MATHEMATICS OF COMPUTATION, VOLUME 29, NUMBER 130
APRIL 1975, PAGES 554–558

An Algorithm for Winding Numbers for Closed Polygonal Paths

By Kenneth O. Leland*

Abstract. A winding number algorithm for closed polygonal paths (not necessarily simple) based on the notion of counting the number of oriented "signed cuts" of the negative x axis by the path is given. The algorithm is justified by a theory of integer-valued analogues of the complex log function. The algorithm is much simpler than those of J. V. Petty [this journal, v. 27, 1973, pp. 333–337] and H. R. P. Ferguson [*Notices Amer. Math. Soc.*, v. 20, 1973, p. A–211] and leads to faster computation.

1. Introduction. J. V. Petty [3] and H. R. P. Ferguson [1] have given computer implemented algorithms for computing winding numbers, based upon complex analysis results and which do not involve inverse trigonometric functions or integral approximation techniques. Petty's algorithm does not involve division and Ferguson's can be altered slightly to eliminate division. A theory of integer-valued analogues of the complex analytic log function is developed in [2] which yields two quite different approaches to winding numbers and degree theory in general. The simpler one, reported here, cannot be generalized to the higher-dimensional case. The algorithm based on it is much simpler than those of Petty and Ferguson and leads to faster computation. Essentially, the algorithm traverses the path noting the number of times the path cuts the negative x axis, $(-\infty, 0)$, assigning each such cut a plus or minus sign, depending on whether the cut was from above the axis to below or vice versa. The sum of these "signed cuts" is the winding number given by complex function theory.

2. The Theory. Let P be a continuous function on a real closed interval $[a, b]$ into the complex plane C. P is called a path. P is said to be closed if $P(a) = P(b)$. Suppose P is closed and piecewise continuously differentiable. Then, classically, for z_0 a point of C not lying in $P^* = \text{range}(P) = \{P(t) | a \leqslant t \leqslant b\}$, the winding number $w_P(z_0)$ of P about z_0 is defined as

Received September 28, 1973.
AMS (MOS) subject classifications (1970). Primary 30–04, 65E05; Secondary 30A90.
Key words and phrases. Computation of winding numbers, closed polygonal paths, turning point string, computer program, integer-valued log function analogues, topological analysis, signed cut.

* This research performed while the author was a National Research Council senior resident research associate.

$$(2\pi i)^{-1}\int_{P^*}(z - z_0)^{-1}\,dz = (2\pi i)^{-1}\int_a^b (P(t) - z_0)^{-1}\,dP(t).$$

If for $x \in [a, b]$ one sets

$$f(x) = \int_a^x (P(t) - z_0)^{-1} dP(t) + c_0,$$

where c_0 is a suitable constant determined by $P(a)$, then we observe that $e^{f(x)} = P(x)$ for $x \in [a, b]$ and

$$2\pi i\, w_P(z_0) = f(b) - f(a).$$

Whyburn [4] and others have made use of such "log" functions to define winding numbers. This equivalent definition, used in this paper, obviates the need to employ integrals or impose differentiability requirements on P.

For $z, w \in C$, let $[z, w]$ denote the *directed* closed interval of C, $\{(1 - t)z + tw | 0 \leqslant t \leqslant 1\}$. Let $Z = \langle z_0, z_1, \cdots, z_n\rangle$ be a finite sequence of points of C. Then P is called a polygonal path with (turning point) string Z, if there exists a subdivision $a = x_0 < \cdots < x_n = b$ of $[a, b]$ compatible with P, that is, $P(x_k) = z_k$ and $P([x_j, x_{j+1}]) = [x_j, x_{j+1}]$ for all $k = 0, 1, \cdots, n$ and $j = 0, 1, \cdots, n - 1$.

For $z \in C$, set $h(z) = 0$ if $\mathrm{Im}(z) \geqslant 0$ and set $h(z) = 1$ if $\mathrm{Im}(z) < 0$. Then h is the characteristic function of the lower open half-plane.

THEOREM 1. *Let P be a closed polygonal path with string $\langle z_0, \cdots, z_n\rangle$ such that $0 \notin P^*$. For $k = 0, 1, \cdots, n - 1$, set*:

(1) $s_k = 0$ *if* $[z_k, z_{k+1}]$ *contains no point of* $(-\infty, 0)$.

(2) $s_k = h(z_{k+1}) - h(z_k)$ *if* $[z_k, z_{k+1}]$ *contains a point of* $(-\infty, 0)$. *Then*

$$w_P(0) = \sum_{j=0}^{n-1} s_k.$$

We observe that if $s_k = 1$, *then* $[z_k, z_{k+1}]$ *cuts* $(-\infty, 0)$ *from above to below, etc.*

Proof. We require a suitable "branch" g of the analytic log function. We can choose g on $C_0 = C - \{0\}$ so that:

(1) $e^{g(z)} = z$ for $z \in C_0$.

