

The assessment results are derived in two categories, namely, the visual assessment and quantitative assessment. For visual assessment, the extracted flow paths on the plates, the cones, and the inward cones are shown in Figures 7, 8, and 9, respectively. In these figures, the extracted flow paths are presented in blue lines, and contour lines are shown in yellow lines. According to Shin and Paik (2017), differences between the extracted flow paths and the exact slope lines on the plates may be obvious, as well as the symmetry of the extracted flow paths on the cones. Hence, two upstream cells on each plate are chosen discretionarily for comparison (Figure 7). Their exact slope lines are presented in purple lines, with the extracted paths highlighted in black. In addition, a few extracted flow paths (short paths; e.g., see in Figure 7g) beginning from locations where upstream extracted paths diverge are also highlighted. Furthermore, the extracted flow paths out of the cone tips are highlighted in black to show the symmetry in Figure 8.

The 1 m resolution Himmelblau terrain is chosen for visual assessment. The extracted flow paths on the Himmelblau terrain are too complex for visualization. Thus, only 12 cells with the largest mean values of $APLD$ by the four algorithms, for example, D8-LTD, iGD8, FAD8, and iFAD8, are selected. Here D8 is not taken into consideration because it is an admittedly inaccurate algorithm with largest deviations (e.g., Orlandini et al., 2003, 2014; Shin & Paik, 2017). Then the extracted flow paths, the exact slope lines, and the contour lines are presented severally in blue, yellow, and gray lines for clear visualization in Figure 10.

Generally, according to the visual assessments, iFAD8 extracts reasonable flow paths on all 10 abstracted terrains. The GLDs for nine simple terrains are listed in Table 2, and the GLDs for the Himmelblau terrain with

