Received April 27, 2022, accepted May 29, 2022, date of publication June 3, 2022, date of current version June 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3180175

A Novel Algorithm to Delineate Surface Water Paths on Digital Elevation Model Image With Boundary Element Method

AJALAWIT CHANTAVEEROD[1], KAMPOL WORADIT[2], (Member, IEEE), ANDREW D. SEAGAR[3], (Life Senior Member, IEEE), AND THUNYAWAT LIMPITI[4], (Member, IEEE)

[1]Center of Excellence in Sustainable Disaster Management, Department of Electrical Engineering, School of Engineering and Technology, Walailak University, Tha Sala, Nakhon Si Thammarat 80161, Thailand
[2]Optimization Theory and Applications for Engineering Systems (OASYS) Research Group, Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand
[3]School of Engineering and the Built Environment, Griffith University, Gold Coast, QLD 4222, Australia
[4]Department of Electrical Engineering, School of Engineering and Technology, Walailak University, Tha Sala, Nakhon Si Thammarat 80161, Thailand

Corresponding authors: Ajalawit Chantaveerod (cajalawi@mail.wu.ac.th) and Kampol Woradit (kampol.w@cmu.ac.th)

This work was supported by the Center of Excellence in Sustainable Disaster Management (CESDM) funded by the Agricultural Research Development Agency (Public Organization) under Grant PRP6405032390.

ABSTRACT It is well-known that the delineated surface water paths (SWP) from a DEM image are essential in hydrology. Both grid-based and contour-based algorithms were proposed in the literature to improve the delineated SWP position's accuracy. Compared to the grid-based algorithm, the contour-based algorithms usually give more accurate results but require higher computation resources, especially in case of the wide catchment. This paper introduces a new contour-based algorithm which is formulated with the semi-analytical solution of Laplace's partial differential equation with boundary element method. This approach allows the algorithm to determine the SWP in any direction in order to keep the delineated SWP smooth. The proposed algorithm was validated with the standard synthetic surfaces, where theoretical SWPs are known for accuracy evaluation. The obtained SWPs are more accurate than that of the popular grid-based algorithms. Moreover, the proposed algorithm requires less computation resources when considering very long contour lines. In experimentation with a real DEM image, all delineated SWPs are absolute (no broken part) when the contour interval is equal to or less than 20 meters, and the spacing between adjacent discrete elevation data is 20 meters or narrower. This algorithm helps the hydrologists estimate the catchment area, which is useful for water management in flood and drought prevention.

INDEX TERMS Flow direction algorithm, digital elevation model (DEM), surface water path (SWP), boundary element method (BEM).

I. INTRODUCTION

Digital elevation model (DEM) is a digital representation of a topographic map from remote sensing technology. In some parts of the world, DEM is synonymous with DTM (Digital Terrain Model). In the United States, DTM is defined as a model representing the bare earth surface, where all-natural and built features were removed [1]. Since current data capture technologies such as field survey, photogrammetry, remote sensing or Light Detection and Ranging (LiDAR) enable the high quality DTM, the DTM elevation data is very engaging for land-use planning, geographical modeling, hydrological modeling, and estimation of natural three-dimensional catchments [1], [2]. The size of the catchment area is an essential parameter in the hydrological model for estimating the water flow rate and evaluating the flood risk level. With the concept of a physical-based model (mechanistic or white box model), the size of the catchment area can be physically observed from a catchment model that is drawn by the surface water path (SWP). The catchment model's effectiveness depends on the algorithm that is used to delineate the SWP from DTM. For the last few decades,

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Di Martino.

there have been two main approaches to develop the SWP delineation algorithm, namely i) processing the DEM elevation data in the raster form [3], [4] and ii) processing the data in contour form [5]. To demonstrate that these algorithms are able to identify the water flow path, the SWP position error indicator is often used to measure the accuracy of these SWP delineation algorithms. It calculates the error of the SWP position relative to the theoretical SWPs or the real position of the SWPs. The relation between the resolution of DTM and the accuracy of SWP is not straightforward. For example, increasing the resolution in simple terrain case does not improve accuracy [4]. Therefore, a new algorithm is needed to delineate the SWP from the very large DEM image with adequate accuracy and affordable complexity. To improve the accuracy, an algorithm should exploit the water flow path model in addition to relying on raw data [6].

Numerous researchers give precedence to the macroscopic SWP characteristic, which is the effect of gravitation; therefore, the algorithms are usually formulated with only the streamline flow concept, not including the turbulent flow concept. A streamline is a curve whose tangent at any point gives the direction of the fluid velocity at that point and is time-invariant. This concept is analogous to a line of force in a static electric or magnetic field [7]. The papers [8]–[10] initiated this concept and demonstrated that their algorithms could generate the SWPs using the spatial distribution of slopes in the DTM. These algorithms became the fundamental tools in the hydrological models, and their extensions were grouped into three main approaches, namely the grid-based, TIN-based and contour-based algorithms [11].

In the approach to delineate the SWP with the grid-based algorithm, the D8 algorithm proposed by [12] was the pioneering work, followed by several papers that improve the algorithm's accuracy when the DEM image contains missing pixels. The extensions of its algorithm are divided into two main groups [13]: i) single flow direction (SFD) algorithms, e.g. Rho4 [14], Rho8 and D8-LTD [15] and ii) multiple flow direction (MFD) algorithms, e.g. FDFM [16], DEMON [17] and D∞ [18]. The enhanced D8 algorithms are employed widely in the hydrological model because the implementations are simple and require less computational resources. However, the errors of the grid-based algorithms is sensitive to the accuracy of the DTM [19]. The accuracy of a DTM depends on the pixel size, slope and quantization level [20]–[24]. They also found that the accuracy of the grid-based algorithms is limited with the oversimplified assumptions on the behavior of the water flow [10], [25], [26]. Later, [4] evaluated the accuracy of the six popular algorithms, namely D8, Rho8, D8-LTD, FDFM, MFD, and D∞. They compared the results simulated by these algorithms with the theoretical solutions of SWPs on natural watersheds. Unfortunately, they found that the error of the SWP positions falls in the range of 46.0% to 161.4%. The maximum is gained by D8 or FDFM algorithms, and the minimum by D8-LTD or D∞ approach.

In another approach, the algorithms delineate the SWP from contour lines, which are extracted from a DTM. The [9] initiated the contour-based algorithm with the streamline flow concept of [7], [27] to eliminate the spurious SWPs delineated by the grid-based algorithms in the case of divergent areas. With the algorithm's efficacy, many hydrologists, who need to estimate a sub-catchment area based on the SWP, implemented this algorithm in the catchment model [23], [28]–[32]. To further enhance this algorithm, a procedure that increases the algorithm's robustness while determining the SWP near the ridge is proposed in [29]. Later, the [30] adopt that procedure for partitioning a catchment into contour-based elements as simple and automated as possible. Furthermore, [31], [33]–[35] extend this approach to solve their problems in the hydrological model. However, [5], [32] pointed out that the algorithms mentioned above were still semi-automated and had some problems when delineating the SWP on the complex terrains such as ridges, saddles, and peaks. Finding a fully automated algorithm to deal with complex terrains is one of the most significant research challenges. Then, [5], [32] proposed the modified algorithm by using skeleton construction techniques, whereas [36] created another algorithm that is formulated with Triangulated irregular networks (TINs) structure. TINs were proved to be useful in many hydrological applications [37]–[41]. Unfortunately, the existing algorithms require high computational resources, especially for complex terrains, where the regions of high heterogeneity must be fitted with more data points. Although the existing contour-based algorithms give accurate SWP results, they are not attractive for a hydrological model because the contour-based algorithms still cannot be implemented with affordable computational resources [42].

Even until recently, the accuracy-complexity tradeoff of SWP delineation is still an open problem [4], [42]. Each application of SWP delineation requires a different level of accuracy. Existing algorithms can delineate SWP with sufficient accuracy for applications in geomatics, geographic information systems (GIS), mapping, and surveying [43]. On the other hand, the rainfall-runoff model application, which simulates the process to learn the water cycle for decision making and future prediction, requires higher accuracy, which is not achieved by existing algorithms [44]. One of the important applications is the sub-catchment area estimation, which is the essential input parameter of all hydrological models for studying hydrologic phenomena, hydrologic cycle, the impact of climate change, and soil properties on hydrology and water resources. This research aims to develop an algorithm with reasonable complexity to delineate the SWP with sufficient accuracy for sub-catchment area estimation. Obviously, there are two possible approaches: i) improving the accuracy of the grid-based algorithm and ii) reducing the complexity of the contour-based algorithm. Among the existing algorithms, both grid-based and contour-based algorithms delineate the SWPs by using elevation and slope data over the whole region, which is discretized into hydraulically connected elements such as grids or triangular elements. The grid-based algorithm used to delineate the SWP is more popular than the contour-based algorithm because of its simplicity and fewer computational resource requirements. However, it selects the most appropriate pixel as the endpoint of the SWP segment by observing only the surrounding eight pixels, which causes a local error. The local inaccuracy propagates to other pixels and accumulates into a serious error. With the limited resolution of the DEM image, the accuracy at each pixel cannot be improved, and hence the error issue of the grid-based algorithm cannot be mitigated. On the other hand, the contour-based algorithm gives high accuracy but must solve three-dimensional unknown variables, where the number of grid points between two adjacent contour lines considerably increases with the required smoothness. Thus, it is too complicated to implement a distributed hydrological model [42].

Recently, it was pointed out that the existing algorithms for delineating the SWP over DTM image do not meet the required accuracy of hydrological model development which is the important tool for understanding, predicting, and managing water resources [4], [42]. Increasing the DEM resolution in grid-based algorithms does not improve the position accuracy (error floor) since these algorithms exploit only neighboring pixels without extracting topographic information from larger surrounding area [4]. This motivated us to propose a new flow direction algorithm that exploits topographic information to significantly improve the position accuracy and reaches the error floor at higher DTM resolution. Also, the new algorithm must not incur an excessive computational time. The boundary element method (BEM), which provides an outstanding accuracy but demands high computational performance and large memory, becomes more practical with modern computers because it solves a 3-D problem with 2-D domain. For example, BEM was successfully used to develop a theoretical flow model for any potential function in [45]–[49]. Therefore, the practical BEM-based algorithm is proposed to delineate SWP over DEM image. The proposed algorithm is derived from numerical solutions of the governing partial differential equations.

The paper's central research objectives are as follows.

- The elevation function in the DEM image was known as the contour lines obtained from image processing technique [50]. The elevation function between adjacent contour lines is unknown and can be described by 2-dimensional Laplace's partial differential equation, where the known contour lines serve as the boundary values. Our first objective is to formulate the semi-analytical solution of Laplace's partial differential equation with BEM, which discretizes the problem domain along the boundary into piecewise linear curves, where the domain can be parameterized by one variable. The semi-analytical solution includes the topographic information, which improves the delineated SWP accuracy. Moreover, comparing to the TIN structure, the problem domain is reduced from an area between contour lines to contour lines. With the significant accuracy improvement and practical computational complexity, the semi-analytical solution makes a major contribution to the urgent needs of hydrologists to cross the current limit of hydrological modeling.
- The next step is to delineate every SWP, which is a path starting from a source and ending at a sink, one by one. All sources are the points on contour lines, while all sinks are the points between contour lines. For each SWP, we first determine the SWP direction at a source and then iteratively determine the SWP direction until reaching a sink. In doing that, we propose an algorithm that uses the previously obtained elevation function as well as its first and second derivatives in the polar coordinates to determine the SWP direction at every iteration from the source to the sink. Note that we can also delineate the SWP in the reverse direction from the sink to the source because our algorithm basically finds the steepest downslope or steepest upslope direction. Compared to other contour-based algorithms, which delineate an SWP by connecting two points on a pair of adjacent contour lines, our algorithm gives greater continuity along the SWP. Compared to grid-based algorithms, which limit the number of directions to eight at every pixel, our algorithm gives a more exact direction at every point along the SWP. The continuity and smoothness of the proposed algorithm mitigate the issue of grid-based algorithms that cease delineating SWP when they reach a flat plane surface.
- In practice, the proposed algorithm is applied with real terrain and must be able to cope with two causes of complexity rising. First, the contour line encircling a foothill can be very long. Second, DTM with high resolution should be sampled at high sampling rate. Both of them creates a great number of sampling points, which incur a time-consuming computation. The third objective is to design the algorithm that can delineate SWPs with incomplete contour lines, which have missing contour segments and much less sampled points as shown in Fig. 1. Therefore, the proposed algorithm requires less computational resources than other contour-based algorithms.

The remainder of this article is structured as follows. Section II proposes the numerical solution of 2-dimensional Laplace's partial differential equation formulated with BEM. This solution is the elevation function that gives the elevation at any coordinate and is defined by contour integrals, where the elevations along the contours are known. Next, in Section III, we present how to use the elevation data in a DEM image to identify the gradient of the elevation along the contour lines which are the unknown boundary value functions. Both the elevation and the obtained gradient of the elevation along the contour lines are used to interpolate the elevation at any point between the adjacent contour lines in Section IV. Next, Section V presents how to use the contour integrals and the gradient terms for determining the SWP direction at any point along the delineated SWP, and describes the proposed algorithm for delineating the SWP from DEM elevation. In Section VI, we demonstrate and discuss the

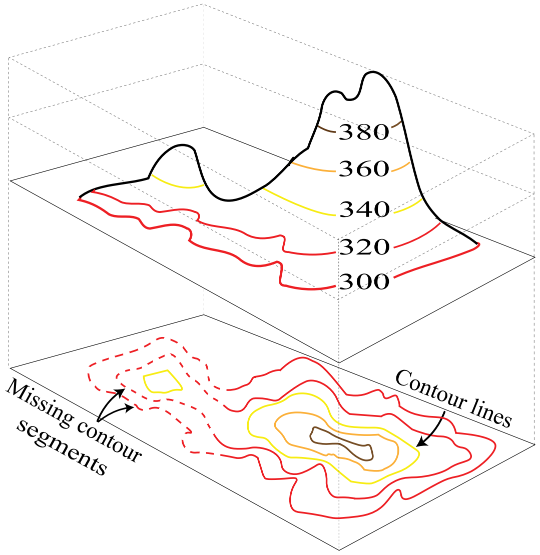


FIGURE 1. Incomplete contour lines.

efficiency of the proposed algorithm on delineating SWPs over both synthetic surfaces and the real terrain. The synthetic surfaces are used to evaluate the position error of the delineated SWPs. The real terrain with very long contour lines is used to validate the proposed algorithm. Last, we conclude the paper in Section VII.

II. PROPOSED FORMULATION OF LAPLACE'S EQUATION WITH BEM

This section proposes the formulation of elevation function, which is the essential function for analyzing the directions of SWPs at any coordinates between the DEM contour lines. To improve the elevation interpolation in the contour-based algorithm of SWP delineation, numerous researchers pay attention to the elevation function formulation with numerical techniques. Reference [42] discusses the existing formulations of the elevation function in the literature, namely the skeleton construction technique and Triangulated Irregular Networks (TINs) technique, and points out that these techniques are not attractive for implementing the distributed hydrological model because of high computational cost. These techniques generate the grid points bounded by DEM contour lines, so they require a large memory size for collecting the values of the elevation functions at all grid points, especially in case of high resolution and/or large domain areas. Moreover, all unknown variables, which are the elevation data at grid points, must be solved even though only some of them are needed for delineating SWPs. To tackle those issues, we propose the integral contour formulation derived from the analytical solution of 2-dimensional Laplace's partial differential equation. In particular, BEM is applied to reformulate the analytical solution for twofold benefits. First, the computational cost is greatly reduced because the unknown variables are elevation gradients on DEM contour lines. Second, not all unknown variables are needed to be solved. If the unknown variables are elevation gradients on a segment of DEM contour lines, and that segment is not close to SWPs, then they are not needed to be solved.

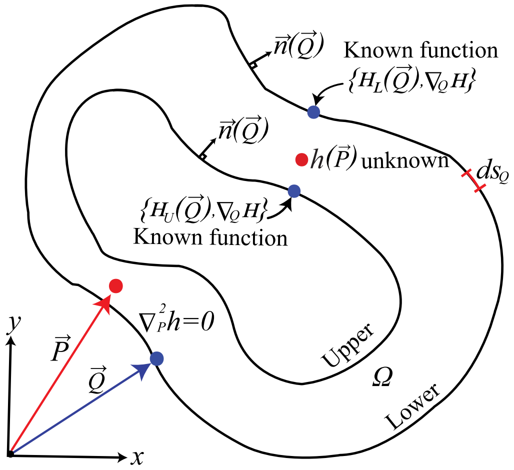


FIGURE 2. Elevation function $h(\vec{P})$ identified by contour lines: $H_U(\vec{Q})$ and $H_L(\vec{Q})$.

A. ANALYTICAL SOLUTION OF LAPLACE'S PDE

A flow line is a path starting at a point on a contour line and ending at a point on the adjacent contour line. With the streamline flow concept, the flow line directions at starting and ending points are perpendicular to their contour lines. Hence, the flow line directions at starting and ending points can be found by calculating the elevation gradients at these points. However, the flow line directions at other points on the path are unknown because the elevation data outside contour lines are not available. There are several algorithms that interpolate these elevation data [5], [9], [27], [29]–[32], [34]. Different types of interpolation give different elevation functions and different patterns of SWP. The most accurate algorithm finds the analytical solution of Laplace's partial differential equation in the context of the conservative vector fields. The solution is used to interpolate terrain surface from the elevation data along the contour lines [51]. Laplace's partial differential equation is considered, because the boundary value functions, including elevation data and their spatial derivatives, are time-invariant. The solution is solved by the boundary integral equation method. Because of its accuracy, we apply this analytical solution of Laplace's partial differential equation for interpolation. The interpolated elevation data and spatial derivatives are then used to determine the directions at all points along the flow line as shown in Section V.

