

Lightweight Calculation of the Area of a Closed Contour by an Inner Skin Algorithm

Hiroyuki Goto¹ and Yoichi Shimakawa²

Abstract

This research presents a lightweight algorithm for calculating the area enclosed by a contour in a two-dimensional space, the shape of which is determined by an elevation function. The area is calculated using Green's theorem, for which a one-dimensional line integration along the contour is required. To trace the contour efficiently, we introduce the notion of a shortcut vector determined by a certain rotation. In view of the feature in tracing and circulating the contour, we call this an 'inner skin algorithm'. Though our primary aim is to apply the algorithm to terrain topography, the method would also be useful in various engineering fields, such as civil engineering, mechanical design, and electromagnetics.

¹ Department of Industrial & System Engineering, Hosei University, Koganei, Tokyo 184-8584, Japan, e-mail: goto-h@hosei.ac.jp

² Department of Computer Science and Technology, Salesian Polytechnic, Machida, Tokyo 194-0215, Japan, e-mail: simakawa@salesio-sp.ac.jp

Mathematics Subject Classification: 33F05, 65D10, 65D30, 86-08, 86A99

Keywords: Area, Closed contour, Elevation function, Green's theorem, Shortcut, Inner skin

1 Introduction

We discover a lightweight algorithm for calculating the area of a region bounded by a closed contour, the shape of which is determined by an elevation function. The salient features of the method are:

- No broken line approximation is used in contour generation.
- No convergence iterations are carried out in contour tracing.
- No double integration is performed in calculating the area.
- No buffering technique is used to detect termination of circulation.

Though our primary interest is in calculating areas of lakes, swamplands, and ponds using digital elevation data, the algorithm itself would also be useful in many other engineering fields.

The problem of tracing and drawing a contour is also referred to as 'contour tracing' and 'contouring', and much attention has been drawn to the problem in various fields. The pioneering research on this topic is Cottafava and Moli [1] which treated cases for which elevations are given over a regular grid. McLain [2] extended the analysis to elevations given only on scattered points. Further extensions have been given by Lopes and Brodlie [3], and Osorio and Brodlie [4]. The former uses a bilinear model as elevation function, and approximates two broken contour lines in a grid by a hyperbola. The latter handles cases for which there is a degree in uncertainty in the elevation function. Interested readers can obtain an overview of the traditional methodologies and recent advances in Li *et al.* [5], and Floriani and Spagnuolo [6].

In terrain modelling and analysis, applications using digital elevation data such

as Geographic Information System (GIS) have been given emphasis in recent years. In using Digital Elevation Model (DEM) data, triangular-based reference points are frequently used to model terrain surfaces. The seminal work using this approach is Gold *et al.* [7], with many improvements and generalisations carried out. Dobkin *et al.* [8] proposes a triangulation method by which a contour can be traced efficiently. One of the most widely used methods in recent years is the Triangular Irregular Network (TIN), a vector-based model that consists of a set of discrete data points (x, y, z) , called mass points. An advantage of this method is the capability of describing the surface at different levels of resolution, and its efficiency in storing data. An application to extract spatial characteristics using TIN is presented in Li and Ai [9].

Our current interest is in calculating areas of closed contours. If we require high-precision results, making a fine grid is a straightforward solution. However, the volume of required reference data expands quadratically in proportion to resolution. This situation is the same whether we use regular grids or TINs, because both of these are based on piecewise linear approximations.

However, if we can approximate the contour by a nonlinear elevation function, resolution may not be that important, allowing precision to be targeted independently. From the above-mentioned papers, we find use of a continuous nonlinear elevation function, for example, McLain [2], Lopes and Brodlie [3], and Iri *et al.* [10], although use of elevation function values only on regular grids essentially reduces the problem to piecewise linear approximations. Thus, the benefit in using a nonlinear function representing the curve has not been exploited to the full.

With this perspective, we develop an algorithm that efficiently calculates the area of a closed contour by using several properties of the elevation function not utilised thus far. The primary essences of the proposed algorithm are:

- The method can be understood as a vector-based method and not a raster-based one.

- The concepts of shortcut vector and angle are introduced for tracing the contour.
- One-dimensional line integration is carried out along the contour.
- The set of traced points are not recorded and only the initial and latest points are used to determine termination.

In view of the behaviour in tracing and circulating the target contour, we refer to the proposed approach as inner skin algorithm.

2 Proposed Method

After introducing some prerequisites on elevation functions, we focus on calculating the area of a closed curve. We propose an algorithm for efficiently tracing a contour and a method for adjusting the step size for line integration.