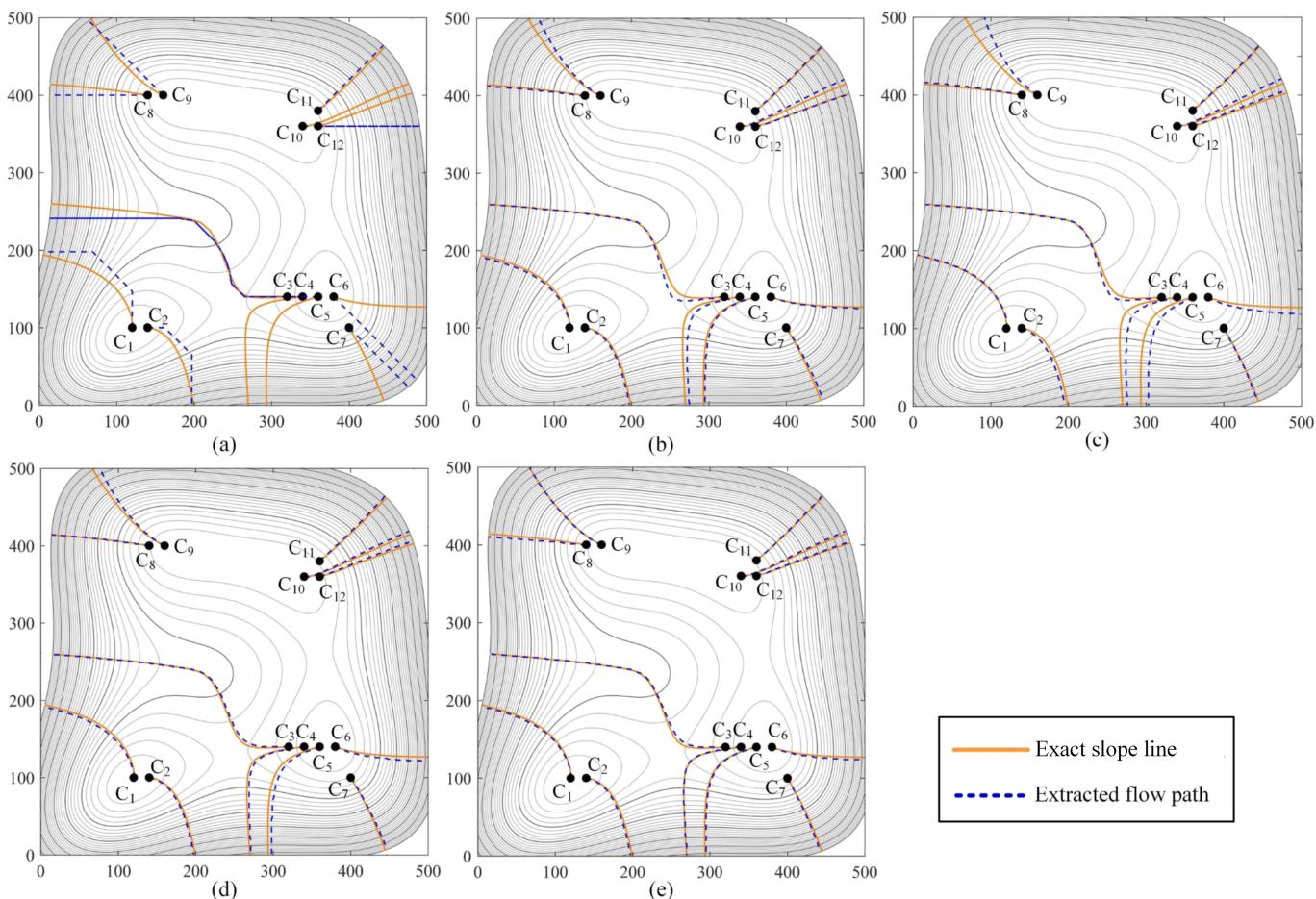


Figure 10. Exact slope lines and extracted flow paths by (a) D8, (b) D8-LTD, (c) iGD8, (d) FAD8, and (e) iFAD8 for 12 selected cells in the 1 m-resolution Himmelblau terrain. The coordinates of these cells are C1 (120, 100), C2 (140, 100), C3 (320, 140), C4 (340, 140), C5 (360, 140), C6 (380, 140), C7 (400, 100), C8 (140, 400), C9 (160, 400), C10 (340, 360), C11 (360, 380), and C12 (360, 360).

different resolutions are listed in Table 3. The results show that iFAD8 totally produces the least systematic bias with the lowest values of the GLDs among all algorithms on the 10 terrains. Hence, iFAD8 is proved to be an effective nondispersive drainage direction algorithm to reproduce exact slope lines by both visual assessment and quantitative assessment.

Flow paths on different plates extracted by different algorithms are shown in Figure 7. Due to the local search rule (Paik, 2008), D8 loses the lateral propagation and extracts straight flow paths along the longitudinal direction on the planar plate (Figure 7a). The extracted flow paths of iFAD8 are parallel to each other and extend along the exact slope lines (Figure 7e). So iFAD8 has a better performance than D8-LTD, iGD8 on this planar plate (Figures 7b and 7c). The D_{∞} and ND_{∞} directions are the same on any planar plate because all triangular facets constructed from points in a planar plate have the same aspect. Hence, FAD8 (Figure 7d) provides the same flow paths as iFAD8 (Figure 7e). Furthermore, iFAD8 also provides a more appropriate flow path than other algorithms on the concave or convex plate. For instance, except the results of iFAD8 (Figures 7j and 7o), obvious false flow paths covering at least 20% of the total area can be observed for all other results. D8 provides apparently wrong flow paths on these plates again (Figures 7f and 7k). Meanwhile, the performances of D8-LTD, iGD8, and FAD8 are affected by the profile concavity. As shown in these figures, the highlighted flow paths in black extracted by D8-LTD, iGD8, and FAD8 are either understeered on the concave plate (Figures 7g–7i) or oversteered on the convex plate (Figures 7l–7n). The phenomenon has caused some flow paths to diverge from or to merge into other flow paths. Thus, there are some false independent flow paths that do not source from the border of the plates in Figures 7g–7i and 7l–7n. These false paths on concave or convex terrains lay in the bias of LDDs provided by D_{∞} that is adopted

Table 2
Global Lateral Deviations (GLDs) Calculated for Different Algorithms on Different Simple Abstract Terrains