Fig. 2 presents the point $\vec{P}$ in a 2-dimensional domain Ω, covering the region, which is composed of a homogeneous media. To interpolate the elevation data at the point $\vec{P}$, [51] applied Laplace's partial differential equation to formulate the governing equation of the problem for the scalar elevation function $h(\vec{P})$:

$$\nabla^2 h(\vec{P}) = \left(\frac{\partial^2 h}{\partial x^2}\right) + \left(\frac{\partial^2 h}{\partial y^2}\right) = 0. \qquad (1)$$

Many kind of techniques can be used to solve the Laplace's equation. In [51], they selected the boundary integral equation method with Green's theorem [52] to solve the analytical solution of Laplace's equation. The obtained particular solution can be written as

$$\psi h(\vec{P}) = \oint_C \ln|\vec{P}-\vec{Q}| \left(\nabla H(\vec{Q}) \cdot \hat{n}(\vec{Q})\right) ds_Q - \oint_C \left(\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2} \cdot \hat{n}(\vec{Q})\right) H(\vec{Q})\, ds_Q. \quad (2)$$

The boundary C contains any point where the elevation is known and that point is on any contour line. The boundary C may contain points on several contour lines. For example, the boundary C contains the points on both the lower contour line $H_L(\vec{Q})$ and the upper contour line $H_U(\vec{Q})$. The boundary C is not only the domain of the known elevation function $H(\vec{Q})$ but also the domain of the potential gradient $\nabla H(\vec{Q})$. The direction of the boundary C at the point Q is specified with a unit normal vector $\hat{n}(\vec{Q})$. The value of ψ is determined by a point $\vec{P}$'s neighborhood, which is the part of a small disc around a $\vec{P}$ point inside the boundary C. For example, considering a point $\vec{P}$ on a contour line, the neighborhood of a point $\vec{P}$ is a half-disc, and so $\psi = \pi$. If the point $\vec{P}$ is between the contour lines, then the neighborhood of a point $\vec{P}$ is a full disc, and so $\psi = 2\pi$.

The elevation function $h(\vec{P})$ in (2) can be used to model the SWP accurately with an assumption that no contour line is missing, when the domain Ω is bounded by the contour lines C_j, where j is the index of a contour line. For example, considering two adjacent contour lines with different elevations, the obtained SWP will be a path from the upper contour line to the lower contour line. As another example, if a contour line obstructs between two adjacent contour lines, the path of SWP will make a detour around the obstructing contour line. Moreover, this analytical approach can generate the continuous SWP over the domain Ω and the characteristics of the obtained SWP conform to the streamlined concept. The obtained SWP represents the water flow from the upper to the lower contour line. The flow is guaranteed not to be in a reverse direction because the value of the elevation function $h(\vec{P})$ is bounded between the local minima (the elevation value of the contour line $H_L(\vec{Q})$) and the local maxima (the elevation value of the contour line $H_U(\vec{Q})$).

B. PROPOSED SOLUTION WITH BEM FOR DEM CONTOUR LINES

The analytical solution mentioned in (2) can reproduce the elevation function $h(\vec{P})$ from the known elevation on the boundary $H(\vec{Q})$. However, it cannot promptly interpolate the elevation data from DEM contour lines. Here, we aim to take advantage of both the analytical solution and the DEM image. There are two ways to do interpolation between the contour lines for SWP delineation.

First, we must represent the DEM elevation data along the contour lines into the elevation function in analytic form or closed-form. Then, we calculate its gradient in the analytic form. The available elevation data may be employed to approximate the function $H(\vec{Q})$, but it is very complicated to approximate the gradient function along the fixed contour lines. In practice, the contour-based DEM elevation data cannot be automatically described with an analytic function on the computer because the elevation is random and vast. Hence, this paper proposes the second way, where we approximate the DEM elevation data along a contour line with many piecewise Lipshitz functions [53]. Then, we apply these piecewise functions in the BEM process to obtain the numerical solution by reformulating the analytical solution in (2). In the mentioned analytical solution, a whole boundary integral is replaced with the definite boundary integrals on the piecewise intervals. The number of intervals equals the number of piecewise elevation functions. The numerical solution is formulated as follows.

Consider a boundary C. The boundary integral of (2) is split into N definite integrals, each of which is indexed by $j = 1 \ldots N$. Each interval is denoted by C_j. Then, we obtain the semi-analytical solution:

$$\psi h(\vec{P}) = \sum_{j=1}^{N} \int_{C_j} \ln|\vec{P}-\vec{Q}| \left(\nabla H(\vec{Q}) \cdot \hat{n}(\vec{Q})\right) ds_Q - \sum_{j=1}^{N} \int_{C_j} \left(\frac{\left(\vec{P}-\vec{Q}\right) \cdot \hat{n}(\vec{Q})}{|\vec{P}-\vec{Q}|^2}\right) H(\vec{Q})\, ds_Q. \quad (3)$$

For example, we implement (3) for two adjacent contour lines. In Fig. 3, the two contour lines from a DEM image are the boundary which is surrounding the interior region Ω. When we select N sample points along two contour lines, the lower and upper contour lines contain N_{C_L} and N_{C_U} intervals, respectively. Then, we obtain

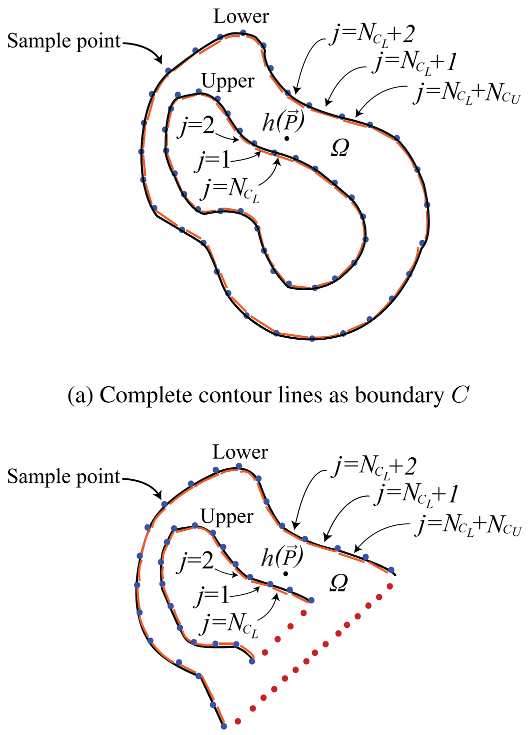
$$\psi h(\vec{P}) = \sum_{j=1}^{N_{C_L}+N_{C_U}} \int_{C_j} \ln|\vec{P}-\vec{Q}| \left(\nabla H(\vec{Q}) \cdot \hat{n}(\vec{Q})\right) ds_Q - \sum_{j=1}^{N_{C_L}+N_{C_U}} \int_{C_j} \left(\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2} \cdot \hat{n}(\vec{Q})\right) H(\vec{Q})\, ds_Q \quad (4)$$

The boundary values defined in the semi-analytical solution comprise the elevation function $H(\vec{Q})$ and the normal component of the potential gradient $\nabla H(\vec{Q})$. The normal component of $\nabla H(\vec{Q})$ is $\nabla H(\vec{Q}) \cdot \hat{n}(\vec{Q})$ and denoted by ∇H_n. Both functions on a point in the interval C_j are approximated with the polynomial functions of a dummy variable s:

$$H_j(s) = \zeta_j \frac{s}{L_j} + \xi_j, \quad (5)$$

$$\nabla H_{n,j}(s) = \alpha_j \frac{s^3}{L_j^3} + \beta_j \frac{s^2}{L_j^2} + \gamma_j \frac{s}{L_j} + \eta_j, \quad (6)$$

where $\zeta_j, \xi_j, \alpha_j, \beta_j, \gamma_j$ and η_j are constants, and s is the distance from the middle of segment C_j, and L_j is the length of segment



(a) Complete contour lines as boundary C

(b) Incomplete contour lines as boundary C

FIGURE 3. Contour lines that are used as boundary *C* and are split into *N* small intervals.

C_j and is written as

$$-\frac{L_j}{2} \leq s \leq \frac{L_j}{2}. \tag{7}$$

The accuracy of this approximation depends on two conditions. First, the elevation function is smooth within each interval, so that the potential gradient is continuous. Second, the number of definite split integrals N is large enough that the difference between the original and the approximated contour line is small. Note that the larger number of split definite integrals incurs a longer computational time. In fact, increasing the polynomial orders of $H_j(s)$ and $\nabla H_{n,j}(s)$ also improves the approximation accuracy, but the computational time sharply increases. Note that this approximation can be applied with both complete and incomplete contour lines as shown in Fig. 3 (a) and (b), respectively. This approximation gives the polynomial coefficients that will be used to determine the boundary value function in the numerical solutions in Section III-A.

With polynomial approximation, we substitute $H_j(s)$ and $\nabla H_{n,j}(s)$, defined in (5) and (6), respectively, for $H(\vec{Q})$ and $\nabla H_n(\vec{Q})$ in (3). The semi-analytical solution in (3) becomes

$$\begin{aligned}\psi\ h(\vec{P}) = \sum_{j=1}^{N} \Big(& \alpha_j F_{\alpha_j}(\vec{P},\vec{Q}) + \beta_j F_{\beta_j}(\vec{P},\vec{Q}) \\ &+ \gamma_j F_{\gamma_j}(\vec{P},\vec{Q}) + \eta_j F_{\eta_j}(\vec{P},\vec{Q})\Big) \\ &- \sum_{j=1}^{N} \Big(\zeta_j F_{\zeta_j}(\vec{P},\vec{Q}) + \xi_j F_{\xi_j}(\vec{P},\vec{Q})\Big)\end{aligned} \tag{8}$$

where

$$F_{\alpha_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s^3}{L_j^3} \ln|\vec{P}-\vec{Q}|\ ds \tag{9}$$

$$F_{\beta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s^2}{L_j^2} \ln|\vec{P}-\vec{Q}|\ ds \tag{10}$$

$$F_{\gamma_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s}{L_j} \ln|\vec{P}-\vec{Q}|\ ds \tag{11}$$

$$F_{\eta_j}(\vec{P},\vec{Q}) = \int_{C_j} \ln|\vec{P}-\vec{Q}|\ ds \tag{12}$$

$$CCLF_{\zeta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s}{L_j} \left(\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2} \cdot \hat{n}(\vec{Q})\right) ds \tag{13}$$

$$F_{\xi_j}(\vec{P},\vec{Q}) = \int_{C_j} \left(\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2} \cdot \hat{n}(\vec{Q})\right) ds, \tag{14}$$

The F_{α_j}, F_{β_j}, F_{γ_j}, F_{η_j}, F_{ζ_j} and F_{ξ_j} are geometric terms, referred to as shape functions. They are the functions of the boundary (contour line's path) and the position of the point $\vec{P}$. Both boundary and the point $\vec{P}$ will be found in Section III.

III. ANALYSIS OF BOUNDARY VALUE FUNCTION

This section presents how to use the DEM contour line for identifying the boundary value functions in (8). The DEM contour line is a discrete sequence of referenced points located along the contour line, and is available from both grid-based DEM and contour-based DEM [54]. The discrete sequence of points is used to calculate the elevation function $H(\vec{Q})$ along the boundary and the six shape functions $F_{*_j}(\vec{P},\vec{Q})$ in Section III-A according to our numerical solution formulation in (8). Then, the obtained $H(\vec{Q})$ is used to numerically compute the gradient of the elevation function along the contour line as shown in Section III-B. The gradient will be used to interpolate the elevation function between the contour lines and to delineate the streamlines in Section V.

A. CALCULATION OF ELEVATION FUNCTION ON THE CONTOUR LINES AND THE SHAPE FUNCTIONS

To implement the proposed numerical solution, we use the information of the DEM contour lines which comprise the coordinates of the 2D points along the DEM contour lines and the elevations at those points. We use the position information of the points along the boundary C_j for calculating the integration of the shape functions as shown in (9) to (14), and use the elevation levels at those points for approximating the elevation function $H_j(\vec{Q})$ along subdomain C_j as shown in (5).

In Fig. 3 (a), the DEM contour lines were systematically sampled into points, whose coordinates define the points $\vec{Q}$ along the subdomain C_j. The DEM was also sampled into points to define the points $\vec{P}$ in the domain Ω. The coordinates of both $\vec{Q}$ and $\vec{P}$ are used to calculate the integration of the shape functions, namely F_{α_j}, F_{β_j}, F_{γ_j}, F_{η_j}, F_{ζ_j} and F_{ξ_j}. The subdomain C_j comprises the points along the straight line from point $\vec{Q}_j$ to point $\vec{Q}_{j+1}$. Its direction is from $\vec{Q}_j$ to $\vec{Q}_{j+1}$, and is used to identify the unit normal vector $\hat{n}(\vec{Q})$. Before integrating the shape functions, we transform the coordinates of $\vec{P}$ and $\vec{Q}$ into the local coordinates which are defined with the variable s so that the line integration can be done in closed-form.

After defining the coordinates of the point $\vec{Q}$ on each subdomain C_j, we use the elevation data at those points for approximating the elevation function $H_j(\vec{Q})$. The elevation levels at $\vec{Q}_j$ and $\vec{Q}_{j+1}$ are used to calculate the coefficients of the approximated functions in (5). Usually, DEM image contains both ordinary contour lines and incomplete contour lines. In case of ordinary contour lines, each contour has the constant elevation data denoted H_0, so the coefficients η and ζ are zero and H_0, respectively. In case of very long or incomplete contour lines, the elevation levels over the gap of the contour lines are not constant, and we approximate the varying elevation levels on subdomain C_j by piecewise linear function as shown in Fig. 3 (b). For example, the elevation levels at the point $\vec{Q}_j$ and the point $\vec{Q}_{j+1}$ are H_j and H_{j+1}, respectively. The coefficients η_j and ζ_j are the slope and $H_j(\vec{Q})$-intercept, respectively.

B. ANALYSIS OF ELEVATION GRADIENT ON THE CONTOUR LINES

To present the process to determine four unknown coefficients for any subdomain of the boundary integral, we divide the whole boundary C into the N small boundary C_j, where $j = 1, \ldots, N$. The process is structured as follows.

1) OBSERVING ALONG SUBDOMAIN C_i

In Section II-A, the function reproduced by the analytical solution is the elevation function $h(\vec{P})$, where the point $\vec{P}$ is in either domain Ω or boundary C. In Section II-B, we put the observed point $\vec{P}$ in the domain Ω, and we employ the reproduced function from the numerical solution for interpolation at the points between adjacent contour lines. Here, we put the observed point $\vec{P}$ along the boundary C and apply the numerical solution for determining the gradient terms. Now, we have the $4N$ unknown coefficients of the function $\nabla H_n(\vec{Q})$; therefore, we will put the observed point $\vec{P}$ at $4N$ different positions along the boundary C.

To identify the gradient terms in the numerical solution, we resolve the four unknown coefficients, which comprise α_j, β_j, γ_j, and η_j, of the function $\nabla H_n(\vec{Q})$ (defined in (6)) in each boundary C_j by solving a set of linear equations. We formulate a set of four equations with all unknown coefficients by selecting four different observed points $\vec{P}$ along a subdomain C_i, where C_i denotes the i-th subdomain of the boundary C and is the location of four selected points. Along the boundary C_i, we put the observed points $\vec{P}$ at $\frac{s}{L_i} \in \{-\frac{1}{2}, -\frac{1}{4}, 0, \frac{1}{2}\}$. For example, when putting $\vec{P}$ at $s = -1/2$ on C_i, we obtain:

$$\begin{aligned}
&\psi h\left(\vec{P}\left(-\frac{1}{2}\right)\right) \\
&= \sum_{j=1}^{N}\left(\alpha_j F_{\alpha_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right)\right. \\
&\quad + \beta_j F_{\beta_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) + \gamma_j F_{\gamma_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) \\
&\quad \left. + \eta_j F_{\eta_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right)\right) - \sum_{j=1}^{N}\left(\zeta_j F_{\zeta_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right)\right. \\
&\quad \left. + \xi_j F_{\xi_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right)\right),
\end{aligned} \tag{15}$$

where $h(\vec{P}\left(-\frac{1}{2}\right))$ can be determined with the elevation function $H_i(s = -\frac{1}{2})$ in (5):

$$H\left(-\frac{1}{2}\right) = -\frac{1}{2}\zeta_i + \xi_i. \tag{16}$$

The shape functions of $\nabla H_n(\vec{Q})$ can be determined with the integrations:

$$\begin{aligned}
F_{\alpha_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) &= \int_{C_j} \frac{s^3}{L_j^3} \ln\left|\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}\right| \, ds_Q, \\
F_{\beta_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) &= \int_{C_j} \frac{s^2}{L_j^2} \ln\left|\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}\right| \, ds_Q, \\
F_{\gamma_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) &= \int_{C_j} \frac{s}{L_j} \ln\left|\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}\right| \, ds_Q, \\
F_{\eta_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) &= \int_{C_j} \ln\left|\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}\right| \, ds_Q,
\end{aligned} \tag{17}$$

and the shape functions of $H(\vec{Q})$ can be determined with the integrations:

$$\begin{aligned}
F_{\zeta_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) &= \int_{C_j} \frac{s}{L_j} \frac{\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}}{\left|\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}\right|^2} \cdot \hat{n}(\vec{Q}) ds_Q \\
F_{\xi_j}\left(\vec{P}\left(-\frac{1}{2}\right), \vec{Q}\right) &= \int_{C_j} \frac{\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}}{\left|\vec{P}\left(-\frac{1}{2}\right) - \vec{Q}\right|^2} \cdot \hat{n}(\vec{Q}) ds_Q.
\end{aligned} \tag{18}$$

Similarly, when putting $\vec{P}$ at $s = -\frac{1}{4}$, 0 and $\frac{1}{4}$ on C_i, $h(\vec{P}\left(-\frac{1}{4}\right))$, $h(\vec{P}(0))$ and $h(\vec{P}\left(\frac{1}{4}\right))$ can be determined with the elevation function $H_i\left(s = -\frac{1}{4}\right)$, $H_i(s = 0)$ and $H_i\left(s = \frac{1}{4}\right)$, respectively.