2.1 Prerequisites

In three-dimensional space, the elevation function is determined by $z = f(x, y) \equiv f(\mathbf{x})$. We suppose that f is continuous in x and y , and $\nabla f \neq (0 \ 0)^T$ follows on the target contour.

In general, a contour at height h can be understood as a collection of points for which $f(\mathbf{x}) = h$ holds. However, without loss of generality, we can set $f(\mathbf{x}) - h$ as the new elevation function of $f(\mathbf{x})$, and assume

$$f(\mathbf{x}) = 0. \tag{1}$$

2.2 Area of the region by a closed curve

We derive a formula for calculating the area of a closed region surrounded by a simply-connected curve.

In two-dimensional space, let D be a region formed by a simply-connected curve, and let ∂D be the oriented boundary as defined by moving counter clockwise along the boundary of D . Here, we consider Green's theorem [11]:

$$\oiint_D \nabla \times \mathbf{u} dx dy = \oint_{\partial D} \mathbf{u}^T dx .$$

where \mathbf{u} is an arbitrary class C^1 vector. By putting $\mathbf{u} = (-y \ x)^T$, this is rewritten

$$2 \iint_D dx dy = \oint_{\partial D} (x dy - y dx) . \quad (2)$$

We note here that the half of the left hand-side is equal to the area of region D , which we shall denote by S .

Let the current \mathbf{x} be a solution of Eq. (1) and consider an infinitesimal displacement $d\mathbf{x} \equiv (dx \ dy)^T$. The Taylor expansion of $f(\mathbf{x} + d\mathbf{x})$ to first order is

$$f(\mathbf{x} + d\mathbf{x}) = f(\mathbf{x}) + d\mathbf{x}^T \nabla f(\mathbf{x}) .$$

If we displace along the contour ∂D , $f(\mathbf{x} + d\mathbf{x}) = f(\mathbf{x}) = 0$ is followed and $d\mathbf{x}$ is perpendicular to $\nabla f(\mathbf{x})$, a normal vector of the contour at \mathbf{x} . Letting the unit normal vector be denoted by \mathbf{e}_n ($= \nabla f(\mathbf{x}) / |\nabla f(\mathbf{x})|$), we create the directional vector by rotating \mathbf{e}_n for $\pi/2$ in a counter clockwise direction. We then obtain the unit directional vector \mathbf{e}_d as follows:

$$\mathbf{e}_d = J(\pi/2) \cdot \mathbf{e}_n = (-f_y \ f_x)^T / |\nabla f| ,$$

where $J(\theta)$ represents the two-dimensional rotation matrix given below:

$$J(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} .$$

Denoting the length of dx by dl , it follows $dx = (dx \ dy)^T = dl \cdot e_d = (-f_y \ f_x)^T (dl / |\nabla f|)$. By substituting this into the right hand-side of Eq. (2), the area S is calculated by

$$S = \frac{1}{2} \oint_{\partial D} \frac{xf_x + yf_y}{\sqrt{f_x^2 + f_y^2}} dl = \oint_{\partial D} g(\mathbf{x}) dl, \quad (3)$$

where

$$g(\mathbf{x}) = (\mathbf{x}^T \mathbf{e}_n) / 2. \quad (4)$$

The feature of this formula is that the area is calculated by a one-dimensional line integration, and not double integration.

2.3 Contour tracing

To perform the line integration in Eq. (3), we consider a method for tracing the contour ∂D . If dl were infinitesimal, dx would be exactly perpendicular to $\nabla f(\mathbf{x})$. However, in numerical computations, dl is obtained by partitioning the contour into small finite increments. We replace dl and dx by δl and $\delta \mathbf{x}$, respectively, to denote this discretisation. The specific form of $\delta \mathbf{x}$ is

$$\delta \mathbf{x} = \delta l \cdot \mathbf{e}_d. \quad (5)$$

After discretisation, point $\mathbf{x} + \delta \mathbf{x}$ may deviate from ∂D , and

$$f(\mathbf{x} + \delta \mathbf{x}) \equiv \delta f, \quad (6)$$

is not equal to zero. Thus, we consider a shortcut vector $\delta \mathbf{s}$ that satisfies

$$f(\mathbf{x} + \delta \mathbf{s}) = 0, \quad (7)$$

by rotating the directional vector $\delta \mathbf{x}$:

$$\delta \mathbf{s} = J(\theta) \cdot \delta \mathbf{x}, \quad (8)$$

where θ is the counter-clockwise rotation angle.

We depict the relationships between the relevant vectors, contour, and angle in Figure 1. The figure shows the case for $\delta f > 0$ and the circulation direction is counter clockwise. In geometrical terminology, this corresponds to a circulation for a lake; in contrast, the direction is clockwise if $\delta f < 0$, corresponding to a circulation for a hill.