(2) g is continuous on $C_0 - (-\infty, 0)$.

(3) For $c \in (-\infty, 0)$,

(a) $$g(c) = \lim_{z \to c, \mathrm{Im}(z) \geqslant 0} g(z) = \log|c| + i\pi,$$

(b) $$\lim_{z \to c, \mathrm{Im}(z) < 0} g(z) = \log|c| - i\pi.$$

Clearly, $g(z) + 2\pi i \cdot h(z)$ is continuous on $C_0 - (0, +\infty)$.

For $x \in [x_k, x_{k+1})$, $k = 0, 1, \cdots, n - 1$, set

$$u(x) = g \circ P(x) + 2\pi i \left\{ |s_k| \cdot \left[h \circ P(x) - h \circ P(x_k) + \sum_{j=0}^{k-1} s_j \right] \right\},$$

where $g \circ P(x) = g(P(x))$ for $x \in [a, b]$ and $\Sigma_{j=0}^{-1} s_j = 0$, and set

$$u(b) = g \circ P(b) + 2\pi i \sum_{j=0}^{n-1} s_k.$$

Now $e^{u(x)} = e^{g \circ P(x)} = P(x)$ for $x \in [a, b]$ and

$$u(b) - u(a) = \left[g \circ P(b) + 2\pi i \sum_{j=0}^{n-1} s_j \right] - g \circ P(a) = 2\pi i \sum_{j=0}^{n-1} s_j,$$

since $P(a) = P(b)$.

Now if we show that u is continuous, then, from the definition of winding number, we would have

$$2\pi i w_P(0) = u(b) - u(a) = 2\pi i \sum_{j=0}^{n-1} s_j.$$

Let $k = 0, 1, \cdots, n-1$. Now if $[z_k, z_{k+1}]$ contains a point of $(-\infty, 0)$, then for all $x \in [x_k, x_{k+1}]$ and some $c_k \in C$, we have $P(x) \in [z_k, z_{k+1}] \subseteq C_0 - (0, +\infty)$ and

$$u(x) = (2\pi i \cdot h + g) \circ P(x) + c_k,$$

and thus u is continuous on $[x_k, x_{k+1})$. Similarly, if $[z_k, z_{k+1}]$ contains no point of $(-\infty, 0)$, then for all $x \in [x_k, x_{k+1})$ and some $c_k \in C$, $P(x) \in C_0 - (-\infty, 0)$ and

$$u(x) = g \circ P(x) + c_k,$$

and thus u is continuous on $[x_k, x_{k+1})$. Thus u is continuous on $[a, b]$ and the theorem follows.

We observe that if we set $u_0(x) \equiv [u(x) - g \circ P(x)](2\pi i)^{-1}$, then u_0 is integer-valued and u_0 may be regarded as an integer-valued analogue of the function u derived from the analytic theory.

To implement Theorem 1 as an algorithm, we need the following lemma.

LEMMA 2. *Let $A = a + bi$, $B = c + di \in C$ such that $b < 0$ and $d \geqslant 0$ or $d < 0$ and $b \geqslant 0$ and set*

$$Up = +1 \quad \text{if } d < b \quad \text{and} \quad Up = -1 \quad \text{if } b < d$$

and set Det $= bc - ad$. *Then*

(1) $0 \in [A, B]$ *if and only if* Det $= 0$.

(2) $[A, B]$ *contains a point of* $(-\infty, 0)$ *if and only if* Det $\cdot Up < 0$.

Proof. For some unique $t \in [0, 1]$, we have that $D = (1 - t)A + tB = A + t(B - A)$ lies on the x axis. Then

$$0 = \text{Im}(D) = b + t(d - b) \quad \text{and} \quad t = b/(b - d),$$

and thus $D = \text{Real}(D) = a + t(c - a) = (-da + bc)/(b - d) = \text{Det}/(b - d)$. Thus $0 \in [a, b]$ if and only if $D = 0$ and $D = 0$ if and only if Det = 0. Moreover, $D \in (-\infty, 0)$ if and only if $\text{Det}/(b - d) < 0$ and the latter holds if and only if $\text{Det} \cdot Up < 0$.

3. The Algorithm. We are given a closed polygonal path P with string $\langle z_0, \cdots, z_n \rangle$ and a point Q of C. For $k = 0, 1, \cdots, n$, set

$$a_k + b_k \imath = z_k - Q.$$

The algorithm consists of going through the following steps for each $k = 0, 1, \cdots, n - 1$ and either finding that $Q \in [z_k, z_{k+1}] \subseteq P^*$ or computing s_k; and then (if $Q \notin P^*$) forming the sum $\Sigma_{k=0}^{n-1} s_k$ which is the desired winding number $w_P(Q)$.