2) REWRITING WITH VECTOR EQUATION

Now, we obtain a set of four equations after setting $\vec{P}$ at four different positions on the subdomain C_i. Obviously, they are vector equations and can be rewritten in an alternative form:

$$[\boldsymbol{\psi}_i][\boldsymbol{H}_i] = [\boldsymbol{F}_{\nabla H,ij}][\boldsymbol{\nabla H}_{n,j}] - [\boldsymbol{F}_{H,ij}][\boldsymbol{H}_j] \tag{19}$$

where $i = 1, \cdots, N$ is the index of each set of four $\vec{P}$ positions on the same subdomain C_i, and $j = 1, \cdots, N$ is the index of subdomain C_j, which serves as the j-th subdomain of the boundary integration. $[\boldsymbol{H}_i]$ is 2×1 boundary vector, which contain two coefficients of a function $H(\vec{P})$ along C_i and is defined as

$$[\boldsymbol{H}_i]^T = \begin{bmatrix} \zeta_i & \xi_i . \end{bmatrix}. \tag{20}$$

$[\boldsymbol{H}_j]$ is the concatenation of all $[\boldsymbol{H}_i]$ and becomes $2N \times 1$ boundary vector, which is written as

$$[\boldsymbol{H}_j]^T = \begin{bmatrix} \zeta_1\ \xi_1 & \cdots & \zeta_i\ \xi_i & \cdots & \zeta_N\ \xi_N \end{bmatrix}, \tag{21}$$

$[\nabla \boldsymbol{H}_{n,j}]$ is a $4N \times 1$ boundary vector, which contains four coefficients of a function $\nabla H_n(\vec{Q})$ along C_j and is defined as

$$[\nabla \boldsymbol{H}_{n,j}]^T = \begin{bmatrix} *_1 & \cdots\ \alpha_j\ \beta_j\ \gamma_j\ \eta_j & \cdots\ *_N . \end{bmatrix}, \tag{22}$$

$[\boldsymbol{\psi}_i]$ is a $4 \times 2N$ block diagonal matrix, which contains four neighborhood factors of each set of four $\vec{P}$ positions along C_i and is defined as

$$[\boldsymbol{\psi}_i] = \begin{bmatrix} 0 \cdots & -\frac{1}{2}\psi_i(-\frac{1}{2}) & \psi_i(-\frac{1}{2}) & \cdots 0 \\ 0 \cdots & -\frac{1}{4}\psi_i(-\frac{1}{4}) & \psi_i(-\frac{1}{4}) & \cdots 0 \\ 0 \cdots & 0 & \psi_i(0) & \cdots 0 \\ 0 \cdots & \frac{1}{4}\psi_i(\frac{1}{4}) & \psi_i(\frac{1}{4}) & \cdots 0 \end{bmatrix}. \tag{23}$$

$[\boldsymbol{F}_{H,ij}]$ is $4 \times 2N$ geometry matrix, which contains two integrals of shape functions $F_{\zeta_j}(\vec{P}_i)$ and $F_{\xi_j}(\vec{P}_i)$ at $\vec{P}$ on C_i and at $\vec{Q}$ on C_j and is defined as

$$[\boldsymbol{F}_{H,ij}] = \begin{bmatrix} * \cdots & F_{\zeta_j}(-\frac{1}{2}) & F_{\xi_j}(-\frac{1}{2}) & \cdots * \\ * \cdots & F_{\zeta_j}(-\frac{1}{4}) & F_{\xi_j}(-\frac{1}{4}) & \cdots * \\ * \cdots & F_{\zeta_j}(0) & F_{\xi_j}(0) & \cdots * \\ * \cdots & F_{\zeta_j}(\frac{1}{4}) & F_{\xi_j}(\frac{1}{4}) & \cdots * \end{bmatrix}. \tag{24}$$

$[\boldsymbol{F}_{\nabla H,ij}]$ is $4 \times 4N$ geometry matrix, which contains four integrals of shape functions $F_{\alpha_j}(\vec{P}_i)$, $F_{\beta_j}(\vec{P}_i)$, $F_{\gamma_j}(\vec{P}_i)$ and $F_{\eta_j}(\vec{P}_i)$ at $\vec{P}$ on C_i and at $\vec{Q}$ on C_j and is defined as $[\boldsymbol{F}_{\nabla H,ij}]$, as shown at the bottom of the next page.

3) SOLVING EQUATION SYSTEM

To create the equation system for determining the $[\nabla \boldsymbol{H}_{n,j}]$ vector, we repeat the process of equation formulation with (19) for all subdomains C_i, with $i = 1 \ldots N$. Then, we obtain N sets of four vector equations, which can be written as a linear equation system:

$$\begin{bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_i \\ \vdots \\ \boldsymbol{\psi}_N \end{bmatrix} [\boldsymbol{H}_i] = \begin{bmatrix} \boldsymbol{F}_{\nabla H,1j} \\ \vdots \\ \boldsymbol{F}_{\nabla H,ij} \\ \vdots \\ \boldsymbol{F}_{\nabla H,Nj} \end{bmatrix} [\nabla \boldsymbol{H}_{n,j}] - \begin{bmatrix} \boldsymbol{F}_{H,1j} \\ \vdots \\ \boldsymbol{F}_{H,ij} \\ \vdots \\ \boldsymbol{F}_{H,Nj} \end{bmatrix} [\boldsymbol{H}_j]. \tag{25}$$

Next, we obtain the gradient vector $[\nabla \boldsymbol{H}_{n,j}]$ by solving the equation system:

$$[\nabla \boldsymbol{H}_{n,j}] = \begin{bmatrix} \boldsymbol{F}_{\nabla H,1j} \\ \vdots \\ \boldsymbol{F}_{\nabla H,ij} \\ \vdots \\ \boldsymbol{F}_{\nabla H,Nj} \end{bmatrix}^{-1} \left(\begin{bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_i \\ \vdots \\ \boldsymbol{\psi}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{F}_{H,1j} \\ \vdots \\ \boldsymbol{F}_{H,ij} \\ \vdots \\ \boldsymbol{F}_{H,Nj} \end{bmatrix} \right) [\boldsymbol{H}_j]. \tag{26}$$

IV. CALCULATION OF ELEVATION AND ITS GRADIENTS BETWEEN THE CONTOUR LINES

The existing contour-based algorithm delineates SWP by employing the interpolation techniques that must discretize the problem domain, which is the interior area between two adjacent contour lines. The unknown elevations over the whole domain must be solved to do the interpolation. This incurs high complexity in a process to delineate the SWP. To reduce the computational resource requirements, we propose the process to delineate the SWP by interpolating only some points between the contour lines. The proposed process focuses only the points which are used to delineate the SWP, and interpolate the elevation data at those points and their neighborhoods. We solve the elevation function, representing the change of elevation levels at those points and their neighborhoods by the proposed numerical solution in (8) in Section II-B. The obtained elevation function will be used to determine the direction of an SWP in Section V. The elevation function comprises the elevation function $h(\vec{P})$ and its derivatives at the point $\vec{P}$. Therefore, this section presents how to interpolate the elevation at the points between the contour lines and how to calculate the first and the second derivatives of the elevation function.

A. INTERPOLATED ELEVATION FUNCTION OVER REGION BETWEEN DEM CONTOUR LINES

Delineating SWP between DEM contour lines is to recursively calculate the next $\vec{P}$, starting from a chosen point between contour lines. The recursion will lead $\vec{P}$ to the contour line with lower elevation. The recursive algorithm involves with the elevation function and its derivative. The elevation function $h(\vec{P})$ is formulated with the proposed numerical solution in (8) in Section II-B. Firstly, we rewrite (8) in the vector form:

$$\psi h(\vec{P}) = \left[\boldsymbol{G}_{\nabla H,j}(\vec{P})\right] \left[\nabla \boldsymbol{H}_{n,j}\right] - \left[\boldsymbol{G}_{H,j}(\vec{P})\right] \left[\boldsymbol{H}_j\right], \tag{27}$$

where $\boldsymbol{G}_{H,j}(\vec{P})$ is $1 \times 2N$ geometry vector which contains two integrals of shape functions: $F_{\zeta_j}(\vec{P})$ and $F_{\xi_j}(\vec{P})$. The integration results are in (13) and (14) in Section II-B. Therefore, $\boldsymbol{G}_{H,j}(\vec{P})$ is defined as

$$[\boldsymbol{G}_{H,j}(\vec{P})] = \begin{bmatrix} * \cdots & F_{\zeta_j}(\vec{P}) & F_{\xi_j}(\vec{P}) & \cdots * \end{bmatrix},$$

$[\boldsymbol{H}_j]$ is $2N \times 1$ boundary vector which contains two coefficients of a function $H(\vec{Q})$ along C_j. These coefficients are equal to the components of boundary vector $[\boldsymbol{H}_j]$ in (26) and are used in solving the boundary value function in Section III-B. $[\boldsymbol{G}_{\nabla H,j}(\vec{P})]$ is $1 \times 4N$ geometry vector which contains four integrals of shape functions: $F_{\alpha_j}(\vec{P})$, $F_{\beta_j}(\vec{P})$, $F_{\gamma_j}(\vec{P})$ and $F_{\eta_j}(\vec{P})$. The integration results are in (9), (10), (11), and (12) in Section III-A. Therefore, $[\boldsymbol{G}_{\nabla H,j}(\vec{P})]$ is defined as

$$[\boldsymbol{G}_{\nabla H,j}(\vec{P})] = \left[* \cdots \; F_{\alpha_j}(\vec{P}) \; F_{\beta_j}(\vec{P}) \; F_{\gamma_j}(\vec{P}) \; F_{\eta_j}(\vec{P}) \quad \cdots *\right], \tag{28}$$

$[\boldsymbol{\nabla H}_{n,j}]$ is $4N \times 1$ boundary vector which contains four coefficients of a function $\boldsymbol{\nabla H}(\vec{Q})$ along C_j. These coefficients are solved with the equation system in (26) in Section III-B3.

B. GRADIENT OF INTERPOLATED ELEVATION FUNCTION OVER REGION BETWEEN DEM CONTOUR LINES

There are several ways to find the derivative of elevation function. The slope at a point is the essential information to delineate SWP. Most methods calculate the slope at a designated point from the interpolated elevations of two points. The interpolation in existing methods is done in several directions, and only the one that gives the steepest slope is chosen. In this section, we present a different approach, where the slope can be immediately calculated from a closed-form formula. Specifically, the closed-form formula is derived by taking the derivation of the numerical solution in (3) at the point $\vec{P}$. After obtaining the boundary value functions in Section III, namely the elevation function and its gradient on the contour lines, we use them to calculate the interpolated elevation function and the slope at same point $\vec{P}$, mentioned in Section IV-A. The slope $\nabla_P h(\vec{P})$ at point $\vec{P}$ comprises the derivative of the elevation function $h(\vec{P})$ with respect to x (h_x) and the derivative of the elevation function $h(\vec{P})$ with respect to y (h_y). To obtain two derivatives at point $\vec{P}$, we determine the gradient of a function $h(\vec{P})$ by applying the vector operator ∇ to the scalar function $h(\vec{P})$. The scalar function $h(\vec{P})$ is reproduced from the proposed semi-analytical solution in (3). The process to calculate the gradient of elevation function $h(\vec{P})$ at the point between the DEM contour lines is structured as follows.

1) FORMULATING THE GRADIENT OF PROPOSED SOLUTION

Again, we apply (3) in Section II-B to formulate the gradient of semi-analytical solution at point $\vec{P}$. Taking the gradient operator to (3) gives

$$\psi\left[\nabla_P h(\vec{P})\right] = \sum_{j=1}^{N} \int_{C_j} [\nabla_P \Theta] \left(\nabla H(\vec{Q}) \cdot \hat{n}(\vec{Q})\right) ds_Q - \sum_{j=1}^{N} \int_{C_j} [\nabla_P \Xi]\, H(\vec{Q})\, ds_Q, \tag{29}$$

where $\nabla_P \Theta$ and $\nabla_P \Xi$ are gradient terms in brackets, and can be calculated analytically. Then, we obtain the resulting expression as

$$\nabla_P \Theta = \frac{\vec{P} - \vec{Q}}{|\vec{P} - \vec{Q}|^2} \tag{30}$$

$$\nabla_P \Xi = \frac{1}{|\vec{P} - \vec{Q}|^2}\hat{n} - 2\frac{\left(\vec{P} - \vec{Q}\right) \cdot \hat{n}(\vec{Q})}{|\vec{P} - \vec{Q}|^4}\left(\vec{P} - \vec{Q}\right) \tag{31}$$

2) APPROXIMATING THE BOUNDARY VALUE FUNCTION

To formulate the numerical solution of $\nabla_P(h(\vec{P}))$, we substitute the polynomial approximations of $H(\vec{Q})$ and $\nabla H_n(\vec{Q})$ functions (which are defined in (5) and (6), respectively) into (29). It is similar to obtaining the numerical solution of $h(\vec{P})$ in Section II-B. Then, the semi-analytical solution in (29) becomes the numerical solution in (47) in Appendix A.

3) REWRITING WITH VECTOR FORM

After obtaining the numerical solution of $\nabla_P(h(\vec{P}))$ in (47), we can use it to determine the value of the gradient function $\nabla_P(h(\vec{P}))$ with the the boundary value functions, namely $H(\vec{Q})$ and $\nabla H(\vec{Q}) \cdot \hat{n}(\vec{Q})$. We substitute the coefficients of the boundary value functions from the vector $[\boldsymbol{H}_j]$ and $[\boldsymbol{\nabla H}_{n,j}]$ in (26) into (47) and rewrite it in a vector form as

$$\psi \nabla_P(h(\vec{P})) = \left[\boldsymbol{G'}_{\nabla H,j}(\vec{P})\right]\left[\boldsymbol{\nabla H}_{n,j}\right] - \left[\boldsymbol{G'}_{H,j}(\vec{P})\right]\left[\boldsymbol{H}_j\right], \tag{32}$$

where $\boldsymbol{G'}_{H,j}(\vec{P})$ is $1 \times 2N$ geometry vector, which contains two integrals of shape functions: $F'_{\zeta_j}(\vec{P})$ and $F'_{\xi_j}(\vec{P})$. Both shape functions are of the point $\vec{P}$ locating between contour lines and of the point $\vec{Q}$ locating on C_j, and are defined as

$$[\boldsymbol{G'}_{H,j}(\vec{P})] = \left[* \cdots \quad F'_{\zeta_j}(\vec{P}) \quad F'_{\xi_j}(\vec{P}) \quad \cdots *\right]. \tag{33}$$

$$[\boldsymbol{F}_{\nabla H,ij}] = \begin{bmatrix} * \cdots & F_{\alpha_j}(-\frac{1}{2}) & F_{\beta_j}(-\frac{1}{2}) & F_{\gamma_j}(-\frac{1}{2}) & F_{\eta_j}(-\frac{1}{2}) & \cdots * \\ * \cdots & F_{\alpha_j}(-\frac{1}{4}) & F_{\beta_j}(-\frac{1}{4}) & F_{\gamma_j}(-\frac{1}{4}) & F_{\eta_j}(-\frac{1}{4}) & \cdots * \\ * \cdots & F_{\alpha_j}(0) & F_{\beta_j}(0) & F_{\gamma_j}(0) & F_{\eta_j}(0) & \cdots * \\ * \cdots & F_{\alpha_j}(\frac{1}{4}) & F_{\beta_j}(\frac{1}{4}) & F_{\gamma_j}(\frac{1}{4}) & F_{\eta_j}(\frac{1}{4}) & \cdots * \end{bmatrix}.$$

$[\boldsymbol{G'}_{\nabla H,j}(\vec{P})]$ is $1 \times 4N$ geometry vector, which contains four integrals of shape functions: $F'_{\alpha_j}(\vec{P})$, $F'_{\beta_j}(\vec{P})$, $F'_{\gamma_j}(\vec{P})$ and $F'_{\eta_j}(\vec{P})$. These four shape functions are of the point $\vec{P}$ locating between the contour lines and of the point $\vec{Q}$ locating on C_j, and are defined as

$$[\boldsymbol{G'}_{\nabla H,j}(\vec{P})] = \left[*\cdots\ F'_{\alpha_j}(\vec{P})\ F'_{\beta_j}(\vec{P})\ F'_{\gamma_j}(\vec{P})\ F'_{\eta_j}(\vec{P})\quad \cdots *\right]. \tag{34}$$

C. GRADIENT OF GRADIENT OF INTERPOLATED ELEVATION FUNCTION OVER REGION BETWEEN DEM CONTOUR LINES

In this paper, a second-order series expansion will be used to determine the direction of the SWP over any point in the catchment basin with the proposed algorithm in Section V. The first derivative at any point $\vec{P}$ was found in Section IV-B. This section finds the gradient of the gradient of interpolated elevation function, which is a tensor and can be written as

$$\nabla_P^2 h = \begin{pmatrix}\hat{a}_x & \hat{a}_y\end{pmatrix}\begin{pmatrix}h_{xx} & h_{xy}\\ h_{yx} & h_{yy}\end{pmatrix}\begin{pmatrix}\hat{a}_x\\ \hat{a}_y\end{pmatrix}, \tag{35}$$

where $h_{xx} = \partial h_x/\partial x, h_{xy} = \partial h_x/\partial y, h_{yx} = \partial h_y/\partial x$ and $h_{yy} = \partial h_y/\partial y$. The vector function $\nabla_P h(\vec{P})$ is reproduced from the proposed semi-analytical solution as shown in (29) in Section IV-B1. The process to calculate the gradient of gradient of the elevation function $h(\vec{P})$ at the point between the DEM contour lines is structured as follows.