Abstract terrain		D8	D8-LTD	iGD8	FAD8	iFAD8
Plates	Planar	40.32	2.88	2.87	2.71	2.71
	Concave	40.32	3.18	3.11	3.01	2.71
	Convex	40.27	3.65	3.70	3.51	2.87
Cones	Planar	7.81	4.15	3.66	3.07	2.59
	Concave	7.95	4.41	3.85	3.61	3.47
	Convex	7.94	3.79	3.61	3.56	3.49
Inward cones	Planar	6.98	2.67	5.18	2.97	2.64
	Concave	6.94	3.04	5.10	3.21	3.00
	Convex	6.90	2.96	5.06	3.09	2.82

Note. Unit is in 10^{-1} m.

by D8-LTD, iGD8, and FAD8. The better performance of iFAD8 than other methods, especially FAD8, shows that ND_{∞} can provide better LDDs for cells on concave or convex terrains than that of D_{∞} . In addition, like other algorithms, iFAD8 also produces some nonparallel flow paths sourcing from the left border on the convex plate (Figure 7o). This is probably due to that ND_{∞} only provides partly improvements to the LDDs but cannot address the issue perfectly on terrains with concavity.

The flow paths extracted by different methods on the cones are shown in Figure 8. For the cones, the theoretical flow paths should be radially symmetric straight lines, and the ideal extracted flow paths should be radially zigzag lines (Paik, 2008). D8 exhibits its disadvantage clearly to manufacture too many faulty parallel flow paths (Figures 8a, 8f, and 8k). Most extracted paths of D8 are straight and deviate from the theoretical directions. D8-LTD, iGD8, FAD8, and iFAD8 provide radially diverging zigzag

flow paths, which are rational on these divergent terrains. On all the cones, we find that the global asymmetry exists for flow paths extracted by D8-LTD (Figures 8b, 8g, and 8l), and this phenomenon also appears in the results of FAD8 and iFAD8 when they are applied to the concave cone (Figures 8i and 8j). Orlandini and Moretti (2009b) explained that the choice for the D_{∞} direction will result in the asymmetry when multiple triangular facets by D_{∞} have the same maximum slope. It is also an applicable explanation for the asymmetrical result of FAD8 on the convex cone because FAD8 relies on the D_{∞} directions as well. Employing the D_{∞} direction to determine the possible range, the ND_{∞} direction is also under the influence of the D_{∞} direction. In addition, flow paths causing the asymmetry are generally along the radial lines and can act as an appropriate trajectory for the drainage process. Hence, it seems that all the algorithms except D8 have reasonable performances over the cones according to visual assessment.

The extracted flow paths of different algorithms on the inward cones are shown in Figure 9. It seems that D8-LTD, iGD8, FAD8, and iFAD8 have similar results, which are much better than that of D8 on these terrains. Generally, good results of different nondispersing algorithms on inward cones verify the widely accepted view that is, nondispersing algorithms are appropriate for runoff concentration simulation on convergent terrains (e.g., Xiong et al., 2014; Zhou et al., 2011).

Extracted flow paths for 12 selected cells on the 1 m-resolution Himmelblau terrain are exhibited in Figure 10. As shown in Figure 10a, D8 produces many unreasonable straight flow paths. For instance, D8 cannot drain the flow into the valley line at the weakly convergent valley according to the extracted flow paths for C_3 , and even worse, it provides several false flow paths (e.g., for C_4 and C_5) that do not exist at all. Several flow paths extracted by D8-LTD, iGD8, and FAD8 diverge from the exact slope lines seriously (Figures 10b–10d). Again, iFAD8 reproduces the exact slope lines more successfully.

For quantitative assessment, the values of GLDs are calculated for every abstract terrain with all the algorithms. The results in Table 2 show that iFAD8 has the best performance on all nine simple terrains. The GLDs on nine simple terrains of iFAD8 are 84.1%, 14.4%, 27.2%, and 8.5% lower than those of D8, D8-LTD, iGD8, and FAD8 on average, respectively. Furthermore, as the resolution goes to zero, all algorithms except D8 can provide flow paths roughly consistent with the exact slope lines on the Himmelblau terrain (Table 3).

Particularly, iFAD8 is always the best algorithm with the lowest GLDs to reproduce the exact slope lines on the Himmelblau terrain.