Here, we consider the Taylor expansion of $f(\mathbf{x} + \delta s)$ for θ around $\theta = 0$ to first-order:

$$f(\mathbf{x} + \delta s) = f(\mathbf{x} + J(\theta) \cdot \delta \mathbf{x}) \cong f(\mathbf{x} + \delta \mathbf{x}) + \left. \frac{df}{d\theta} \right|_{\theta=0} \cdot \theta \quad . \quad (9)$$

Then, we determine θ from Eq. (7) which is understood as being the steepest descent method [12] for just one iteration.

For later, we consider the derivative of $J(\theta)$ for θ :

$$\frac{dJ(\theta)}{d\theta} = \begin{pmatrix} -\sin \theta & -\cos \theta \\ \cos \theta & -\sin \theta \end{pmatrix} = J\left(\theta + \frac{\pi}{2}\right) \quad .$$

With respect to $\mathbf{x} + \delta s$, its derivative with respect to θ is

$$\frac{d\mathbf{x}}{d\theta} = \frac{dJ(\theta)}{d\theta} \delta \mathbf{x} = J\left(\theta + \frac{\pi}{2}\right) \cdot \delta l \cdot J\left(\frac{\pi}{2}\right) \cdot \mathbf{e}_n = -\delta l \cdot J(\theta) \cdot \mathbf{e}_n \quad .$$

Moreover,

$$\frac{df}{d\theta} = (\nabla f)^T \frac{d\mathbf{x}}{d\theta} = -(\nabla f)^T J(\theta) \mathbf{e}_n \cdot \delta l = -|\nabla f| \cos \theta \cdot \delta l \quad .$$

Then, at $\theta = 0$, we obtain

$$\left. \frac{df}{d\theta} \right|_{\theta=0} = -|\nabla f| \delta l \quad . \quad (10)$$

Recalling Eqs. (6), (7), (9), and (10), the approximation of θ can be expressed as follows:

$$\theta \cong \frac{\delta f}{|\nabla f| \delta l}. \quad (11)$$

Substituting Eq. (5) in Eq. (8), δs is reduced to

$$\delta s = \delta l \cdot J(\theta) \cdot \mathbf{e}_d = \delta l \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \mathbf{e}_d = \delta l \cdot (\cos \theta \cdot \mathbf{e}_d - \sin \theta \cdot \mathbf{e}_n) \quad . \quad (12)$$

If we consider small increments δl , $\theta \cong 0$ holds and Eq. (12) can be simplified. Noting that

$$\sin \theta \cong \theta \quad \text{and} \quad \cos \theta \cong 1 \quad (13)$$

holds for $\theta \cong 0$, Eq. (12) can be approximated by

$$\delta s \cong \delta l \cdot (\mathbf{e}_d - \theta \cdot \mathbf{e}_n) = \delta \mathbf{x} + \delta \mathbf{m} \quad ,$$

where

$$\delta \mathbf{m} = -\delta l \cdot \theta \cdot \mathbf{e}_n \quad . \quad (14)$$

If we shift the current point \mathbf{x} based on

$$\mathbf{x}_{\text{new}} \leftarrow \mathbf{x} + \delta \mathbf{s} \quad ,$$

\mathbf{x}_{new} is close to or exactly on ∂D . By repeating this incremental displacement we ultimately return to the initial point, completing a circuit along the contour.

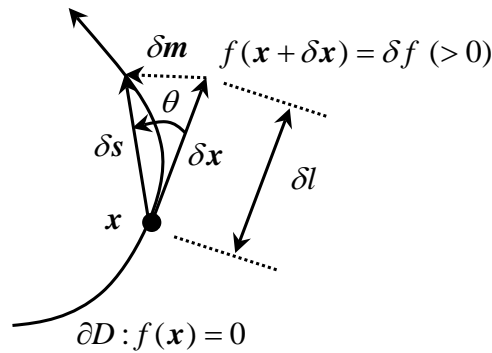


Figure 1: Relationship between directional and shortcut vectors

2.4 Numerical integration

We now examine numerical integration of Eq. (2). Here we consider the line segment $xdy - ydx$ on the right-hand side. With the help of the shortcut vector, we replace dx with δs . Then using Eqs. (5) and (8), we obtain

$$\frac{1}{2}(xdy - ydx) = \frac{1}{2} \mathbf{x}^T J \left(-\frac{\pi}{2} \right) \delta s = \frac{\mathbf{x}^T}{2} J \left(-\frac{\pi}{2} \right) J(\theta) \mathbf{e}_d \delta l = \frac{\mathbf{x}^T J(\theta) \mathbf{e}_n}{2} \cdot \delta l.$$