1) FORMULATING GRADIENT OF GRADIENT OF PROPOSED SOLUTION

From (29) in Section IV-B1, we formulate the gradient of $\nabla_P(h(\vec{P}))$ as

$$\begin{aligned}\psi\left[\nabla_P^2 h(\vec{P})\right] &= -\sum_{j=1}^{N}\int_{C_j}[\nabla_P\Lambda]\,H(\vec{Q})\,ds_Q\\ &\quad+\sum_{j=1}^{N}\int_{C_j}\left[\nabla_P\Psi\left(\vec{P},\vec{Q}\right)\right]\left(\nabla H(\vec{Q})\cdot\hat{n}(\vec{Q})\right)ds_Q\end{aligned} \tag{36}$$

where $\nabla_P\Lambda$ and $\nabla_P\Psi$ are gradient terms, and can be calculated analytically as follow.

$$\begin{aligned}\nabla_P\Lambda &= \frac{-2}{|\vec{P}-\vec{Q}|^4}\hat{n}\left(\vec{P}-\vec{Q}\right)+\frac{-2\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^4}\boldsymbol{I}\\ &\quad+\frac{-2\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^5}\left(\vec{P}-\vec{Q}\right)\hat{n}\\ &\quad+\frac{8\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^6}\left(\vec{P}-\vec{Q}\right)^2\end{aligned} \tag{37}$$

$$\nabla_P\Psi = \frac{\boldsymbol{I}-2\left(\vec{P}-\vec{Q}\right)^2}{|\vec{P}-\vec{Q}|^2}, \tag{38}$$

and $\boldsymbol{I}$ is an identity tensor, which can be defined as

$$\boldsymbol{I} = \begin{pmatrix}\hat{a}_x & \hat{a}_y\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\hat{a}_x\\ \hat{a}_y\end{pmatrix}, \tag{39}$$

and $\left(\vec{P}-\vec{Q}\right)^2$, $\hat{n}\left(\vec{P}-\vec{Q}\right)$ and $\left(\vec{P}-\vec{Q}\right)\hat{n}$ are also tensors, where

$$\begin{aligned}\vec{R} &= \begin{pmatrix}\hat{a}_x & \hat{a}_y\end{pmatrix}\begin{pmatrix}R_x\\ R_y\end{pmatrix}\\ \vec{R}^2 &= \begin{pmatrix}\hat{a}_x & \hat{a}_y\end{pmatrix}\begin{pmatrix}R_x^2 & R_xR_y\\ R_xR_y & R_y^2\end{pmatrix}\begin{pmatrix}\hat{a}_x\\ \hat{a}_y\end{pmatrix},\\ \hat{n}\vec{R} &= \begin{pmatrix}\hat{a}_x & \hat{a}_y\end{pmatrix}\begin{pmatrix}n_xR_x & n_xR_y\\ n_yR_x & n_yR_y\end{pmatrix}\begin{pmatrix}\hat{a}_x\\ \hat{a}_y\end{pmatrix},\\ \vec{R}\hat{n} &= \begin{pmatrix}\hat{a}_x & \hat{a}_y\end{pmatrix}\begin{pmatrix}R_xn_x & R_yn_x\\ R_yn_x & R_yn_y\end{pmatrix}\begin{pmatrix}\hat{a}_x\\ \hat{a}_y\end{pmatrix}.\end{aligned} \tag{40}$$

2) APPROXIMATING THE BOUNDARY VALUE FUNCTION

To formulate the numerical solution of $\nabla_P^2(h(\vec{P}))$, we substitute the polynomial approximations of $H(\vec{Q})$ and $\nabla H_n(\vec{Q})$ functions (which are defined in (5) and (6)) into (36). Similar to obtaining the numerical solution of $h(\vec{P})$ in Section II-B, the semi-analytical solution in (36) becomes the numerical solution in (54) in Appendix B.

3) REWRITING WITH VECTOR FORM

After obtaining the numerical solution of $\nabla_P^2(h(\vec{P}))$ in (54), we can use it to determine the value of the gradient of gradient function $\nabla_P^2(h(\vec{P}))$ with the the boundary value functions, namely $H(\vec{Q})$ and $\nabla H(\vec{Q})\cdot\hat{n}(\vec{Q})$. We substitute the coefficients of the boundary value functions from the vector $[\boldsymbol{H}_j]$ and $[\boldsymbol{\nabla H}_{n,j}]$ in (26) into (54) and rewrite it in a vector form as

$$\begin{aligned}&\psi\,\nabla_P^2(h(\vec{P}))\\ &\quad=\left[\boldsymbol{G''}_{\nabla H,j}(\vec{P})\right]\left[\boldsymbol{\nabla H}_{n,j}\right]-\left[\boldsymbol{G''}_{H,j}(\vec{P})\right]\left[\boldsymbol{H}_j\right],\end{aligned} \tag{41}$$

where $\boldsymbol{G''}_{H,j}(\vec{P})$ is $1 \times 2N$ geometry vector, which contains two integrals of shape functions: $F''_{\zeta_j}(\vec{P})$ and $F''_{\xi_j}(\vec{P})$. Both shape functions are of the point $\vec{P}$ locating between contour lines and of the point $\vec{Q}$ locating on C_j, respectively. They are defined as

$$[\boldsymbol{G''}_{\nabla H,j}(\vec{P})] = \left[*\cdots\ F''_{\zeta_j}(\vec{P})\ F''_{\xi_j}(\vec{P})\ \cdots *\right] \tag{42}$$

and $[\boldsymbol{G''}_{\nabla H,j}(\vec{P})]$ is $1 \times 4N$ geometry vector, which contains four integrals of shape functions: $F''_{\alpha_j}(\vec{P})$, $F''_{\beta_j}(\vec{P})$, $F''_{\gamma_j}(\vec{P})$ and $F''_{\eta_j}(\vec{P})$. These shape functions are of the point $\vec{P}$ locating between contour lines and the point $\vec{Q}$ locating on the C_j. They are defined as

$$[\boldsymbol{G''}_{\nabla H,j}(\vec{P})] = \left[*\cdots\ F''_{\alpha_j}(\vec{P})\ F''_{\beta_j}(\vec{P})\ F''_{\gamma_j}(\vec{P})\ F''_{\eta_j}(\vec{P})\cdots *\right] \tag{43}$$

V. DELINEATION OF SURFACE WATER PATH

This section proposes an algorithm to delineate the SWP over any point in the catchment basin. Unlike other algorithms, the proposed algorithm can estimate the size of a sub-catchment area, which is only a part of the whole area, because it can generate the SWP without interpolation over all points between the contour lines. This feature helps us reduce computational time because generating the SWP requires only the information in the area that the SWP is passing. The proposed algorithm comprises three main steps. The first step is to prepare the information to predict an endpoint of the SWP segment after identifying the starting point on the sub-catchment area. The starting point information comprises the elevation and its gradients at that point. It can be calculated with the numerical solution in Section IV. In the second step, we predict the endpoint of the SWP segment based on the approximated surface in polar coordinates. We solve a second-order series expansion to approximate the elevation of a neighborhood of the starting point by using the numerical results obtained from the first step. The third step is to predict the next SWP segment. We use the endpoint of the SWP segment as the starting point of the next SWP segment. Therefore, this section presents the second and third steps of the algorithm.

A. DETERMINATION OF THE SWP SEGMENT

Before delineating the whole SWP from an upper contour line to a lower contour line, we must determine the location of each SWP segment. We choose a point between contour lines as the starting point of the first segment and choose the endpoint from a neighborhood of the starting point where the endpoint has the maximum or minimum elevation. In doing that, we approximate the elevation over the neighborhood of the starting point by using the surface in polar coordinates.

1) CREATING SURFACE OVER THE NEIGHBORHOOD OF THE STARTING POINT

In Section II, we use the solution of Laplace's equation to interpolate the elevation at any point along the SWP. Here, we still solve its solution in a polar coordinate system, and apply that solution for creating surface over the neighborhood of the starting point of the SWP segment. Laplace's equation in a two-dimensional polar coordinate system is written as

$$\nabla^2 h(r,\theta) = \frac{\partial^2 h}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 h}{\partial \theta^2} + \frac{1}{r}\frac{\partial h}{\partial r} = 0, \tag{44}$$

where r is the distance from the starting point of the SWP segment to the endpoint, and θ is between 0 and 360 degrees. We can obtain its solution in general form by using the separation of variables. Then, the solution in a particular form is obtained by fitting this application and can be written as

$$h(r,\theta) = a_0 + \sum_{m=1}^{\infty} r^m (a_m \cos m\theta + b_m \sin m\theta). \tag{45}$$

Even though the series expansion in (45) is exact, it is an infinite series. For computation efficiency, we substitute a_m and b_m for the finite number of terms. In doing this, we keep the expansion area small ($r < r_0$) to limit the approximation error, where r_0 denotes the length of the SWP segment. Hence, SWP is a concatenation of several short SWP segments, each of which has its own series expansion. At the beginning, the surface over the neighborhood of the starting point of SWP segment in (45) is approximated as (71) in Appendix C. The direction over this surface will be chosen to delineate the first segment of SWP in the next step.

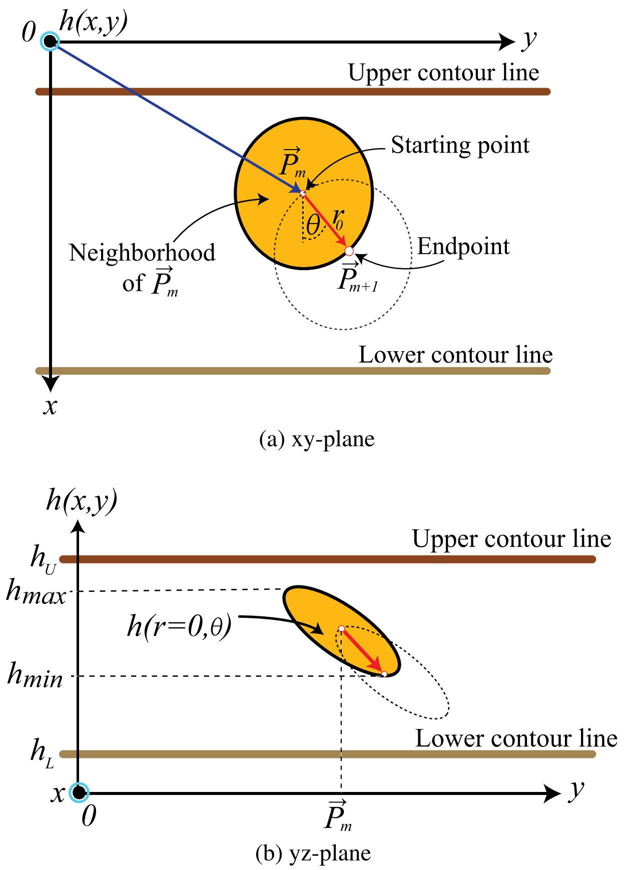


FIGURE 4. Determining the direction of the SWP segment with the approximated surface in polar coordinates $h(r,\theta)$ for $r < r_0$.

2) DETERMINING SWP DIRECTION

The SWP direction is emanating from the starting point of the SWP segment and toward the endpoint of that segment or the starting point of the next SWP segment. Given that the small disc around the starting point is contained in the region between the contour lines, we select the endpoint of the current SWP segment from the neighborhood of the starting point. Otherwise, no endpoint is selected, and SWP delineation finishes. When setting the length of the SWP segment as r_0, we consider the elevation at any point on the boundary of the disc from the approximated surface in the polar coordinates $h(r,\theta)$, which can be determined in (71).

Fig. 4 shows an example of approximated surface in the polar coordinates $h(r,\theta)$ for $r < r_0$. We select the path, which starts from the point $\vec{P}$ toward the endpoint that gives steepest descent. To do that, we find the endpoint, which minimizes $h(r, \theta)$ on the circle with $r = r_0$. In fact, another approach is to select the path, which starts from the point $\vec{P}$ toward the endpoint with the steepest ascent. To do that, we find the endpoint, which maximizes $h(r, \theta)$ on the circle with $r = r_0$. In this paper, we use the steepest descent approach. Therefore, we identify the endpoint that minimizes $h(r, \theta)$ by finding $\theta_{\min}$, where $dh(r, \theta)/d\theta = 0$ on the circle $r = r_0$, as shown in Fig. 4 (b). An example of $\theta_{\min}$ finding with standard technique is given in Appendix D.

B. PROPOSED ALGORITHM

In this section, we propose a recursive algorithm for delineating an SWP, which is the concatenation of several SWP segments over a catchment area, in which the elevation is represented with contour lines in a DEM image. The algorithm comprises five phases as follows.

1) ALLOCATING THE MEMORY ON A COMPUTER

To delineate the SWP over a large catchment area, the algorithm should efficiently use computer memory to collect the contour line data and the position of the delineated SWPs. The memory management must allocate portions of memory and free them for reuse when no longer needed. The input data comprises the coordinates of the contour lines and a list of zoning. To apply the contour lines for being the boundary condition in the proposed algorithm, we regularly sample some points along the original contour lines obtained from a DTM and reconstruct the polylines (chain of line segments) as the input data. The number of segments depends on the sampling rate. A list of zoning is information describing the boundary of each sub-area. Fig. 5 presents the elevation over a catchment area, which is represented with N contour lines and K zones. The zone z_1 is bounded by the C_1 and C_2, and the zone z_K is bounded by the C_{N-1} and C_N. The algorithm collects each coordinate, denoted as $\vec{Q}_j^{C_n}$, of the contour line C_n. The output data comprises the boundary values along the contour lines C_n, namely the elevation functions and the gradient of elevation function, denoted by $H(\vec{Q}_j^{C_n})$ and $\nabla H(\vec{Q}_j^{C_n})$, respectively. These output data will be used to interpolate the elevation at the points, which the SWP is passing.

2) ITERATIVELY ASSIGNING SWP SEGMENTS

Fig. 4 presents the starting point and the endpoint of an SWP segment, indexed by m^{th}. The endpoint of the m^{th} segment is the starting point of the $(m+1)^{th}$ segment. This point is determined from the starting point of the m^{th} segment as explained in Section V-A. The starting point and the endpoint of the m^{th} segment are denoted by $\vec{P}_m$ and $\vec{P}_{m+1}$, respectively. The starting point of the 1^{st} segment $\vec{P}_1$ is arbitrarily designated from any point in the region lying within two adjacent contour lines (or any zone; e.g., z_1, z_2, …, or z_K). Notice that the 1^{st} segment starts from the upper contour line, and the last segment ends at the lower contour line.

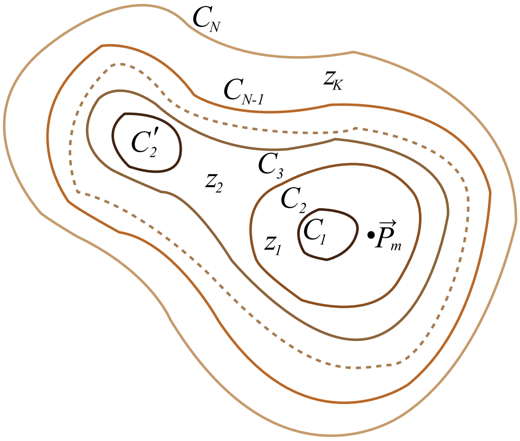


FIGURE 5. *K* Zones and *N* Contour lines.

3) PREPARING INFORMATION AT $\vec{P}_m$

Before the algorithm computes the endpoint of the current SWP segment, it requires the elevation and its derivatives at point $\vec{P}_m$. In this phase, hence, the algorithm computes $h(\vec{P}_m)$ and its derivatives $(\nabla h(\vec{P}_m), \nabla^2 h(\vec{P}_m))$ with the numerical solutions which are explained in Section IV. These computations require a part of the input data collected in the memory. For example, in Fig. 5, the point $\vec{P}_m$ is in the zone z_1, so the algorithm requires only the inputf data involving the contour lines C_1 and C_2. The required input data comprise the coordinates of points, i.e., the $\vec{Q}_j^{C_1}$ and $\vec{Q}_j^{C_2}$, as well as the elevation and gradient functions along the contour lines, i.e., $H(\vec{Q}_j^{C_1})$, $H(\vec{Q}_j^{C_2})$, $\nabla H(\vec{Q}_j^{C_1})$ and $\nabla H(\vec{Q}_j^{C_2})$. The calculations of both $H(\vec{Q}_j^{C_n})$ and $\nabla H(\vec{Q}_j^{C_n})$ are explained in Section III.

4) DETERMINING DIRECTION OF THE SWP SEGMENT

After the elevation and its derivatives at point $\vec{P}_m$, namely $h(\vec{P}_m)$, $\nabla h(\vec{P}_m)$ and $\nabla^2 h(\vec{P}_m)$ were obtained, this phase employs those calculated results for determining $\theta_{\min}$. The calculation of $\theta_{\min}$ is explained in Section V-A. Then, the algorithm uses the $\theta_{\min}$ to determine the endpoint of current SWP segment. Note that the length of the SWP segment (r_0) can be adjusted, and this will give a different endpoint.

5) DETERMINING THE STARTING POINT OF THE NEXT SWP SEGMENT

In the last phase, the algorithm collects the coordinates of the endpoint of the current SWP segment into the memory and starts to delineate the next SWP segment by using the endpoint of the current SWP segment as the starting point $\vec{P}_{m+1}$. The algorithm repeats from the second phase until the endpoint of the SWP segment is not located in the zone z_N.

All five phases can be written as pseudocodes as shown in Algorithm 1 and Algorithm 2, respectively.