Through the abstract terrain applications, we find that generally FAD8 has better performances than other D_{∞} -depended algorithm (D8-LTD and iGD8) on most terrains. For instance, the GLDs on nine simple terrains of FAD8 are 6.5% and 20.5% lower than those of D8-LTD and iGD8 on average. Moreover, FAD8 has a lower GLD than D8-LTD or iGD8 on the Himmelblau terrain for most selected resolutions. Hence, it is obvious that the flow aggregation method adopted by iFAD8 and FAD8 is a better holistic approach to reproduce exact slope lines. In addition, the general better performance

Table 3
Global Lateral Deviations (GLDs) Calculated for Different Algorithms on the Himmelblau Terrain With Different Resolutions

Algorithm	Resolution (m)						
	0.1	0.5	1	2	5	10	20
D8	94.98	96.01	96.62	95.18	97.48	95.75	102.98
D8-LTD	3.39	5.14	7.63	11.60	25.39	38.37	70.89
iGD8	3.40	5.38	8.73	14.03	32.41	47.30	70.50
FAD8	3.45	5.08	7.58	11.47	22.96	42.27	72.98
iFAD8	3.34	4.95	6.58	11.04	23.54	37.13	59.46

Note. Unit is in 10^{-1} m.

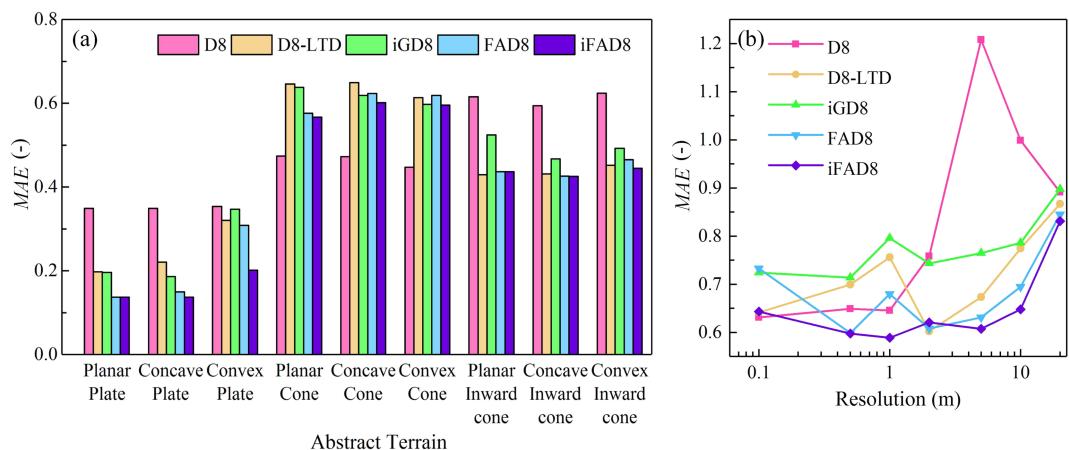


Figure 11. Mean absolute errors (MAEs) between the numerical SCAs (specific catchment area) and the theoretical SCAs derived from terrain functions. (a) Nine 1 m-resolution simple abstract terrains are used, and (b) the Himmelblau terrain with seven different resolutions is considered.

of iFAD8 than FAD8 shows that the $ND\infty$ method based on the technique of flexible triangular facet construction is exactly an improvement for $D\infty$ to provide the LDDs for the nondispersing drainage direction method.

3.3.2. Specific Catchment Area Assessment

The MAEs between the numerical and theoretical SCAs on different abstract terrains are shown in Figure 11. In Figure 11a, iFAD8 generally outperforms the other algorithm on the plates and the cones and nearly has the same best performance as D8-LTD on the inward cones. FAD8 behaves as well as D8-LTD on the cones and the inward cones and is better than D8-LTD on the plates. Similar to the results of existing research (e.g., Zhou et al., 2011), the numerical SCA derived based on D8 algorithm may be much closer to the theoretical value than any other algorithms on the divergent terrains (e.g., the cones). However, on any plate or inward cone, the performance of D8 is the worst among all the algorithms. As the DEM resolution goes to zero, the MAEs of different algorithms on the Himmelblau terrain will decrease in a fluctuating way (Figure 11b). The reason is that the upstream area by nondispersing algorithms for a point especially in relatively finer resolution scenarios (<5 m) is fluctuating in a divergent region (Han et al., 2018), so the cell will be assigned a variable SCA. Nevertheless, in different resolution scenarios, such cells are randomly distributed. Thus, within the 256 selected cells in this study, the MAEs between numerical and theoretical SCAs will also result in a fluctuation (Figure 11b).