This implies that $g(\mathbf{x})$ in the line integration Eq. (3) is replaced by

$$g(\mathbf{x}) = [\mathbf{x}^T J(\theta) \mathbf{e}_n] / 2,$$

rather than Eq. (4). Using Eq. (13), this is further simplified to

$$g(\mathbf{x}) = \mathbf{x}^T (\cos \theta \cdot \mathbf{e}_n + \sin \theta \cdot \mathbf{e}_d) / 2 \cong \mathbf{x}^T (\mathbf{e}_n + \theta \cdot \mathbf{e}_d) / 2. \quad (15)$$

2.5 Curvature properties

We consider the Taylor expansion of $f(\mathbf{x} + \delta \mathbf{x})$ to second-order:

$$f(\mathbf{x} + \delta \mathbf{x}) \cong f(\mathbf{x}) + \delta \mathbf{x}^T \nabla f + \delta \mathbf{x}^T H(\mathbf{x}) \delta \mathbf{x} / 2,$$

where $H(\mathbf{x}) = \nabla \nabla^T f(\mathbf{x})$ represents the value of the Hessian matrix at point \mathbf{x} .

Noting that $f(\mathbf{x} + \delta \mathbf{x}) - f(\mathbf{x}) = \delta f$, $f(\mathbf{x}) = 0$, and $\delta \mathbf{x}^T \nabla f \cong 0$, we obtain

$$\delta f \cong \delta \mathbf{x}^T H(\mathbf{x}) \delta \mathbf{x} / 2.$$

Subsequently, recalling Eqs. (5) and (11), θ can be approximated as

$$\theta \cong \frac{\delta \mathbf{x}^T H(\mathbf{x}) \delta \mathbf{x}}{2 |\nabla f| \delta l} = \frac{\delta l}{2 |\nabla f|} \mathbf{e}_d^T H(\mathbf{x}) \mathbf{e}_d = \frac{1}{2} k(\mathbf{x}) \cdot \delta l \propto \delta l, \quad (16)$$

where $k(\mathbf{x})$ is known as signed curvature at \mathbf{x} . Denoting the radius of curvature on \mathbf{x} by $\rho(\mathbf{x})$, this is also expressed by

$$\delta l \cong \rho(\mathbf{x}) \cdot |2\theta|. \quad (17)$$

We note here that the right-hand side represents the length of an arc with central point \mathbf{x} , radius $\rho(\mathbf{x})$, and central angle $|2\theta|$.

2.6 Adjustment of the step size

A sudden change in the curve along the contour would be indicated by the shortcut angle θ increasing at the point. Accordingly, declines in precision accrue for θ , $\sin \theta$, and $\cos \theta$ in Eqs. (11) and (12). For such cases, the step size δl should be shortened to avoid deviation from the contour.

Recalling the proportionality of θ to δl in Eq. (16), we adjust δl in a manner so that Eq. (13) still holds. Let $\theta_M (> 0)$ and $\delta l_M (> 0)$ be the maximum limit of $|\theta|$ and the step size δl , respectively.

If $|\theta| > \theta_M$ occurs at a point, then we decrease the step size in accordance with:

$$\delta l \Leftarrow \frac{\theta_M}{\min(|\theta|, \pi/2)} \cdot \delta l. \quad (18)$$

The minimum operation in the denominator is to avoid stall if $|\theta|$ obtained by Eq. (11) is very large. After this adjustment, we have to recalculate the relevant values that depend on δl : *i.e.*, $\delta \mathbf{x}$, δf , and θ .

In contrast, if $|\theta| \leq \theta_M$, we do not magnify δl at this step to avoid recalculation, and magnify it in the next step. Even if this condition holds, we set δl to δl_M at the maximum even if $|\theta| \cong 0$. This is because there may be a sudden change in the curve on the small segment. In view of these, the adjustment process for δl in $|\theta| \leq \theta_M$ can be expressed as

$$\delta l \leftarrow \min \left[\frac{\theta_M}{\max(|\theta|, \varepsilon)} \cdot \delta l, \delta l_M \right], \quad (19)$$

where ε is a very small positive value, which plays a role if $|\theta|$ is zero or very small.