To summarize, the computation is initiated by determining the endpoint of the first SWP segment from the assigned point with Algorithm 1. Then, Algorithm 1 is looped until

Algorithm 1 An Algorithm to Delineate SWP

```
Input: list of the zone's names in this catchment, list of
    the contour line's names in each zone, list of the $\vec{Q}$
    point's coordinates along the contour lines, coordinates
    of the assigned point: $\vec{P}_a$
Output: [$\vec{P}_m$] list of coordinates of the SWP from the
    assigned point: $\vec{P}_a$ to the ending point of the last
    SWP segment: $\vec{P}_{m+1}$
    Initialisation : $\vec{P}_m \leftarrow \vec{P}_0$; $z_k \leftarrow$ Zoning($\vec{P}_m$);
    $z_{\text{Previous}} \leftarrow -1$; FINISH $\leftarrow 0$ ;
    LOOP Process
While (FINISH==0) do
if $z_k \neq z_{\text{Previous}}$ then
    compute [$\boldsymbol{H}$] with (21);
    compute [$\boldsymbol{\nabla H}_n$] with (26);
end if
compute $h(\vec{P}_m)$ with (27);
compute $\nabla h(\vec{P}_m)$ with (32);
compute $\nabla^2 h(\vec{P}_m)$ with (41);
$z_{\text{Previous}} \leftarrow z_k$
$\vec{P}_{m+1} \leftarrow$ Next_SWP($\vec{P}_m$) in Algorithm 2;
$z_k \leftarrow$ Zoning($\vec{P}_{m+1}$);
if $z_k \in$ CATCHMENT then
    $\vec{P}_m \leftarrow \vec{P}_{m+1}$;
    FINISH $\leftarrow 0$;
else
    FINISH $\leftarrow 1$;
end if
End While
```

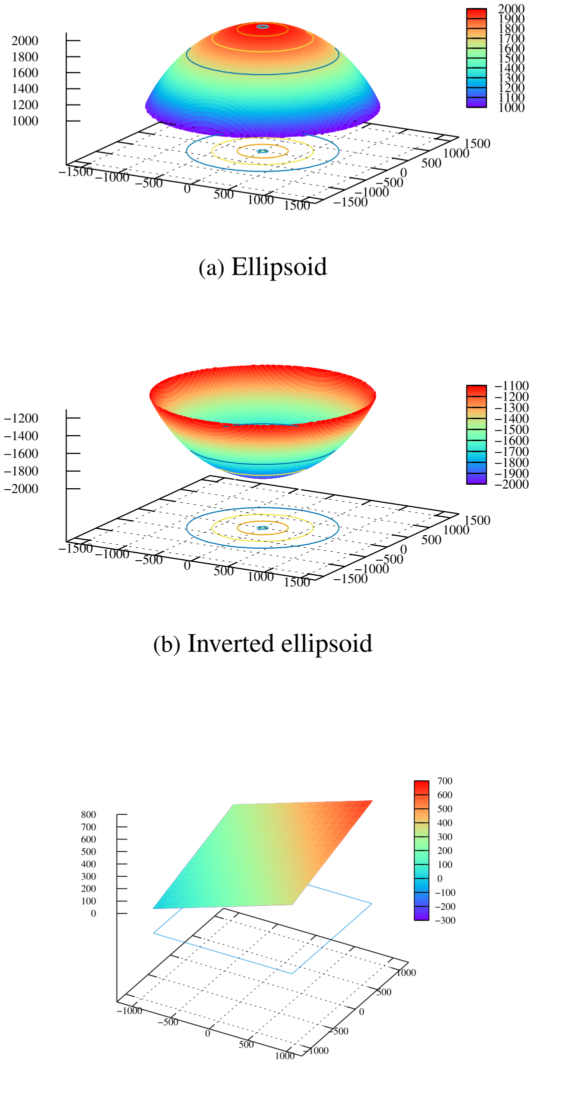
Algorithm 2 Next_SWP($\vec{P}_m$)

```
Input: $\vec{P}_m$, $h(\vec{P})$, $\nabla_P h(\vec{P})$, $\nabla^2_P h(\vec{P})$, $r_0$
Output: $\vec{P}_{m+1}$
$h_x \leftarrow$ Find $\nabla h(\vec{P})$ in $x$-component;
$h_y \leftarrow$ Find $\nabla h(\vec{P})$ in $y$-component;
$h_{xx} \leftarrow$ Find $\nabla^2 h(\vec{P})$ in $xx$-component;
$h_{xy} \leftarrow$ Find $\nabla^2 h(\vec{P})$ in $xy$-component;
Solve root:{ $\theta_{\min}, \theta_{\max}$ } from (76);
$x\_\vec{P}_{m+1} \leftarrow x\_\vec{P}_m + r_0 \cos\theta_{\min}$;
$y\_\vec{P}_{m+1} \leftarrow y\_\vec{P}_m + r_0 \sin\theta_{\min}$;
```

the endpoint of the SWP segment locates outside the catchment area. Finally, each loop calls Function Next_SWP in Algorithm 2 to determine the endpoint of the current SWP segment, which is the starting point of the next SWP segment.

VI. RESULTS AND DISCUSSIONS

This section demonstrates how to delineate the SWP over the synthetic and natural surfaces with the proposed algorithm, and presents the validation of the extracted SWP. First, we present the performance of the proposed algorithm when there is no incomplete contour line. The comparisons are made with the D8 algorithm [4] since the implementation of the existing contour-based algorithm is not affordable in distributed hydrological model [42]. Next, we analyze the position accuracy of the delineated SWP by using the elevation data derived from the formulas of standard synthetic surfaces, namely ellipsoid surface (hill), inverted ellipsoid[1] (pit),



(a) Ellipsoid

(b) Inverted ellipsoid

(c) Inclined plane

FIGURE 6. Spatial distributions of elevations.

TABLE 1. Formulas of synthetic surfaces and equations of theoretical SWPs.

	Formulas of synthetic surfaces	Equations of theoretical SWPs
Ellipsoid	$\frac{x^2+y^2}{(1600)^2} + \frac{z^2}{(2000)^2}=1\ (z>0)$	$y = kx$
Inverted	$\frac{x^2+y^2}{(1600)^2} + \frac{z^2}{(2000)^2}=1\ (z<0)$	$y = kx$
Inclined Plane	$z = 2x + 1.5y + 100$	$y = \frac{1.5}{2}x + k$

Note: k in the equation of theoretical SWP is obtained by substituting point coordinates into the equation.

[1] *Inverted ellipsoid* is referred to as *inverse ellipsoid* in [4].

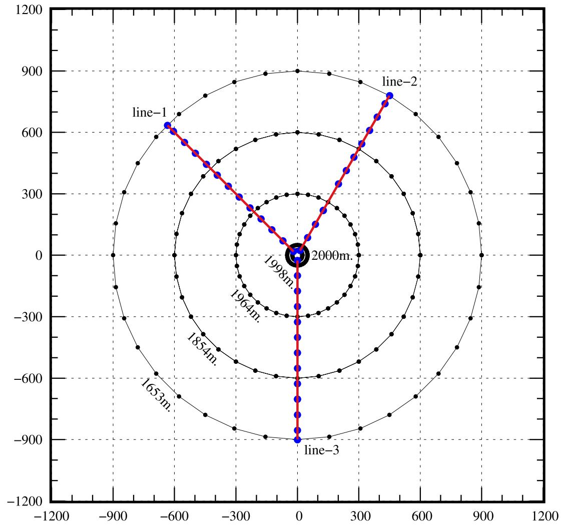


FIGURE 7. Ellipsoid: Positions of sampled points along contour lines and positions of the SWPs including line 1, line 2 and line 3. The sampled points are plotted by the black points. The theoretical and extracted SWPs are plotted by the red lines and the blue triangular points.

and inclined plane (flat land). Then, the case that there are incomplete contour lines is considered. We use the elevation of natural terrain based on a DEM image, which comprises the hill's features, pit, and flat land. The SWPs delineated from the incomplete contour lines are compared with the SWPs delineated from complete contour lines to evaluate the effectiveness of the proposed algorithm.

A. SYNTHETIC SURFACE

Reference [4] presented the method for measuring the position error of extracted SWPs to systematically evaluate the accuracy of various flow direction algorithms, namely the D8 algorithm and its extensions. The position error is the difference between the extracted SWP position and the theoretical (true) SWP position derived from the formulas of the standard synthetic surface. Furthermore, the results in [4] reveal that the D8 algorithm and its extensions have non-negligible error floors. Therefore, it is of interest to use this evaluation method to also measure the position error of our proposed algorithm, so that the comparisons can be made.

1) THEORETICAL SWPS

We use the theoretical SWP positions as the references for measuring the error of the extracted SWPs. The theoretical SWP is the closed-form solution which is derived from the formulas of the synthetic surface mentioned in [4]. Table 1 lists the formulas of three synthetic surfaces and the equations of theoretical SWPs. The 3-D graphics of the synthetic surfaces are illustrated in Fig. 6.

The positions of theoretical SWPs in three cases, namely ellipsoid, inverted ellipsoid, and inclined plane, are plotted with red lines in Fig. 7, Fig. 8 and Fig. 9, respectively. First, the ellipsoid that models a hill in Fig. 6 (a) gives the theoretical SWP, radiating out from the top of the ellipsoid as shown in Fig. 7. Second, the inverted ellipsoid that models a pit in Fig. 6 (b) gives the theoretical SWP, sinking from the border of the inverted ellipsoid to its center as shown in Fig. 8. Last, the inclined plane that models flat plane in Fig. 6 (c) gives the theoretical SWP, flowing in parallel with the direction in Fig. 9.

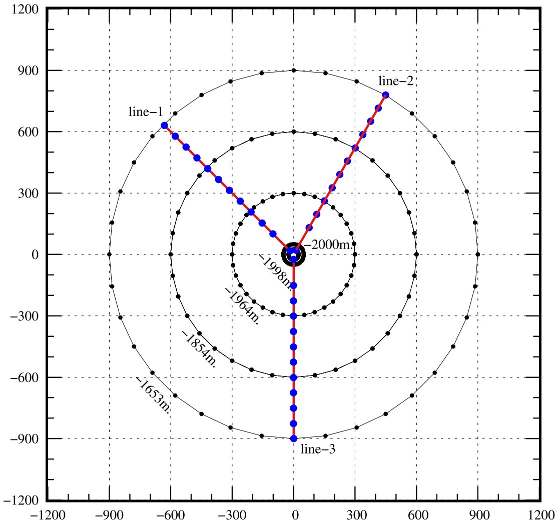


FIGURE 8. Inverted Ellipsoid: Positions of sampled points along contour lines and positions of the SWPs including line 1, line 2 and line 3. The sampled points are plotted by the black points. The theoretical and extracted SWPs are plotted by the red lines and the blue triangular points.

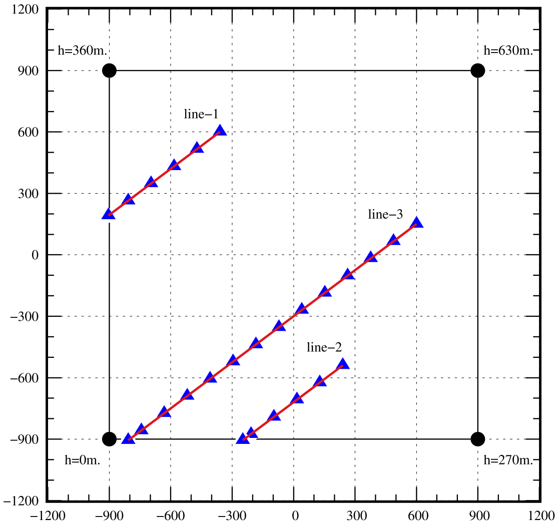


FIGURE 9. Inclined Plane: Positions of sampled points along contour lines and positions of the SWPs including line 1, line 2 and line 3. The sampled points are plotted by the black points. The theoretical and extracted SWPs are plotted by the red lines and the blue triangular points.

2) EXTRACTED SWPS OVER ELLIPSOID

This work uses the formulas of the ellipsoid listed in Table 1 to generate the contour lines which are in a circle shape. The ellipsoid comprises five contour lines, namely 2000, 1998, 1964, 1854, and 1653 meters. Along a contour line, we select a point at every θ_s degrees. For a fair comparison, θ_s is set at 10 degrees, so that the distance between adjacent points is not shorter than that of D8 resolution (4.5 × 4.5 meters). Fig. 7 presents the sampled points (in black) which are generated with $\theta_s = 10$. Totally, a boundary C in the numerical solution in (3) consists of $5 \times (360/\theta_s)$ intervals (C_j).

The proposed algorithm delineates an extracted SWP over a catchment area by dividing the catchment area into four zones, where five contour lines with their elevations serve as the boundary. Along the boundary, the algorithm needs a memory size of $5 \times (360/\theta_s)$ for keeping the calculated coordinates of $\vec{Q}$ points. We set the length of an SWP segment as $r_0 = 1.51\ m.$, and locate the starting point $\vec{P}_1$ in the zone z_1, which is bounded by the contour lines with $h = 2000$m and $h = 1998$m. To apply the method of [4], we put the starting points very close to the position, used to evaluate the error of the D8 algorithm. The distance between the starting points and the ellipsoid center is set at 30 meters, so that the coordinate of the starting points is close to the corner point of a DEM pixel. We select three starting points along a circle with a radius of 900 meters to delineate three extracted SWPs in the different directions, namely 135, 60, and 270 degrees, where the extracted SWPs are labeled by "Line-1", "Line-2", and "Line-3", respectively and are plotted with blue markers in Fig. 7.

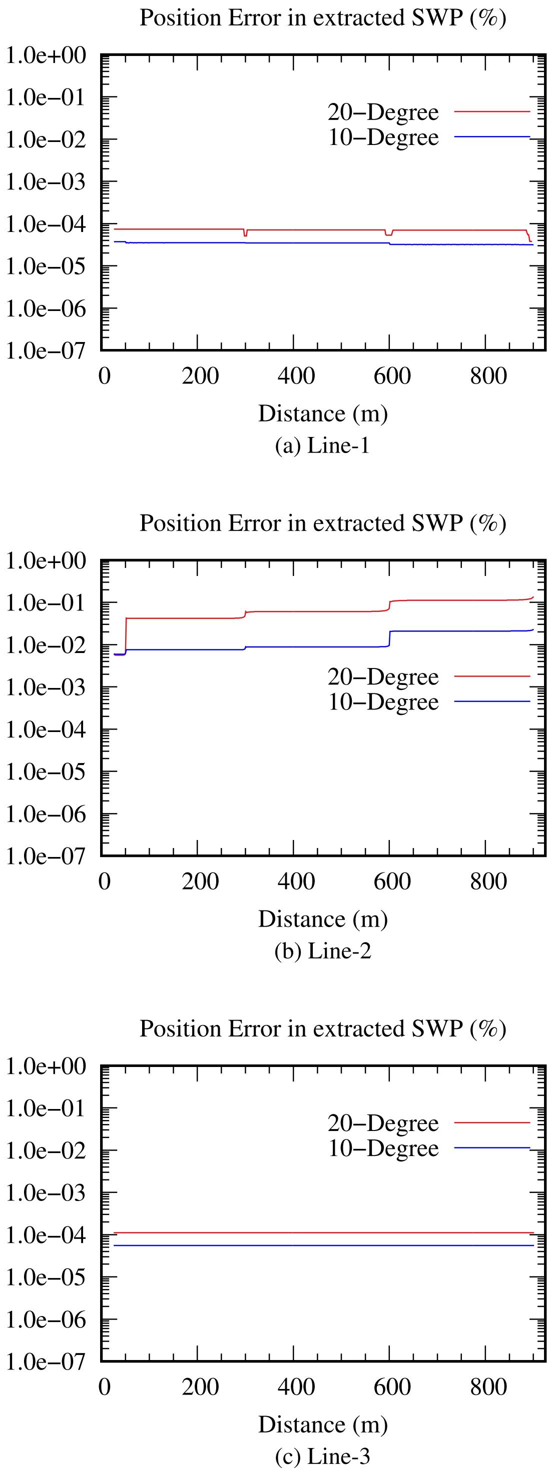
To evaluate the accuracy of three extracted SWPs, we determine the position error of the extracted SWP by using the formula

$$E(\%) = \frac{|\theta^m_{Cal} - \theta_{\text{Exact}}|}{\theta_{\text{Exact}}} \times 100, \tag{46}$$

where θ_{Exact} is the constant angle of the theoretical SWP, and θ^m_{Cal} is the angle of the extracted SWP which is converted from the coordinates of the ending point of the m^{th} segment of extracted SWP. First, the positions of the "Line-1" extracted SWP in Fig. 7 are compared with $\theta_{\text{Exact}} = 135$ degrees, and the position error $E(\%)$ versus the distance of the SWP flowing from the ellipsoid center are plotted in Fig. 10 (a). Obviously, the value of $E(\%)$ is below 0.001 at all distances. Second, the positions of the "Line-2" extracted SWP are compared with $\theta_{\text{Exact}} = 60$ degrees, and the position error $E(\%)$ versus the distance of the SWP flowing from the ellipsoid center are plotted in Fig. 10 (b). The value of $E(\%)$ is below 0.1 at all distances. In case that θ_s is increased to 20 degrees, the value of $E(\%)$ is below 0.2 at all distances. Last, the positions of the "Line-3" extracted SWP are compared with $\theta_{\text{Exact}} = 270$ degrees, and the position error $E(\%)$ versus the distance of the SWP flowing from the ellipsoid center are plotted in Fig. 10 (c). The value of $E(\%)$ is below 0.0002 at all distances.

3) EXTRACTED SWPS OVER INVERTED ELLIPSOID

This work uses the formulas of the inverted ellipsoid listed in Table 1 to generate the contour lines, which are in a circle shape. The inverted ellipsoid comprises five contour lines, namely -2000, -1998, -1964, -1854 and -1653 meters, respectively. Notice that the coordinates of contour lines are the same as those in ellipsoid, but their elevations are negative. They are plotted in Fig. 8. Totally, a boundary C in the numerical solution in (3) consists of $5 \times (360/\theta_s)$ intervals (C_j).