At last, it can be figured out that iFAD8 is generally a steady algorithm to reproduce theoretical SCA and can provide the lowest envelope line for all resolutions considered. FAD8 is proved to be second to iFAD8 on the Himmelblau terrain as shown in Figure 11b. Through the assessment based on SCA in Figure 11, both FAD8 and iFAD8 exhibit the advantages to the existing path-based algorithms (D8-LTD and iGD8).

4. Concluding Remarks

A new nondispersing algorithm for drainage direction extraction on grid DEMs is presented in this study. The new algorithm called iFAD8 determines the drainage directions from upstream cells to downstream cells with two techniques. First, a flexible triangular facet construction technique is employed to improve the classic $D\infty$ triangular facets and then obtain a new infinite direction named $ND\infty$. In this study, the $ND\infty$ direction is taken as the LDD for every cell. Second, in a given cell, all the flow packages are aggregated to the eight possible directions that results in zigzag-like flow path around the LDDs.

The iFAD8 algorithm is compared with other nondispersing algorithms, including D8, D8-LTD, iGD8, and FAD8, on 10 abstract terrains. Among them, FAD8 also proposed in this study adopts the flow aggregation technique while it takes the $D\infty$ directions as the LDDs. Here FAD8 is an enhancement to D8-LTD and mainly mitigates the boundaries effects. Assessments of the derived slope lines and SCAs for all the algorithms are provided. Both the results show that iFAD8 outperforms other nondispersing algorithms on all

abstract terrains. Moreover, FAD8 performs generally better than D8, D8-LTD, and iGD8. These results indicate that both the flexible triangular facet construction and flow aggregation techniques can improve the performance of the nondispersive algorithm. Furthermore, compared to existing algorithms, the complexity of iFAD8 will not cause unaffordable computation efficiency (see in Appendix A). Hence, the iFAD8 method is worthy of consideration when nondispersive flow paths are needed. The authors tend to recommend iFAD8 to high-resolution applications (i.e., <10 m); otherwise, the tangential curvature used for ND ∞ is sensitive and may be distorted on low-resolution terrains.

Appendix A: Comparison of Computational Efficiency

It is difficult to compare the computational efficiency directly between different nondispersive drainage direction algorithms because they are implemented by different programming languages. Therefore, D8 is chosen as a reference, which is simple enough to be implemented correctly by different programming languages. The CPU run times averaged over 10 times are measured for different algorithms with a DEM (500×632 cells) used by Shin and Paik (2017). Implemented by Java codes, D8, FAD8, and iFAD8 consume 0.048, 0.475, and 0.477 s, respectively, on a computer equipped with an Intel Xeon E5-1620 v4 CPU and 16 GB of memory. FAD8 and iFAD8 cost almost 10 times the run time of D8. Shin and Paik (2017) measured that the run times of D8, D8-LTD, and iGD8 were 0.08, 4.51, and 0.81 s with Fortran codes on another computer. Ten times the run time of D8 were also required by iGD8, while D8-LTD took more time. Although there are differences in programming languages and computer performances, the run times of FAD8 and iFAD8 are as acceptable as other existing algorithms such as D8-LTD and iGD8.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC) (Grants 41730750, 41771025, 41901019, 91647108, and 91747203), the UK-China Critical Zone Observatory (CZO) Programme (41517130071) funded by the NSFC, the Fundamental Research Funds for the Central Universities (2019B17714), and the China Postdoctoral Science Foundation (2019M651679). The authors would like to thank Stefano Orlandini (University of Modena and Reggio Emilia) and Sanghoon Shin (Michigan State University) for the Fortran codes of D8-LTD and iGD8. Associate editor and three anonymous reviewers are also acknowledged for constructive comments and suggestions. The Java codes that implement the FAD8 and iFAD8 algorithms proposed in this paper are available at (<https://github.com/hhuwfp/iFAD8>).