2.7 Small variance curvature

A desirable maximum step size δl_M is considered for a contour with small variance curvature. Let us recall Eq. (17), and let $\bar{\rho}$ be the typical radius of curvature during the circulation of ∂D . If we set $\delta l_M \gg 2\bar{\rho}\theta_M$, the condition $|\theta| > \theta_M$ would often hold during a circulation. This would result in frequent adjustments of δl and recalculation of $\delta \mathbf{x}$, δf , and θ , from which computation times increase. In contrast, if we set $\delta l_M \ll 2\bar{\rho}\theta_M$, the step size δl is always small compared with the target precision. This would also raise inefficiencies in computation. Accordingly, if the curvature of ∂D does not vary significantly and we can estimate $\bar{\rho}$ in advance, even roughly, the computation time can be reduced if we set

$$\delta l_M = 2 \cdot \bar{\rho} \cdot \theta_M.$$

Alternatively, if there is no *a priori* information about the shape, we have to set θ_M and δl_M independently.

2.8 Overall algorithm

We outline the overall algorithm for calculating the area S based on Eq. (3). First, we set the maximum shortcut angle θ_M and the step sizes δl_m and δl_M .

Then, we find a solution \mathbf{x}_0 that satisfies Eq. (1) using an efficient method such as Newton's method. Moreover, let $L \leftarrow 0$ and $S \leftarrow 0$ be the cumulative length and square elements.

Next, we calculate the shortcut angle θ using Eq. (11), where $\delta\mathbf{x}$ and δf are calculated by Eqs. (5) and (6), respectively. If $|\theta| > \theta_M$, we shorten δl using Eq. (18) and recalculate $\delta\mathbf{x}$, δf , and θ . Subsequently, we calculate the line and square elements on \mathbf{x} and add them to L and S , respectively:

$$L \leftarrow L + \delta l, \quad (20)$$

$$S \leftarrow S + g(\mathbf{x}) \cdot \delta l, \quad (21)$$

where we use $g(\mathbf{x})$ in Eq. (15).

Then, we proceed to the next point using the assignment

$$\mathbf{x}_{\text{new}} \leftarrow \mathbf{x} + \delta\mathbf{x} + \delta\mathbf{m}. \quad (22)$$

If $|\theta| < \theta_M$ holds, we magnify δl using Eq. (19). Subsequently, we calculate $\delta\mathbf{x}$, δf , and θ again, add the new segment element, and proceed to the next point.

We repeat this addition and shift until the current point \mathbf{x} lies within the vicinity of the initial point \mathbf{x}_0 . Termination is determined by the condition

$$|\mathbf{x} - \mathbf{x}_0| \leq \delta l \quad \text{and} \quad |\mathbf{x} - \mathbf{x}_0| < d_{\text{pre}}, \quad (23)$$

where d_{pre} is the value of $|\mathbf{x} - \mathbf{x}_0|$ in the previous step. If Eq. (23) holds, we set the next point to \mathbf{x}_0 forcibly. With this final shift, we have completed the circulation.

To aid the readers' comprehension, we present in Figure2 a pseudo code of the overall algorithm. The generic calls 'Find', 'Set', 'Cal', and 'Update' are procedural calls that find a solution for the given argument, perform parameter settings, calculate the quantity given as argument, and update the value of the

given argument, respectively.

We note that the algorithm presented in Figure 2 returns the signed area, which may be negative for a certain type of the elevation functions. Let us recall Eq. (3) here. In the line integration, the sign is positive if the circulation is counter clockwise, referred to earlier as a lakeside circulation. A hillside circulation is determined by clockwise circulation and the resulting integration is negative. Thus taking the absolute value is required for obtaining the area; the sign refers to how the area is oriented.

```

01 Find  $x_0$  subject to  $f(x_0) = 0$  (Eq. 1)
02 Set  $\theta_M, \delta l_M$ 
03  $\delta l \leftarrow \delta l_M, x \leftarrow x_0, L \leftarrow 0, S \leftarrow 0, d_{pre} \leftarrow 0$ 
04 Begin loop
05 Cal  $f_x, f_y, |\nabla f|$ 
06 Cal  $e_n, e_d$ 
07 Cal  $\delta x, \delta f, \theta$  (Eqs. 5, 6, 11)
08 If  $|\theta| > \theta_M$ 
09 Update  $\delta l$  (Eq. 18)
10 Cal  $\delta x, \delta f, \theta$  (Eqs. 5, 6, 11)
11 End if
12 Cal  $\delta m$  (Eq. 14)
13 If  $|x - x_0| \leq \delta l$  and  $|x - x_0| < d_{pre}$  (Eq. 23)
14  $\delta l \leftarrow |x - x_0|$ 
15  $\delta x \leftarrow x_0 - x$ 
16  $\delta m \leftarrow \mathbf{0}$ 
17 Cal  $\delta f, \theta$  (Eqs. 6, 11)
18 End if
19  $d_{pre} \leftarrow |x - x_0|$ 
20 Cal  $g(x)$  (Eq. 15)
21  $L \leftarrow L + \delta l$  (Eq. 20)
22  $S \leftarrow S + g(x) \cdot \delta l$  (Eq. 21)
23  $x \leftarrow x + \delta x + \delta m$  (Eq. 22)
24 If  $x = x_0$  Exit Loop
25 If  $|\theta| \leq \theta_M$ 
26 Update  $\delta l$  (Eq. 19)
27 End if
28 End loop

```

Figure 2: Pseudo code for calculating the area

3 Remarks

We give several complementary remarks and precautions when performing practical applications.