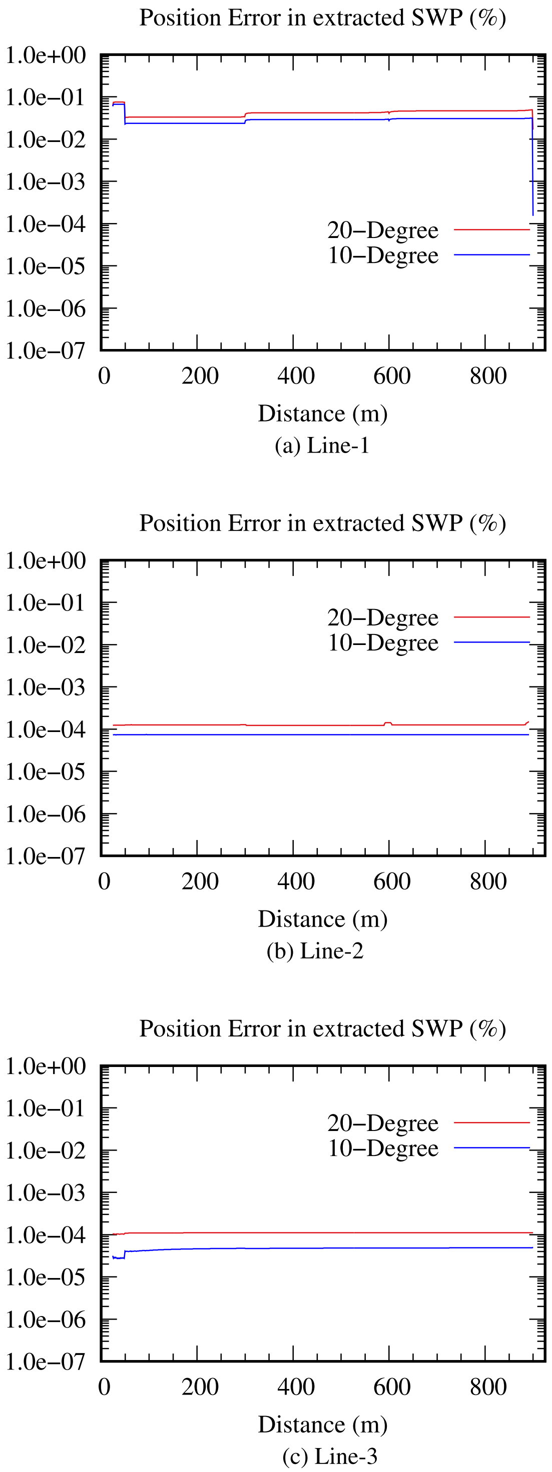


(a) Line-1

(b) Line-2

(c) Line-3

FIGURE 10. Position error *E*(%) versus the distance of the SWP flowing from the ellipsoid center.



(a) Line-1

(b) Line-2

(c) Line-3

FIGURE 11. Position error $E(\%)$ versus the distance of the SWP flowing from the border of inverted ellipsoid to the center.

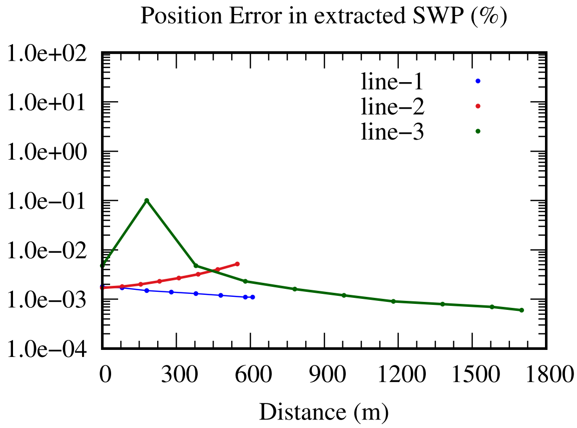


FIGURE 12. Position error $E(\%)$ versus the distance of the SWP flowing over a inclined plane.

Since we generate the coordinates of the contour lines with the same sampling rate, used in ellipsoid, an equal amount of memory is required. We set the length of an SWP segment at $r_0 = 1.51$m and locate the starting point $\vec{P}_1$ in zone z_4, which is bounded by the contour lines $h = -1653$m and $h = -1854$m, so that the starting point is very close to the position, used to evaluate the error of D8 algorithm [4]. The distance between the starting points and the inverted ellipsoid center is set at 900 meters. We select three starting points along a circle with a radius of 900 meters to delineate three extracted SWPs in three directions, namely 135, 60, and 270 degrees, where the extracted SWPs are labeled by "Line-1", "Line-2", and "Line-3", respectively and are plotted with blue markers in Fig. 8.

Fig. 11 presents the position error $E(\%)$ versus the distance of the SWP flowing from the border of inverted ellipsoid to the center. The position error $E(\%)$ is calculated with (46). First, the positions of the "Line-1" extracted SWP are compared with $\theta_{\text{Exact}} = 135$ degrees, and their error $E(\%)$ are below 0.1 and are plotted in Fig. 11 (a). Also, the positions of the "Line-2" and "Line-3" extracted SWPs are compared with $\theta_{\text{Exact}} = 60$ and $\theta_{\text{Exact}} = 270$ degrees, respectively, and their error $E(\%)$ are below 0.0002 and 0.00015, and are plotted in Fig. 11 (b) and (c), respectively.

4) EXTRACTED SWPS OVER INCLINED PLANE

Unlike ellipsoid and inverted ellipsoid, the formula of the inclined plane listed in Table 1 is used to generate only one incomplete contour line which is in a square shape with the dimension 900m × 900m. The four corner points along the incomplete contour have different elevation levels, namely 0, 270, 630 and 360 meters, respectively. The sampled points are plotted in Fig. 9. Then, a boundary C in the numerical solution in (3) consists of 4 intervals (C_j).

After collecting the information of the boundary into the memory, we set the length of an SWP segment at $r_0 = 20.0$m. To apply the method of [4], we put the starting points very close to the position, used to evaluate the error of the D8 algorithm. Hence, we put the starting points at the coordinates $(-360, 600)$, $(240, -540)$ and $(600, 150)$ to delineate the "Line-1", "Line-2" and "Line-3" extracted SWPs in the same direction, i.e. 36.855 degrees. We plot three extracted SWPs with blue triangular markers in Fig. 9.

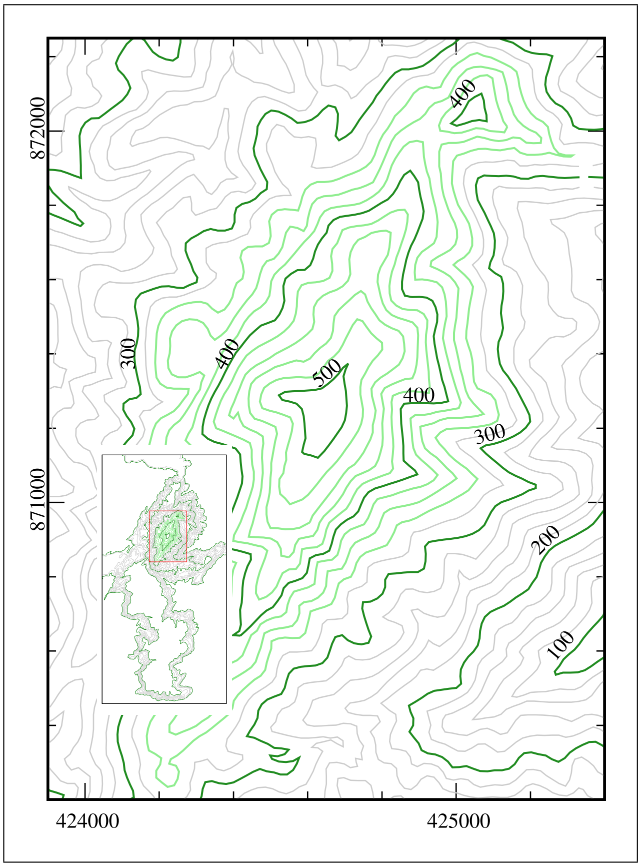


FIGURE 13. Contour lines for calculation.

TABLE 2. Comparison of average SWP position error (%) of the delineated SWPs.

Algorithms	Resolution of input data	Ellipsoid	Inverted Ellipsoid	Plane
Proposed:				
Number of	90	0.120	0.048	-
sampled points	180	0.025	0.032	-
over all contours	4	-	-	0.001
Others:				
D8	Number of	80.1	161.4	115.4
Rho-8	grid cells	69.5	139.8	90.7
D8-LTD	(4.5x4.5 m^2)	47.5	100.7	48.6
FDFM	is 160,000.	76.0	142.1	101.1
FMD-md		56.0	96.1	84.4
D∞		55.9	88.8	75.4

Note: The error of the other algorithms is found in Table 2 of [4].

Fig. 12 presents the position error $E(\%)$ versus the distance of the SWP flowing over the inclined plane. The position error $E(\%)$ is calculated with (46). The positions of the "Line-1", "Line-2" and "Line-3" extracted SWPs are compared with $\theta_{\text{Exact}} = 36.855$ degrees. The errors $E(\%)$ of all extracted SWPs are below 0.0002.

5) COMPARISON TO OTHER ALGORITHMS

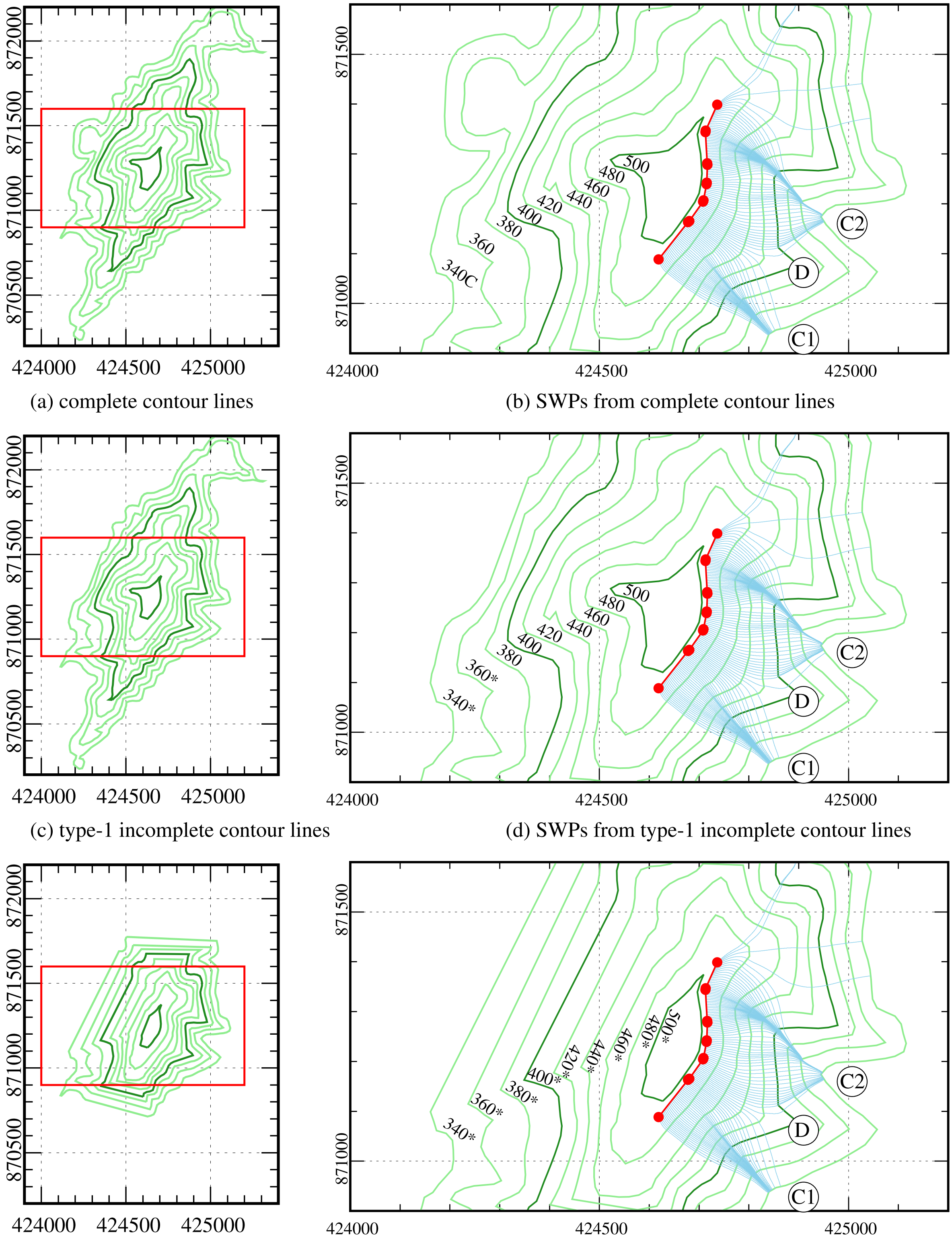
In previous section, the position errors of the delineated SWPs compared to the theoretical SWPs are calculated by using the evaluation method in [4], and found that each surface shape requires different resolution to achieve small position errors. For example, an ellipsoid and an inverted ellipsoid require higher resolution than a inclined plane. In this section, we present the accuracy comparison between the proposed algorithm and existing algorithms, including D8, Rho-8, D8-LTD, FDFM, FMD-md and D∞. For fair comparison, the proposed algorithm is tested with the same set of surface shapes available in [4], namely, ellipsoid, inverted ellipsoid and plane. Table 2 shows the average of position errors of the proposed algorithm in Fig. 10 to Fig. 12 and the average of position errors of the existing algorithms.

The proposed algorithm significantly outperforms all existing algorithms. In the ellipsoid case, the average position error of the proposed algorithm is 0.120% whereas the best existing algorithm (D8-LTD) obtains 47.5%. In the inverted ellipsoid case, the average position error from the proposed algorithm is 0.120% whereas the best existing algorithm (D∞) obtains 88.8%. In the plane, the average position error from the proposed algorithm is 0.001% whereas the best existing algorithm (D8-LTD) obtains 48.6%. Furthermore, while increasing the resolution of input data from 90 to 180 does not improve the existing algorithms [4], it helps the proposed algorithm achieve much smaller average position error in case of ellipsoid and inverted ellipsoid.

B. NATURAL TERRAIN

This work considers the real terrain, referred to as the Bang Wad area (Phuket, Thailand). This area often lacks the water supply for the tourism demand. Therefore the water management system must be improved. The center of the Bang Wad has latitude 7.88708°N and longitude, 98.33433°E. In this area, the geologic substratum is constituted of igneous rock. The topography can be described as fairly complex as a result of surface water erosion and active landslide. To make a quantitative analysis of the extracted SWPs on real terrains, a 30 × 30 DEM matrix is chosen and is downloaded from the Earth Science Data Systems (ESDS) (https://earthdata.nasa.gov/learn/articles/new-aster-gdem) with a horizontal resolution of 30 meters and a vertical resolution of 1 meter. The elevation ranges from 515.0 to 45.0 meter amsl (above mean sea level). Fig. 13 presents a considered area of 5000 × 10000 m^2 in which elevations are illustrated with contour lines from 500.0 to 100.0 meter amsl (above mean sea level). Note that the gray contour lines are at an elevation level of below 300 meters and are very long.

To validate that the proposed algorithm can delineate the SWPs with the incomplete contour lines, we compare the obtained results with the SWPs from the complete contour lines. First, we select an area which has the SWPs flowing down from a top hill to the Bang Wad Dam. The elevation levels of this area are presented with contour lines from 500 to 340 meters. Second, we determine the contour lines' resolution that is appropriate for our proposed SWP delineation algorithm. The appropriate resolution will make the delineation fast and accurate. By trial-and-error, a contour line should contain the discrete elevation data at every 20 meters,



(a) complete contour lines

(b) SWPs from complete contour lines

(c) type-1 incomplete contour lines

(d) SWPs from type-1 incomplete contour lines

(e) type-2 incomplete contour lines

(f) SWPs from type-2 incomplete contour lines

FIGURE 14. Comparison between the SWPs using the complete contour lines and using the type-1/type-2 incomplete contour liens.

and a contour interval between two adjacent contour lines should be 20 meters or below. For the sake of computation time and efficiency, we choose a contour line interval of 20 meters as depicted with green lines in Fig. 13. Third, some

TABLE 3. Computation time.

Case	Number of Contour lines (Complete,Incomplete)	Number of Sampled Points	Time (mins) Phase I - III	Time (mins) Phase IV - V
Complete	(9,0)	1436	20	1
Type-1	(7,2)	1394	8	1
Type-2	(0,9)	879	2	1

contour lines are converted into incomplete contour lines. The contour lines that are not crossed over by the SWPs flowing down from the top hill to the Bang Wad Dam are converted into incomplete contour lines.

Fig. 14 (a) presents the complete contour lines which are the input data of the proposed algorithm depicted with the green lines. To delineate the SWPs from these complete contour lines, we employ Algorithm 1, which determines the SWP direction in steepest descent with $\theta_{\min}$, and let every SWP segment have an equal length of $r_0 = 0.5$ meter. In Fig. 14 (b), we choose the extracted SWPs' starting points $\vec{P}_1$, which are labeled on the red line between the contour lines of 500 and 480 meters. Then, the proposed algorithm delineates the continuous SWPs, some of which converge (labeled with "C1" and "C2") while the others diverge (labeled with "D").