References

- Ariza-Villaverde, A. B., Jiménez-Hornero, F. J., & Gutiérrez de Ravé, E. (2013). Multifractal analysis applied to the study of the accuracy of dem-based stream derivation. *Geomorphology*, 197, 85–95. <https://doi.org/10.1016/j.geomorph.2013.04.040>
- Bonetti, S., Bragg, A. D., & Porporato, A. (2018). On the theory of drainage area for regular and non-regular points. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474, 20170693. <https://doi.org/10.1098/rspa.2017.0693>
- Camporese, M., Paniconi, C., Putti, M., & Orlandini, S. (2010). Surface-subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data. *Water Resources Research*, 46, W02512. <https://doi.org/10.1029/2008WR007536>
- Ciarapica, L., & Todini, E. (2002). Topkapi: A model for the representation of the rainfall-runoff process at different scales. *Hydrological Processes*, 16(2), 207–229. <https://doi.org/10.1002/hyp.342.abs>
- Costa-Cabral, M. C., & Burges, S. J. (1994). Digital elevation model networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas. *Water Resources Research*, 30(6), 1681–1692. <https://doi.org/10.1029/93WR03512>
- Dietrich, W. E., Reiss, R., Hsu, M.-L., & Montgomery, D. R. (1995). A process-based model for colluvial soil depth and shallow landsliding using digital elevation data. *Hydrological Processes*, 9, 383–400. <https://doi.org/10.1002/hyp.3360090311>
- Fairfield, J., & Leymarie, P. (1991). Drainage networks from grid digital elevation models. *Water Resources Research*, 27(5), 709–717. <https://doi.org/10.1029/90WR02658>
- Freeman, T. G. (1991). Calculating catchment area with divergent flow based on a regular grid. *Computers & Geosciences*, 17(3), 413–422. [https://doi.org/10.1016/0098-3008\(91\)90048-I](https://doi.org/10.1016/0098-3008(91)90048-I)
- Gallant, J. C., & Hutchinson, M. F. (2011). A differential equation for specific catchment area. *Water Resources Research*, 47, W05535. <https://doi.org/10.1029/2009WR008540>
- Han, X., Liu, J., Mitra, S., Li, X., Srivastava, P., Guzman, S. M., & Chen, X. (2018). Selection of optimal scales for soil depth prediction on headwater hillslopes: A modeling approach. *Catena*, 163, 257–275. <https://doi.org/10.1016/j.catena.2017.12.026>
- Hooshyar, M., Wang, D., Kim, S., Medeiros, S. C., & Hagen, S. C. (2016). Valley and channel networks extraction based on local topo graphic curvature and k-means clustering of contours. *Water Resources Research*, 52, 8081–8102. <https://doi.org/10.1002/2015WR018479>
- Kienzle, S. (2004). The effect of dem raster resolution on first order, second order and compound terrain derivatives. *Transactions in GIS*, 8(1), 83–111. <https://doi.org/10.1111/j.1467-9671.2004.00169.x>
- Krcho, J. (1991). Georelief as a subsystem of landscape and the influence of morphometric parameters of georelief on spatial differentiation of landscape-ecological processes. *Ecology (CSFR)*, 10(2), 115–157.
- Lea, N. L. (1992). An aspect driven kinematic routing algorithm. In A. J. Parsons & A. D. Abrahams (Eds.), *Overland Flow: Hydraulics and Erosion Mechanics* (pp. 393–407). New York: Chapman and Hall.
- Liu, J., Chen, X., Lin, H., Liu, H., & Song, H. (2013). A simple geomorphic-based analytical model for predicting the spatial distribution of soil thickness in headwater hillslopes and catchments. *Water Resources Research*, 49, 7733–7746. <https://doi.org/10.1002/2013WR013834>
- Liu, J., Chen, X., Zhang, X., & Hoagland, K. D. (2012). Grid digital elevation model based algorithms for determination of hillslope width functions through flow distance transforms. *Water Resources Research*, 48, W04532. <https://doi.org/10.1029/2011WR011395>
- Maxwell, J. C. (1870). On hills and dales. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 40(269), 421–427.
- Mitášová, H., & Hofierka, J. (1993). Interpolation by regularized spline with tension: II. Application to terrain modeling and surface geometry analysis. *Mathematical Geology*, 25(6), 657–669. <https://doi.org/10.1007/BF00893172>
- O'Callaghan, J. F., & Mark, D. M. (1984). The extraction of drainage networks from digital elevation data. *Computer vision, graphics, and image processing*, 28(3), 323–344. [https://doi.org/10.1016/S0734-189X\(84\)80011-0](https://doi.org/10.1016/S0734-189X(84)80011-0)