Step 1. Set $Up = +1$ if $b_k < 0$ and $b_{k+1} \geqslant 0$, and set $Up = -1$ if $b_k < 0$ and $b_{k+1} \geqslant 0$, and set $Up = 0$ if neither case holds. If $Up = 0$, go to Step 2, and if $Up \neq 0$, go to Step 3.

Step 2. If $b_k = b_{k+1} = 0$ and $0 \in [a_k, a_{k+1}]$, then $Q \in P^*$ and the process is terminated. Otherwise, set $s_k = 0$ and start the process again at Step 1 with $k + 1$.

Step 3. Set $\text{Det} = a_{k+1}b_k - a_k b_{k+1}$. If Det = 0, $Q \in [z_k, z_{k+1}] \subseteq P^*$ and the process is terminated. If $Up \cdot \text{Det} > 0$, $[z_k, z_{k+1}]$ contains no point of $(-\infty, 0)$ and we set $s_k = 0$. If $Up \cdot \text{Det} < 0$, then $[z_k, z_{k+1}]$ contains a point of $(-\infty, 0)$ and we set $s_k = -Up$.

We then return to Step 1 and start the process again with $k + 1$.

4. The Computer Program. A CDC FORTRAN EXTENDED program based on the algorithm has been tested on a CDC 6600 computer for several examples. The program was compiled at the most commonly used level of optimization and at the highest level of optimization at the installation of the author. (Improvement attained by using the highest level was negligible.) A typical test example, Case 1, consisted of a closed polygonal path P with a turning point string of 37 points and 27,200 points for which the program computed the winding numbers or determined that the points were on P^* and the algorithm was not required to be carried out fully. Case 1 required 10.66 seconds of central processor time.

For comparison purposes, programs supplied to the author for Petty's and Ferguson's algorithms were compiled and tested on the same examples. Ferguson's program was slightly altered to convert it to integer format. The author's program on the average ran 1.4 times faster than Petty's and five to six times faster than Ferguson's. In particular, for Case 1, Petty's and Ferguson's programs required 14.07

seconds and 56.01 seconds, respectively.

The author was given the results of a comparison of his and Petty's programs run on an IBM 360/65 computer. On this machine, on the average, the author's program ran twice as fast as Petty's. In particular, for Case 1, the times were 9.47 seconds versus 19.75 seconds. Thus, in general, the author's program is 1.4 to 2.0 times faster than Petty's depending on the choice of machine.

The author's algorithm is sensitive to the number of cuts of the x axis made by the path, whereas Petty's algorithm is sensitive to the number of cuts of both the x and y axis, especially the y axis. If examples are deliberately constructed to maximize cuts of the x axis and minimize cuts of the y axis, it is possible to make Petty's program run much faster than the author's. In general, however, if a randomly constructed example is rotated ninety degrees, or the x and y axis are interchanged, the running time for both programs is little changed. Ferguson's algorithm is equally sensitive to cuts of the x axis and of the y axis. Rotation, etc. of any of the examples tested had negligible effect on the running time of Ferguson's program.

On an IBM 360/65 [3], Petty's program was seven to ten times as fast as a program based upon evaluating a line integral by using antidifferentiation.

All of the points, including the turning points, were Gaussian integers and all computations were done in integer format. Case 1 ran one-half second slower when handled by the author's program written in floating-point format.

Acknowledgment. The author would like to thank the referee for his suggestions on comparing the speeds of the algorithms discussed in this paper and for test data (run on an IBM 360/65) he provided.

Aerospace Research Laboratories (LB)
Building 450
Wright-Patterson AFB, Ohio 45433

1. H. R. P. FERGUSON, *Point in Polygon Algorithms: A Critical Element in Urban Data Systems*, Urban Data Center, Univ. of Washington, NTIS PB219-671; *Notices Amer. Math. Soc.*, v. 20, 1973. Abstract #701-68-2.

2. K. O. LELAND, "Computer generated winding numbers and integer valued analogues of the log function" (Preliminary Report), *Notices Amer. Math. Soc.*, v. 20, 1973. Abstract #73T-B177.

3. J. V. PETTY, "A winding number algorithm for closed polygonal paths," *Math. Comp.*, v. 27, 1973, pp. 333–337.

4. G. T. WHYBURN, *Topological Analysis*, 2nd. rev. ed., Princeton Math. Series, no. 23, Princeton Univ. Press, Princeton, N. J., 1964. MR 29 #2758.