Next, we convert the complete 20-m contour lines in Fig. 14 (a) into the incomplete contour lines in two different types, namely type-1 and type-2 which are plotted in Fig. 14 (c) and (e), respectively. Since we focus on the SWPs located over the right-hand side of the hill, we do not make any change to these contour lines and shorten the length of the contour lines on the left-hand side of the hill. The type-1 incomplete contour lines in Fig. 14 (c) comprise only two incomplete contour lines at the elevation of 360* and 340* meters. The type-2 incomplete contour lines in Fig. 14 (e) comprise all incomplete contour lines at the elevation between 500 and 340 meters. Then, we use previous configuration in Algorithm 1 to delineate the SWPs in cases of the type-1 and type-2 incomplete contour lines, which are shown in Fig. 14 (d) and (f), respectively. The obtained results point out that the SWPs' positions in Fig. 14 (d) are exactly the same as the reference which are the extracted SWPs from the complete contour lines in Fig. 14 (b). Moreover, the SWPs are converging at the same points, which are labeled with "C1" and "C2", and are diverging at the same points, which are labeled with "D". Likewise, in Fig. 14 (f) the converging SWPs (which are labeled with "C1" and "C2") and diverging SWPs (which are labeled with "D") are exactly the same as the delineated SWPs from the complete contour lines. Therefore, the proposed algorithm maintains the accuracy of the delineated SWPs even though the input contour lines are incomplete.

To demonstrate the advantage of the proposed algorithm in terms of computation time, we measure the computation time of each mentioned case. The computation time depends on i) the number of zones, ii) the number of the contour lines, iii) the number of the sampled points along a contour line, and iv) the length of the SWP segment. To analyze the performance of the proposed algorithm, we compare computation times in three cases, namely complete contour lines, type-1 incomplete contour lines and type-2 incomplete contour lines, which are shown in Fig. 14 (a), (c) and (e), respectively. In each case, we designate a catchment area of nine zones and a position as the starting point for all cases. The calculation is in the environment with Intel(R) Xeon(R) processor (E3-1225 v5) 4-core 3.7GHz/8MB cache, 8GB memory, and Fedora Linux (Workstation) version 35. The used compiler is GCC 11.2.1 (Red Hat 11.2.1-9). In Table 3, the proposed algorithm can reduce the total computation time from 21 to 3 minutes when using the type-2 incomplete contour lines.

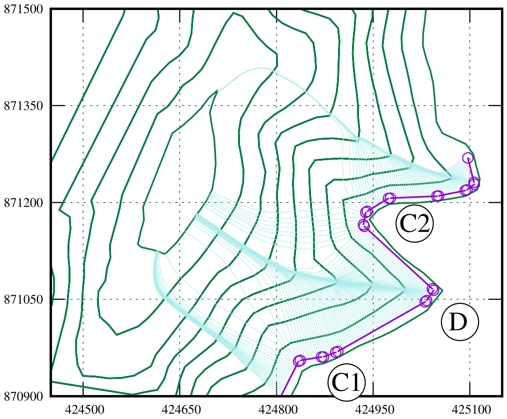


FIGURE 15. Delineating SWPs with $\theta_{\max}$ and the type-2 incomplete contour lines.

All along, the SWPs are delineated from upper contour line to lower contour line. In fact, the proposed algorithm can also delineate SWPs in the reverse direction, that is, from lower contour line to upper contour line as shown in the following example. First, the input data are chosen. We choose the type-2 incomplete contour lines as the input data to demonstrate the robustness of the proposed algorithm as shown in Fig. 15. Second, we configure Algorithm 1 with $\theta_{\max}$ instead of $\theta_{\min}$ to guide the SWP direction in steepest ascent. Third, we put the starting points along the twelve sink mouths which are depicted along the purple line in Fig. 15. Executing the Algorithm 1, we obtain the SWPs which are depicted with the blue lines in Fig. 15. Note that the sink mouths depicted with "C1" and "C2" have the SWPs which are surrounding wider area than the sink mouths labeled with "D". We obtain the same results when choosing the type-1 incomplete contour lines or the complete contour lines as the input data. With all the obtained results, the proposed algorithm can delineate valid reverse SWPs. The purpose of delineating reverse SWPs is to identify the boundary of the catchment area. The size of the catchment area is an essential parameter in the hydrological model [42]. It is used to calculate the water flow rate and evaluate the flood risk level.

VII. CONCLUSION

This paper introduces a new contour-based algorithm for delineating the SWPs from a DEM image. Unlike conventional contour-based algorithms, which demands computation resources, the proposed algorithm requires less computation resources while maintaining the position accuracy of the delineated SWPs. The semi-analytical solution of Laplace's partial differential equation is used to identify the SWP direction in either the steepest downslope or steepest upslope. With the formulated semi-analytical solution based on BEM, the problem domain is discretized along the contour lines instead of the area surrounding the contour lines. Therefore, the domain is parameterized by one variable instead of two variables. Also, the possible SWP direction ranges from 0 to 360 degrees. To validate the proposed algorithm, the method to measure the SWP position error, compared with the theoretical SWPs, in [4] is used here. The standard synthetic surfaces, where theoretical SWPs exist, namely ellipsoid, inverse ellipsoid, and inclined plane, were used for validation. The position error $E(\%)$ versus the distance of the SWP over the ellipsoid, the inverse ellipsoid, and the inclined plane are all below 0.0002, and are less than the error of any grid-based algorithms, which are in the ranges from 16.3 to 75.2.

Moreover, the semi-analytical solution is robust to BEM with incomplete contour lines, which demand even less computation resources. The real terrain with dam and mountains, where incomplete contour lines exist due to the very long contour lines at foothills, were used to show the robustness. In doing that, we compare the SWPs that are delineated from incomplete contour lines, which are shortened differently. All obtained SWPs are smooth all the way from the upper contour line to the lower. Most importantly, they converge identically when the length of the SWP segment is set at or less than 0.5 meter, and the contour line interval is 20 meters or less, and the spacing between adjacent discrete elevation data is 20 meters or narrower. Also, they diverge identically.

One of the main practical advantages, the proposed algorithm can also delineate SWPs from the lower contour line to the upper. In other words, SWPs are delineated in the reverse direction of water flow. The delineated SWPs can be used to identify the catchment area's boundary that is beneficial for the catchment's estimation in the hydrological model to analyze the water flow rate. However, we found that, in the phase of determining the direction of the SWP segment, the criteria for choosing the direction of the steepest ascent is less accurate when the starting points of the SWP segment are designated inside the inappropriate area; for example, the center of the area where the spatial distance between adjacent contour lines exceeds 50 meters. Therefore, we should be aware of this limitation when applying the proposed algorithm to estimate the catchment area.

APPENDIX A
APPROXIMATION OF GRADIENT

The numerical solution of $\nabla_P(h(\vec{P}))$ is

$$\begin{aligned}\psi\ \nabla_P h(\vec{P}) = &\sum_{j=1}^{N}\Big(\alpha_j F'_{\alpha_j}(\vec{P},\vec{Q}) + \beta_j F'_{\beta_j}(\vec{P},\vec{Q}) \\ &+ \gamma_j F'_{\gamma_j}(\vec{P},\vec{Q}) + \eta_j F'_{\eta_j}(\vec{P},\vec{Q})\Big) \\ &- \sum_{j=1}^{N}\Big(\zeta_j F'_{\zeta_j}(\vec{P},\vec{Q}) + \xi_j F'_{\xi_j}(\vec{P},\vec{Q})\Big) \end{aligned} \tag{47}$$

where:

$$F'_{\alpha_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s^3}{L_j^3}\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2}\,ds_Q \tag{48}$$

$$F'_{\beta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s^2}{L_j^2}\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2}\,ds_Q \tag{49}$$

$$F'_{\gamma_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s}{L_j}\frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2}\,ds_Q \tag{50}$$

$$F'_{\eta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{\vec{P}-\vec{Q}}{|\vec{P}-\vec{Q}|^2}\,ds_Q \tag{51}$$

$$\begin{aligned}F'_{\zeta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s}{L_j}\Bigg(&\frac{1}{|\vec{P}-\vec{Q}|^2}\hat{n} \\ &-2\frac{\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}(\vec{Q})}{|\vec{P}-\vec{Q}|^4}\left(\vec{P}-\vec{Q}\right)\Bigg)\,ds_Q\end{aligned} \tag{52}$$

$$\begin{aligned}F'_{\xi_j}(\vec{P},\vec{Q}) = \int_{C_j} \Bigg(&\frac{1}{|\vec{P}-\vec{Q}|^2}\hat{n} \\ &-2\frac{\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}(\vec{Q})}{|\vec{P}-\vec{Q}|^4}\left(\vec{P}-\vec{Q}\right)\Bigg)\,ds_Q\end{aligned} \tag{53}$$

The F'_{α_j}, F'_{β_j}, F'_{γ_j}, F'_{η_j}, F'_{ζ_j} and F'_{ξ_j} are geometric terms, as known as shape function. Their values do not depend on the elevation function and its gradient. They depend only on the shape of the boundary and the position of the point $\vec{P}$, and can be calculated after the boundary and point $\vec{P}$ were specified.

APPENDIX B
APPROXIMATION OF GRADIENT OF GRADIENT

The numerical solution of $\nabla_P^2(h(\vec{P}))$ is

$$\begin{aligned}\psi\left[\nabla_P^2 h(\vec{P})\right] &= -\sum_{j=1}^{N}\Big(\zeta_j F''_{\zeta_j}(\vec{P},\vec{Q}) + \xi_j F''_{\xi_j}(\vec{P},\vec{Q})\Big) \\ &+ \sum_{j=1}^{N}\Big(\alpha_j F''_{\alpha_j}(\vec{P},\vec{Q}) + \beta_j F''_{\beta_j}(\vec{P},\vec{Q}) \\ &+ \gamma_j F''_{\gamma_j}(\vec{P},\vec{Q}) + \eta_j F''_{\eta_j}(\vec{P},\vec{Q})\Big),\end{aligned} \tag{54}$$

where

$$F''_{\zeta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s}{L_j}\left(\frac{-2}{|\vec{P}-\vec{Q}|^4}\hat{n}\left(\vec{P}-\vec{Q}\right) + \frac{-2\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^4}\boldsymbol{I} + \frac{-2}{|\vec{P}-\vec{Q}|^4}\left(\vec{P}-\vec{Q}\right)\hat{n} + \frac{8\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^6}\left(\vec{P}-\vec{Q}\right)^2\right) ds_Q \quad (55)$$

$$RLF''_{\xi_j}(\vec{P},\vec{Q}) = \int_{C_j} \left(\frac{-2}{|\vec{P}-\vec{Q}|^4}\hat{n}\left(\vec{P}-\vec{Q}\right) + \frac{-2\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^4}\boldsymbol{I} + \frac{-2}{|\vec{P}-\vec{Q}|^4}\left(\vec{P}-\vec{Q}\right)\hat{n} + \frac{8\left(\vec{P}-\vec{Q}\right)\cdot\hat{n}}{|\vec{P}-\vec{Q}|^6}\left(\vec{P}-\vec{Q}\right)^2\right) ds_Q. \quad (56)$$

$$RLF''_{\alpha_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s^3}{L_j^3}\left(\frac{1}{|\vec{P}-\vec{Q}|^2}\boldsymbol{I} - 2\frac{1}{|\vec{P}-\vec{Q}|^2}\left(\vec{P}-\vec{Q}\right)^2\right) ds_Q \quad (57)$$

$$RLF''_{\beta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s^2}{L_j^2}\left(\frac{1}{|\vec{P}-\vec{Q}|^2}\boldsymbol{I} - 2\frac{1}{|\vec{P}-\vec{Q}|^2}\left(\vec{P}-\vec{Q}\right)^2\right) ds_Q \quad (58)$$

$$RLF''_{\gamma_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{s}{L_j}\left(\frac{1}{|\vec{P}-\vec{Q}|^2}\boldsymbol{I} - 2\frac{1}{|\vec{P}-\vec{Q}|^2}\left(\vec{P}-\vec{Q}\right)^2\right) ds_Q 0 \quad (59)$$

$$RLF''_{\eta_j}(\vec{P},\vec{Q}) = \int_{C_j} \frac{1}{|\vec{P}-\vec{Q}|^2}\boldsymbol{I} - 2\frac{1}{|\vec{P}-\vec{Q}|^2}\left(\vec{P}-\vec{Q}\right)^2 ds_Q \quad (60)$$

The F''_{α_j}, F''_{β_j}, F''_{γ_j}, F''_{η_j}, F''_{ζ_j} and F''_{ξ_j} are geometric terms, as known as shape functions. Their values do not depend on the elevation function nor its gradient. They depend only on the shape of the boundary and the position of the point $\vec{P}$ and can be calculated once the boundary is specified and the point $\vec{P}$ is chosen.

APPENDIX C
APPROXIMATION OF SURFACE

If the length of the SWP segment is short enough, the acceptable approximation can be obtained using only the constant, linear, and quadratic terms:

$$h(r,\theta) \approx a_0 + r(a_1\cos\theta + b_1\sin\theta) + r^2(a_2\cos 2\theta + b_2\sin 2\theta);\ \ r < r_0 \quad (61)$$

Since the longest length of the SWP segment that still gives an acceptable approximation depends on the shape of the surface, r_0 is chosen by trial-and-error.

To fit the quadratic approximation in (61), we select the value of the coefficients a_0, a_1, a_2, b_1 and b_2 for $r < r_0$. Usually, these coefficients are not unique since they depend on the criterion that we adopt to measure the difference between the surface and the approximation. Here, we adopt the criterion, where the approximation error at $r = 0$ is zero.

Since the approximated surface $h(0,\theta)$ equals the elevation $h(x,y)$ at point $\vec{P}$ and the series expansion has only a_0 left when $r = 0$, we obtain

$$h(0,\theta) = a_0 = h(\vec{P}(x,y)). \quad (62)$$

where the polar coordinates r and θ can be converted to the Cartesian coordinates x and y by using the trigonometric functions, namely sine and cosine: $x = r\cos\theta$ and $y = r\sin\theta$, and $h(\vec{P})$ can be determined from (27).

Next, we get rid of the term with a_0 to find coefficients a_1 and b_1 by taking first derivative to (61), and get rid of the term with a_2 and b_2 by substituting $r = 0$, we obtain

$$\begin{aligned} h_r(0,\theta) &= h_x(\vec{P})x_r \quad + h_y(\vec{P})y_r \\ a_1\cos\theta + b_1\sin\theta &= h_x(\vec{P})\cos\theta + h_y(\vec{P})\sin\theta, \end{aligned} \quad (63)$$

where, $\partial x/\partial r = \cos\theta$ and $\partial y/\partial r = \sin\theta$. When substituting $\theta = 0$ and $\theta = \pi/2$ into (63), we obtain the coefficients a_1 and b_1, respectively, that is

$$a_1 = h_x(\vec{P}) \quad (64)$$

$$b_1 = h_y(\vec{P}), \quad (65)$$

where the coefficient a_1 and b_1 are fit by some components of the gradient of the elevation $\nabla_P h(\vec{P})$, which can be determined from (32). Then, we take the second derivative to the elevation $h(x,y)$ at point $\vec{P}$ with respect to r. We obtain

$$\begin{aligned} h_{rr}(0,\theta) &= (h_x(\vec{P})x_r + h_y(\vec{P})y_r)_r \\ &= h_{xr}x_r + h_x x_{rr} + h_{yr}y_r + h_y y_{rr} \\ &= (h_{xx}x_r + h_{xy}y_r)x_r + h_x x_{rr} \\ &\quad + (h_{yx}x_r + h_{yy}y_r)y_r + h_y y_{rr} \\ &= h_{xx}(x_r)^2 + 2h_{xy}y_r x_r + h_{yy}(y_r)^2 \\ &\quad + h_x x_{rr} + h_y y_{rr} \end{aligned} \quad (66)$$

When substituting $\partial^2 x/\partial r^2 = \partial^2 y/\partial r^2 = 0$, we obtain

$$\begin{aligned} h_{rr}(0,\theta) &= h_{xx}\cos^2\theta + h_{yy}\sin^2\theta \\ &\quad + 2h_{xy}\cos\theta\sin\theta \end{aligned} \quad (67)$$

Since the second derivative of the approximated surface in (61) equals h_{rr} in (67), we obtain

$$\begin{aligned} &2(a_2\cos 2\theta + b_2\sin 2\theta) \\ &\quad = h_{xx}\cos^2\theta \\ &\qquad + h_{yy}\sin^2\theta + 2\,h_{xy}\cos\theta\sin\theta. \end{aligned} \quad (68)$$

When substituting $\theta = 0$ and $\theta = \pi/4$ into (68), we obtain the coefficients a_2 and b_2, respectively, that is

$$\begin{aligned} a_2 &= \frac{1}{2}h_{xx}(\vec{P}) \\ b_2 &= \frac{1}{4}h_{xx}(\vec{P}) + \frac{1}{2}h_{xy}(\vec{P}) + \frac{1}{4}h_{yy}(\vec{P}) \end{aligned} \tag{69}$$

To satisfy (1), $h_{xx}(\vec{P}) + h_{yy}(\vec{P})$ equals zero. Therefore,

$$\begin{aligned} a_2 &= \frac{1}{2}h_{xx}(\vec{P}) \\ b_2 &= \frac{1}{2}h_{xy}(\vec{P}), \end{aligned} \tag{70}$$

where the coefficient a_2 and b_2 are fit by some components of the gradient of gradient of the elevation $\nabla^2_P h(\vec{P})$, which can be determined from (41). Now, we obtain the approximated surface over the neighborhood of the starting point of the SWP segment

$$\begin{aligned} h(r,\theta) \approx{}& h(\vec{P}) \\ &+ r\left[h_x(\vec{P})\cos\theta + h_y(\vec{P})\sin\theta\right] \; ; r < r_0. \\ &+ r^2\left[\frac{1}{2}h_{xx}(\vec{P})\cos 2\theta \; + \; \frac{1}{2}h_{xy}(\vec{P})\sin 2\theta\right] \end{aligned} \tag{71}$$

APPENDIX D
DETERMINATION OF THE STEEPEST DESCENT

To find the θ_{min}, we differentiate the trace on the circle of the approximated surface, that is

$$\begin{aligned} \frac{dh}{d\theta}(r_0,\theta) &= r_0\left[-h_x\sin\theta + h_y\cos\theta\right] \\ &\quad + r_0^2\left[-h_{xx}\sin 2\theta + h_{xy}\cos 2\theta\right]. \end{aligned} \tag{72}$$

Then, we equate the derivative to zero, that is

$$h_x\sin\theta + r_0 h_{xx}\sin 2\theta = h_y\cos\theta + r_0 h_{xy}\cos 2\theta. \tag{73}$$

To solve this equation, trigonometric identities are applied to convert the equation into a fourth order (quartic) equation with either $\cos\theta$ term or $\sin\theta$ term. First, substituting $\sin 2\theta = 2\sin\theta\cos\theta$ and $\cos 2\theta = 2\cos^2\theta - 1$ into (73), as well as replacing $\cos\theta$ by ν and $\sin\theta$ by ω, we obtain

$$(h_x + 2\,h_{xx}r_0\nu)\,\omega = h_y\nu + h_{xy}r_0\left(2\nu^2 - 1\right). \tag{74}$$

Next, since $\nu^2 + \omega^2 = 1$, ω is replaced by $\sqrt{1-\nu^2}$, that is,

$$\begin{aligned} &(h_x + 2\,h_{xx}r_0\nu)\sqrt{1-\nu^2} \\ &\quad = h_y\nu + h_{xy}r_0\left(2\nu^2 - 1\right). \end{aligned} \tag{75}$$

Last, taking square to both sides of the equation to get rid of the square root, we manipulate into

$$c_4\nu^4 + c_3\nu^3 + c_2\nu^2 + c_1\nu + c_0 = 0, \tag{76}$$

where the coefficients c_i equal

$$\begin{aligned} c_4 &= 4\,r_0^2\left(h_{xy}^2 + h_{xx}^2\right) \\ c_3 &= 4\,r_0\left(h_x h_{xx} + h_y h_{xy}\right) \\ c_2 &= h_x^2 + h_y^2 - 4\,r_0^2\left(h_{xx}^2 + h_{xy}^2\right) \\ c_1 &= -2r_0\left(h_y h_{xy} + 2\,h_x h_{xx}\right) \\ c_0 &= h_{xy}^2 r_0^2 - h_x^2. \end{aligned} \tag{77}$$

The standard techniques can be used to solve the roots of the quartic equation in (76), and the four (possible complex) roots: $\nu_1, \nu_2, \nu_3, \nu_4$ are obtained. Next, substituting these roots into (74) gives the four corresponding values: $\omega_1, \omega_2, \omega_3, \omega_4$. Finally, θ_{min} can be calculated using the ν, ω pairs and the arctangent function. Note that θ_{min} are real number when the ν, ω pairs are real numbers with magnitudes ≤ 1. Pairs, which are complex numbers or have magnitudes greater than 1, should be discarded.