- Orlandini, S., & Moretti, G. (2009a). On the determination of surface flow paths from gridded elevation data. *Water Resources Research*, 45, W03417. <https://doi.org/10.1029/2008WR007099>
- Orlandini, S., & Moretti, G. (2009b). Comment on “global search algorithm for nondispersive flow path extraction” by Kyungrock Paik. *Journal of Geophysical Research*, 114, F04004. <https://doi.org/10.1029/2008JF001193>
- Orlandini, S., Moretti, G., Corticelli, M. A., Santangelo, P. E., Capra, A., Rivola, R., & Albertson, J. D. (2012). Evaluation of flow direction methods against field observations of overland flow dispersion. *Water Resources Research*, 48, W10523. <https://doi.org/10.1029/2012WR012067>
- Orlandini, S., Moretti, G., Franchini, M., Aldighieri, B., & Testa, B. (2003). Path-based methods for the determination of nondispersive drainage directions in grid-based digital elevation models. *Water Resources Research*, 39(6), 1144. <https://doi.org/10.1029/2002WR001639>
- Orlandini, S., Moretti, G., & Gavioli, A. (2014). Analytical basis for determining slope lines in grid digital elevation models. *Water Resources Research*, 50, 526–539. <https://doi.org/10.1002/2013WR014606>
- Paik, K. (2008). Global search algorithm for nondispersive flow path extraction. *Journal of Geophysical Research*, 113, F04001. <https://doi.org/10.1029/2007JF000964>
- Pilesjö, P., & Hasan, A. (2014). A triangular form-based multiple flow algorithm to estimate overland flow distribution and accumulation on a digital elevation model. *Transactions in GIS*, 18(1), 108–124. <https://doi.org/10.1111/tgis.12015>
- Quinn, P., Beven, K., Chevallier, P., & Planchon, O. (1991). The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological Processes*, 5, 59–79. <https://doi.org/10.1002/hyp.3360050106>
- Quinn, P. F., Beven, K. J., & Lamb, R. (1995). The $\ln(a/\tan\beta)$ index: How to calculate it and how to use it within the TOPMODEL framework. *Hydrological Processes*, 9(2), 161–182. <https://doi.org/10.1002/hyp.3360090204>
- Seibert, J., & McGlynn, B. L. (2007). A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models. *Water Resources Research*, 43, W04501. <https://doi.org/10.1029/2006WR005128>
- Shin, S., & Paik, K. (2017). An improved method for single flow direction calculation in grid digital elevation models. *Hydrological Processes*, 31(8), 1650–1661. <https://doi.org/10.1002/hyp.11135>
- Tarboton, D. G. (1997). A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water Resources Research*, 33(2), 662–670. <https://doi.org/10.1029/96WR03137>
- Xiong, L., Tang, G., Yan, S., Zhu, S., & Sun, Y. (2014). Landform-oriented flow-routing algorithm for the dual-structure loess terrain based on digital elevation models. *Hydrological Processes*, 28(4), 1756–1766. <https://doi.org/10.1002/hyp.9719>
- Yamazaki, D., Ikeshima, D., Sosa, J., Bates, P. D., Allen, G. H., & Pavelsky, T. M. (2019). MERIT hydro: A high-resolution global hydrography map based on latest topography dataset. *Water Resources Research*, 55, 5053–5073. <https://doi.org/10.1029/2019WR024873>
- Zhou, Q., & Liu, X. (2002). Error assessment of grid-based flow routing algorithms used in hydrological models. *International Journal of Geographical Information Science*, 16(8), 819–842. <https://doi.org/10.1080/13658810210149425>
- Zhou, Q., Pilesjö, P., & Chen, Y. (2011). Estimating surface flow paths on a digital elevation model using a triangular facet network. *Water Resources Research*, 47, W07522. <https://doi.org/10.1029/2010WR009961>