3.1 Elevation function

For the elevation function $z = f(\mathbf{x})$, we can use either an analytically-defined or an experimentally-determined function. A typical approach for the latter case is to use digital elevation data. In particular, if elevations are given on a regular grid, a straightforward method to model terrain surface would be using a bilinear or bicubic spline functions below:

$$f(\mathbf{x}) = a_{00} + a_{10}(x - x_0) + a_{01}(y - y_0) + a_{11}(x - x_0)(y - y_0),$$

$$f(\mathbf{x}) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} (x - x_0)^i (y - y_0)^j,$$

where (x_0, y_0) is the base coordinate of the target cell. If we use either of the above functions, the first derivative ∇f can be calculated analytically. In contrast, if the exact ∇f cannot be obtained, we use an approximation from numerical differentiation.

3.2 Non-simply-connected curves

Suppose our goal is to calculate the area of a lake dotted with islands. Thus far, we have assumed that the target region is simply-connected; the proposed algorithm then cannot be used in the present form. However, recalling that the line integration in Eq. (3) is negative for hillside circulation, the area of a hill can be subtracted conveniently by simply adding the results of the line integration. Thus,

if we can identify all islands within the lake and determine their initial points \mathbf{x}_0 , the area of the lake can be obtained by simply adding the results of the line integration before taking the absolute value.

A simple example of a lake with two islands is shown in Figure 3. The area filled with grey represents the lake. Curve ∂D_1 represents the lake shoreline and curves ∂D_2 and ∂D_3 the island shorelines. Setting S_1 , S_2 , and S_3 as the line integrations based on Eq. (3), $S_1 > 0$, $S_2 < 0$, and $S_3 < 0$ follow and the area of the lake S is obtained by

$$S = S_1 + S_2 + S_3.$$

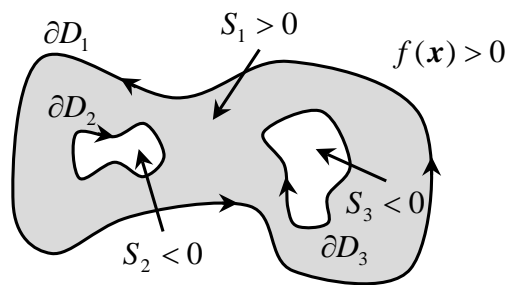


Figure 3: Area of a two-island lake

4 Numerical Computation

We perform numerical computations for several elevation functions $z = f(\mathbf{x})$, for which the shapes of the contours formed by $f(\mathbf{x}) = 0$ are well known and their areas can be analytically obtained. We implement the program codes for GNU Octave version 3.6.2 running on Windows 7, and derive the partial derivatives of $f(\mathbf{x})$ by hand. We compute the areas for various maximum

shortcut angles $\theta_M = 0.25, 0.1, 0.025, 0.01, 0.0025,$ and $0.001,$ setting the maximum step size using $\delta l_M = 2\theta_M$. The initial point is determined using the ‘fsolve’ function initialised by starting with search point $(1, 1)$.

4.1 Circles and ellipses

We consider four functions $f(x) = :$ (1) $x^2 + y^2 - 1$, (2) $1/(x^2 + y^2) - 1$, (3) $x^2/4 + 4y^2 - 1$, and (4) $17(x^2 + y^2) - 30xy - 8$. The contour shape of the first pair is circular, and for the second pair ellipsoidal. Function (4) is obtained by rotating function (3) by $\pi/4$. The area enclosed by these figures is π ; note that a singular point occurs at $(x, y) = (0, 0)$ in function (2), and the circulation direction is clockwise.