ACKNOWLEDGMENT

The authors would like to thank Pakorn Ditthakit for his valuable comments and suggestions. In addition, they would like to thank Sarayoot Nakrod and Saran Wankarem for their assistance on experimentation.

REFERENCES

[1] J. L. Mesa-Mingorance and F. J. Ariza-López, "Accuracy assessment of digital elevation models (DEMs): A critical review of practices of the past three decades," *Remote Sens.*, vol. 12, no. 16, p. 2630, Aug. 2020.

[2] G. K. Devia, B. Ganasri, and G. Dwarakish, "A review on hydrological models," *Aquatic Proc.*, vol. 4, pp. 1001–1007, Jan. 2015.

[3] W.-B. Yu, C. Su, C.-N. Yu, X.-Z. Wang, C.-J. Feng, and X.-C. Zhang, "An efficient algorithm for depression filling and flat-surface processing in raster DEMs," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 12, pp. 2198–2202, Dec. 2014.

[4] Z. Li, T. Yang, C.-Y. Xu, P. Shi, B. Yong, C.-S. Huang, and C. Wang, "Evaluating the area and position accuracy of surface water paths obtained by flow direction algorithms," *J. Hydrol.*, vol. 583, Apr. 2020, Art. no. 124619. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022169420300792

[5] X. Zheng, H. Xiong, J. Gong, and L. Yue, "A robust channel network extraction method combining discrete curve evolution and the skeleton construction technique," *Adv. Water Resour.*, vol. 83, pp. 17–27, Sep. 2015.

[6] N. Lyu, "Brief review on computing resource allocation algorithms in mobile edge computing," in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Jan. 2021, pp. 108–111.

[7] J. C. Maxwell, "L. On hills and dales," *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 40, no. 269, pp. 421–427, Dec. 1870, doi: 10.1080/14786447008640422.

[8] E. F. Wood, M. Sivapalan, K. Beven, and L. Band, "Effects of spatial variability and scale with implications to hydrologic modeling," *J. Hydrol.*, vol. 102, nos. 1–4, pp. 29–47, Sep. 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002216948890090X

[9] I. D. Moore, R. B. Grayson, and A. R. Ladson, "Digital terrain modelling: A review of hydrological, geomorphological, and biological applications," *Hydrol. Processes*, vol. 5, no. 1, pp. 3–30, Jan. 1991. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.3360050103

[10] D. M. Wolock and G. J. McCabe, "Comparison of single and multiple flow direction algorithms for computing topographic parameters in TOPMODEL," *Water Resour. Res.*, vol. 31, no. 5, pp. 1315–1324, May 1995. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/95WR00471

[11] A. Chantaveerod, T. Limpiti, and A. Seagar, "A comparative review of contour and raster based methods for the prediction of surface water flow from DEM data," *KMUTNB Int. J. Appl. Sci. Technol.*, vol. 8, no. 4, pp. 221–232, 2015.

[12] J. F. O'Callaghan and D. M. Mark, "The extraction of drainage networks from digital elevation data," *Comput. Vis., Graph., Image Process.*, vol. 28, no. 3, pp. 323–344, Dec. 1984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0734189X84800110

[13] B. Yong, L.-L. Ren, Y. Hong, J. J. Gourley, X. Chen, Y.-J. Zhang, X.-L. Yang, Z.-X. Zhang, and W.-G. Wang, "A novel multiple flow direction algorithm for computing the topographic wetness index," *Hydrol. Res.*, vol. 43, nos. 1–2, pp. 135–145, Feb. 2012.

[14] J. Fairfield and P. Leymarie, "Drainage networks from grid digital elevation models," *Water Resour. Res.*, vol. 27, no. 5, pp. 709–717, May 1991. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/90WR02658

[15] S. Orlandini, G. Moretti, M. Franchini, B. Aldighieri, and B. Testa, "Path-based methods for the determination of nondispersive drainage directions in grid-based digital elevation models," *Water Resour. Res.*, vol. 39, no. 6, Jun. 2003, Art. no. 1144. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002WR001639

[16] T. G. Freeman, "Calculating catchment area with divergent flow based on a regular grid," *Comput. Geosci.*, vol. 17, no. 3, pp. 413–422, Jan. 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009830049190048I

[17] M. C. Costa-Cabral and S. J. Burges, "Digital elevation model networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas," *Water Resour. Res.*, vol. 30, no. 6, pp. 1681–1692, Jun. 1994. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/93WR03512

[18] D. G. Tarboton, "A new method for the determination of flow directions and upslope areas in grid digital elevation models," *Water Resour. Res.*, vol. 33, no. 2, pp. 309–319, Feb. 1997. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/96WR03137

[19] Q. Zhou and X. Liu, "Error assessment of grid-based flow routing algorithms used in hydrological models," *Int. J. Geograph. Inf. Sci.*, vol. 16, no. 8, pp. 819–842, Dec. 2002, doi: 10.1080/13658810210149425.

[20] P. Bolstad and T. Stowe, "An evaluation of DEM accuracy: Elevation, slope, and aspect," *Photogramm. Eng. Remote Sens.*, vol. 60, no. 11, pp. 1327–1332, 1994.

[21] I. V. Florinsky, "Combined analysis of digital terrain models and remotely sensed data in landscape investigations," *Prog. Phys. Geogr., Earth Environ.*, vol. 22, no. 1, pp. 33–60, Mar. 1998, doi: 10.1177/030913339802200102.

[22] J. P. Walker and G. R. Willgoose, "On the effect of digital elevation model accuracy on hydrology and geomorphology," *Water Resour. Res.*, vol. 35, no. 7, pp. 2259–2268, Jul. 1999. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/1999WR900034

[23] W. Zhang and D. R. Montgomery, "Digital elevation model grid size, landscape representation, and hydrologic simulations," *Water Resour. Res.*, vol. 30, no. 4, pp. 1019–1028, Apr. 1994. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/93WR03553

[24] J. Gao, "Impact of sampling intervals on the reliability of topographic variables mapped from grid DEMs at a micro-scale," *Int. J. Geograph. Inf. Sci.*, vol. 12, no. 8, pp. 875–890, Dec. 1998, doi: 10.1080/136588198241545.

[25] P. Holmgren, "Multiple flow direction algorithms for runoff modelling in grid based elevation models: An empirical evaluation," *Hydrol. Processes*, vol. 8, no. 4, pp. 327–334, Jul. 1994. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.3360080405

[26] P. J. J. Desmet and G. Govers, "Comparison of routing algorithms for digital elevation models and their implications for predicting ephemeral gullies," *Int. J. Geograph. Inf. Syst.*, vol. 10, no. 3, pp. 311–331, Apr. 1996, doi: 10.1080/02693799608902081.

[27] I. D. Moore, E. M. O'Loughlin, and G. J. Burch, "A contour-based topographic model for hydrological and ecological applications," *Earth Surf. Processes Landforms*, vol. 13, no. 4, pp. 305–320, Jun. 1988. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.3290130404

[28] R. B. Grayson, I. D. Moore, and T. A. Mcmahon, "Physically based hydrologic modeling: 1. A terrain-based model for investigative purposes," *Water Resour. Res.*, vol. 28, no. 10, pp. 2639–2658, Oct. 1992. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/92WR01258

[29] W. R. Dawes and D. Short, "The significance of topology for modeling the surface hydrology of fluvial landscapes," *Water Resour. Res.*, vol. 30, no. 4, pp. 1045–1055, Apr. 1994. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/93WR02479

[30] C. J. Maunder, "An automated method for constructing contour-based digital elevation models," *Water Resour. Res.*, vol. 35, no. 12, pp. 3931–3940, Dec. 1999. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/1999WR900166

[31] G. Menduni and V. Riboni, "A physically based catchment partitioning method for hydrological analysis," *Hydrol. Processes*, vol. 14, nos. 11–12, pp. 1943–1962, 2000.

[32] G. Moretti and S. Orlandini, "Automatic delineation of drainage basins from contour elevation data using skeleton construction techniques," *Water Resour. Res.*, vol. 44, no. 5, pp. 1–16, May 2008. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007WR006309

[33] G. Menduni, A. Pagani, M. C. Rulli, and R. Rosso, "A non-conventional watershed partitioning method for semi-distributed hydrological modelling: The package ALADHYN," *Hydrol. Processes*, vol. 16, no. 2, pp. 277–291, 2002. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.340

[34] H. Mizukoshi, "Use of contour-based DEMs for deriving and mapping topographic attributes," *Photogramm. Eng. Remote Sens.*, vol. 68, no. 1, pp. 83–93, 2002.

[35] M. C. Rulli, "A physically based watershed partitioning method," *Adv. Water Resour.*, vol. 33, no. 10, pp. 1206–1215, Oct. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0309170810001235

[36] B. A. DeVantier and A. D. Feldman, "Review of GIS applications in hydrologic modeling," *J. Water Resour. Planning Manage.*, vol. 119, no. 2, pp. 246–261, 1993.

[37] G. E. Tucker, S. T. Lancaster, N. M. Gasparini, R. L. Bras, and S. M. Rybarczyk, "An object-oriented framework for distributed hydrologic and geomorphic modeling using triangulated irregular networks," *Comput. Geosci.*, vol. 27, no. 8, pp. 959–973, Oct. 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098300400001345

[38] E. R. Vivoni, V. Y. Ivanov, R. L. Bras, and D. Entekhabi, "Generation of triangulated irregular networks based on hydrological similarity," *J. Hydrol. Eng.*, vol. 9, no. 4, pp. 288–302, Jul. 2004.

[39] V. Y. Ivanov, E. R. Vivoni, R. L. Bras, and D. Entekhabi, "Catchment hydrologic response with a fully distributed triangulated irregular network model," *Water Resour. Res.*, vol. 40, no. 11, Nov. 2004, Art. no. W11102. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2004WR003218

[40] C. Mita and N. Catsaros, "Simulating pollutant transport over complex terrain: The hydrological component," *Open Chem.*, vol. 10, no. 4, pp. 1223–1235, Aug. 2012, doi: 10.2478/s11532-012-0047-9.

[41] H. R. D. A. Freitas, C. D. C. Freitas, S. Rosim, and J. R. D. F. Oliveira, "Drainage networks and watersheds delineation derived from TIN-based digital elevation models," *Comput. Geosci.*, vol. 92, pp. 21–37, Jul. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098300416300954

[42] T. Pilz, T. Francke, and A. Bronstert, "LumpR 2.0.0: An R package facilitating landscape discretisation for hillslope-based hydrological models," *Geosci. Model Develop.*, vol. 10, no. 8, pp. 3001–3023, Aug. 2017.

[43] S. Kopp and D. Siegel, "The NHD web," in *GIS for Surface Water: Using the National Hydrography Dataset*, Redlands, CA, USA: ESRI Press, Jan. 2018, ch. 7, p. 303.

[44] C. Birkel and A. Correa, *Rainfall-Runoff Modeling: A Brief Overview*. Amsterdam, The Netherlands: Elsevier, Dec. 2019.

[45] A. W. Bohannon and T. Hromadka, "Modeling ground water problems using the complex polynomial method," *Prof. Geologist*, vol. 46, pp. 39–46, Mar./Apr. 2009.

[46] A. W. Bohannon and T. V. Hromadka, "The complex polynomial method with a least-squares fit to boundary conditions," *Eng. Anal. Boundary Elements*, vol. 33, nos. 8–9, pp. 1100–1102, Aug. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0955799709000460

[47] T. R. Dean and T. V. Hromadka, "A collocation CVBEM using program mathematica," *Eng. Anal. Boundary Elements*, vol. 34, no. 4, pp. 417–422, Apr. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0955799709002124

[48] T. R. Dean, T. V. Hromadka, II, T. Kastner, and M. Phillips, "Modeling potential flow using laurent series expansions and boundary elements," *Numer. Methods for Partial Differ. Equ.*, vol. 28, no. 2, pp. 573–586, Mar. 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/num.20636

[49] B. Wilkins, N. Flowerday, A. Kratch, J. Greenberg, B. Redmond, A. Baily, T. Hromadka, K. Hood, R. Boucher, and H. D. Mcinvale, "A computational model of groundwater mound evolution using the complex variable boundary element method and generalized Fourier series," *Prof. Geol.*, vol. 54, no. 1, pp. 1–15, 2017.

[50] L. Polidori and M. El Hage, "Digital elevation model quality assessment methods: A critical review," *Remote Sens.*, vol. 12, no. 21, p. 3522, Oct. 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/21/3522

[51] Y. Chantaveerod, "Relative flood risk model," M.S. thesis, Dept. Elect. Eng., Prince Songkla Univ., Songkla, Thailand, Sep. 2004.

[52] F. J. Rizzo, P. A. Martin, L. Pan, and D. Zhang, "Exact Green's functions and the boundary element method," *WIT Trans. Model. Simul.*, vol. 10, no. 3, pp. 19-26, 1995. [Online]. Available: https://www.witpress.com/elibrary/wit-transactions-on-modelling-and-simulation/10/9814

[53] S. Zlobec, "The fundamental theorem of calculus for Lipschitz functions," *Math. Commun.*, vol. 13, no. 2, pp. 215–232, 2008. [Online]. Available: https://hrcak.srce.hr/30887

[54] I. V. Florinsky, "Digital elevation models," in *Digital Terrain Analysis in Soil Science and Geology*, 2nd ed., I. V. Florinsky, Ed. New York, NY, USA: Academic, 2016, ch. 3, pp. 77–108. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128046326000031

AJALAWIT CHANTAVEEROD was born in Hat Yai, Songkhla, Thailand. He received the bachelor's and master's degrees in electrical engineering from the Prince of Songkhla University, Thailand, in 2002 and 2004, respectively, and the Ph.D. degree in electrical engineering from Chulalongkorn University, Thailand, in 2009. He is currently a Lecturer at the School of Engineering and Technology, Walailak University, Thailand. His current research interests include applications of boundary integral equations in engineering, numerical solution of Maxwell's equations based on Clifford algebra, and parallel algorithm for computational electromagnetics.

KAMPOL WORADIT (Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees in electrical engineering from Chulalongkorn University, Thailand, in 2002 and 2010, respectively. He was a Visiting Student with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, from 2007 to 2008, and was on a Postgraduate Research Attachment with the Modulation and Coding Department, Institute for Infocomm Research, Singapore. He was a Faculty Member with the Department of Electrical Engineering, Srinakharinwirot University, Thailand, from 2010 to 2020. He is currently a Faculty Member with the Department of Computer Engineering, Chiang Mai University, Thailand. His research interests include wireless communications, the Internet of Things, and physical layer security. He received the ECTI-CON Best Paper Award in 2019 and the Royal Golden Jubilee Scholarships from the Thailand Research Fund for his Ph.D. Program.

ANDREW D. SEAGAR (Life Senior Member, IEEE) received the bachelor's and Ph.D. degrees in electrical engineering from the University of Canterbury, New Zealand, in 1978 and 1983, respectively. He worked in various full-time research, industry, and university positions in U.K., Australia, Canada, and Thailand.

THUNYAWAT LIMPITI (Member, IEEE) was born in Phatthalung, Thailand, in 1982. He received the B.Eng. and M.Eng. degrees in telecommunications engineering and the D.Eng. degree in electrical engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2005, 2008, and 2013, respectively. He is currently an Assistant Professor with the School of Engineering and Resources, Walailak University, Thai Buri, Thailand. His current research interests include reconfigurable antenna design, microwave sensor design, and microwave techniques for dielectric properties determination.