Table 1: Calculation results for the area of four elevation functions bounding equal area

	$f(x)$	$\theta_M =$ 0.25	0.1	0.025	0.01	0.0025	0.001
(1)	$x^2 + y^2 - 1$	3.0630 (13) 2.50e-2	3.1223 (32) 6.13e-3	3.1403 (126) 4.13e-4	3.1414 (315) 6.66e-5	3.1416 (1257) 4.16e-6	3.1416 (3142) 6.67e-7
(2)	$\frac{1}{x^2 + y^2} - 1$	-3.1790 (13) 1.19e-2	-3.1269 (32) 4.69e-3	-3.1403 (126) 4.07e-4	-3.1414 (315) 6.64e-5	-3.1416 (1257) 4.16e-6	-3.1416 (3142) 6.67e-7
(3)	$\frac{x^2}{4} + 4y^2 - 1$	3.0370 (25) 6.96e-3	3.1356 (63) 1.91e-3	3.1411 (239) 1.46e-4	3.1415 (591) 2.53e-5	3.1416 (2348) 1.44e-6	3.1416 (5864) 2.61e-7
(4)	$17(x^2 + y^2) - 30xy - 8$	3.0370 (24) 3.32e-2	3.1356 (63) 1.89e-3	3.1411 (239) 1.52e-4	3.1415 (591) 2.53e-5	3.1416 (2348) 1.60e-5	3.1416 (5864) 2.57e-7

Table 1 lists the calculation results. The values in the first, second, and third rows for each cell represent the area, number of segments that complete the circuit, and computation error, respectively. The error is calculated by

$$\text{Error} = |\text{Computed value} - \text{analytical value}| / |\text{analytical value}|.$$

As we can confirm, the computed values approach to π as θ_M is decreased. Regarding the number of segments, the values increase as θ_M decreases and they are almost inversely proportional to θ_M . In view of the third rows, the computation errors decrease sharply as θ_M decreases. In particular, for smaller $\theta_M \leq 0.025$, the errors reduce to approximately 1/100 fold as θ_M is shortened to 1/10 fold.

4.2 Cassini ovals

The Cassini oval is described as the locus for which the multiplication of distances from two foci is constant. The specific function in implicit form is:

$$f(\mathbf{x}) = (x^2 + y^2)^2 - 2a^2(x^2 - y^2) + (a^4 - b^4) = 0,$$

where a and b are positive constants. If $a = b$, the shape is referred to as a lemniscate of Bernoulli. Moreover, for $a \leq b$, it is known that the area surrounded by this figure is represented as

$$S = 2b^2 E((a/b)^2), \quad (24)$$

where $E(k)$ is the incomplete elliptic integral of the second kind defined below:

$$E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \varphi} d\varphi.$$

We now compute for $b=1/0.9$ and $1/0.999$ with $a=1$, and summarise the results in Table 2. The values S in the second column represent the areas computed using Eq. (24) directly. The rough trend regarding the accuracy of the areas and number of segments are the same as that of Table 1. The resulting values approach S as θ_M decreases, and the number of segments increases

approximately inversely proportional to θ_M . For smaller $\theta_M \leq 0.025$, the errors decrease in approximate quadratic order as θ_M decreases.

Figures 4 and 5 depict the trajectories for calculating the area of the Cassini ovals for $(a, b) = (1, 1/0.9)$ and $(1, 1/0.999)$, respectively. Each figure has two trajectories for $\theta_M = 0.25$ and 0.01 . Starting from the initial point located at the upper right corner, the control point circulates in counter clockwise along the contour.

In Figure 4, even if we set $\theta_M = 0.25$, the trajectory does not deviate markedly from the contour, and the computation result has high accuracy in spite of the large step size. In contrast, Figure 5 has a narrow neck and sudden curve changes in the vicinity of origin; the trajectory deviates from the contour where the curvature significantly changes. This would yield a lower precision result compared with the case in Figure 4.

With these results in mind, in treating general shapes, we need to set θ_M , δl_m , and δl_M properly by making an appropriate compromise between precision and computation load.

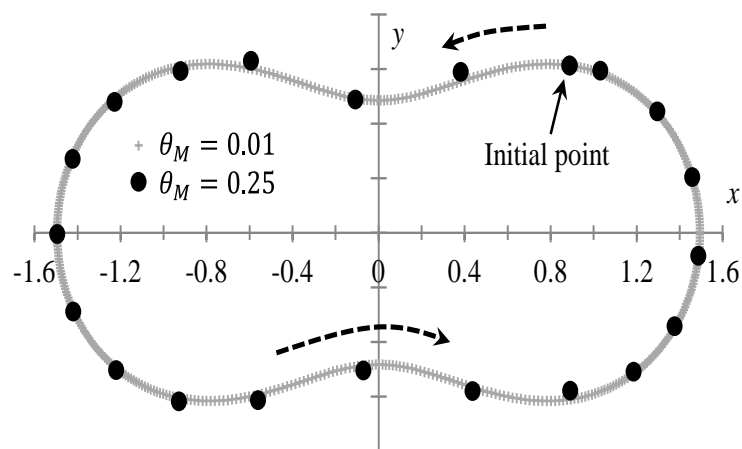


Figure 4: Trajectories for calculating Cassini oval $(a, b) = (1, 1/0.9)$

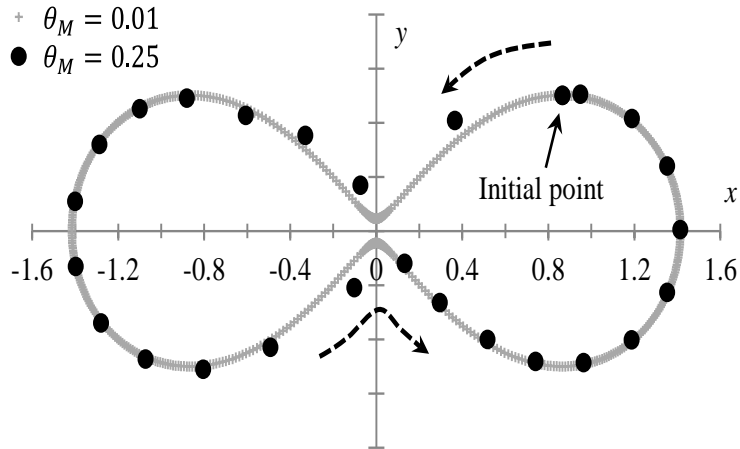


Figure 5: Trajectories for calculating Cassini oval $(a, b) = (1, 1/0.999)$

Table 2: Calculation results for the areas of Cassini ovals with $a = 1$

b	S	$\theta_M = 0.25$	0.1	0.025	0.01	0.0025	0.001
$\frac{1}{0.9}$	3.1290	3.1340	3.1211	2.1288	3.1289	3.1290	3.1290
		(21)	(50)	(193)	(481)	(1917)	(4789)
$\frac{1}{0.999}$	2.0186	1.59e-3	2.51e-3	7.35e-5	3.07e-5	1.91e-6	3.09e-7
		2.0986	2.0136	2.0180	2.0185	2.0186	2.0186
		(26)	(68)	(269)	(663)	(2627)	(6556)
		3.96e-2	2.48e-3	3.23e-4	6.17e-5	2.37e-6	3.75e-7

5 Conclusion

We have developed a lightweight calculation algorithm for the area surrounded by a closed contour. We introduced the notion of a shortcut vector determined by a specified rotation by which an efficient tracing of the contour was accomplished. This approach was referred to as inner skin algorithm.

Through numerical simulations, we confirmed that a compromise between

precision and computation load is necessary. In this article, we have set the parameters for the maximum shortcut angle step size intuitively, and the detailed relationship between computation precision and parameter settings has not been explored. Further study on this issue and precision assurance remains work for the near future.

Acknowledgements. Hiroyuki Goto was supported in part by the Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science No. 23510163.

References

- [1] G. Cottafava and G. L. Moli, Automatic contour map, *Communication of the ACM*, **12**(7), (1969), 386–391.
- [2] D. H. McLain, Drawing contours from arbitrary data points, *The Computer Journal*, **17**(4), (1971), 318–324.
- [3] A. Lopes and K. Brodlie, Accuracy in contour drawing, *Eurographics UK Conference Proceedings*, (1998), 301–311.
- [4] R.-A. Osorio, and K. Brodlie, Contouring with uncertainty, *Proceedings of the 6th Theory & Practice of Computer Graphics Conference*, Manchester, (2008), 59–66.
- [5] Z. Li, , Q. Zhu and C. Gold, *Digital terrain modeling*, CRC Press, New York, 2004.
- [6] L. de Floriani, and M. Spagnuolo (Eds.), *Shape analysis and structuring*, Springer, New York, 2010.
- [7] C.-T. Gold, T. D. Charters and J. Ramsden, Automated contour mapping using triangular element data structures and an interpolant over each irregular

- triangular domain, Proceedings of the ACM SIGGRAPH Computer Graphics, San Jose, (1977), 170–175.
- [8] D. P., Dobkin, S.-F. Levy, and W. P. Thurston, Contour tracing by piecewise linear approximations, ACM Transactions on Graphics, **9**(4), (1990), 389–423.
- [9] J. Li, and T. Ai, A triangulated model for detection of spatial characteristics of GIS data, Proceedings of the 25th International Cartographic Conference, Paris, (2010), 155–159.
- [10] M. Iri, Y. Shimakawa and T. Nagai, Extraction of invariants from digital elevation data with application to terrain topography, Nonlinear Analysis, **47**(8), (2001), 5585–5598.
- [11] S. J. Colley, Vector Calculus, Addison Wesley, Boston, 2011.
- [12] D. G. Luenberger, Optimization by vector space methods, Wiley-Interscience, New Jersey, 1997.