MATHEMATICS OF COMPUTATION, VOLUME 27, NUMBER 122, APRIL, 1973

A Winding Number Algorithm for Closed Polygonal Paths

By J. V. Petty

Abstract. A winding number algorithm for closed polygonal paths (not necessarily simple) is derived using classical complex analysis results and techniques. The algorithm is designed specifically to handle large cases efficiently. The performance of a computer program based on the algorithm is discussed and compared with the performance of a computer program which obtains the winding number directly by antidifferentiation.

1. Introduction. The algorithm does not involve the division operation, inverse trigonometric functions, or integral approximation techniques, making it quite suitable for computer programs which must process any combination of many polygonal paths, polygonal paths with many sides, and/or compute many winding numbers. In addition, if all the complex numbers in a given application are Gaussian integers, then a computer program based on the algorithm can be written completely in fixed point mode. The algorithm has proven to be computationally efficient. Results of efficiency tests of a FORTRAN program based on the algorithm are given in the final section.

To avoid ambiguity, we define briefly the mathematical terminology which is used. By a *curve*, we mean a continuous function C from a closed real interval $[a, b]$ (called the *parameter interval*) into the complex plane. $C(a)$ is called the *initial point*, $C(b)$ the *terminal point* of C. The *inverse* $\bar{C}$ of C is given by $\bar{C}(t) = C(a + b - t)$, $a \leqq t \leqq b$. C is *closed* provided $C(a) = C(b)$. C^* denotes the range of C; i.e., $C^* = \{C(t) \mid a \leqq t \leqq b\}$. A *path* is a piecewise continuously differentiable curve. If C is a path and $z_0 \notin C^*$, then the *winding number* of z_0, $W_C(z_0)$, with respect to C is given by $2\pi i W_C(z_0) = \int_C (z - z_0)^{-1}\, dz$.

Given complex numbers z and w, the *directed line* from z to w is defined by $[z, w] = \{(1 - t)z + tw \mid 0 \leqq t \leqq 1\}$. The distinction between a closed real interval and a directed line in the complex plane is always clear from context. A path P with parameter interval $[a, b]$ is called a *polygonal path* provided there exists a subdivision $a = t_1 < \cdots < t_N = b$ of $[a, b]$ such that $P([t_{n-1}, t_n]) = [P(t_{n-1}), P(t_n)]$, for each $n = 2, 3, \cdots, N$. The *turn-points* of P are $P(t_n)$, $n = 1, 2, \cdots, N$. For convenience of notation, we identify a polygonal path by its turn-points; $P : p_n = x_n + iy_n$, $n = 1, 2, \cdots, N$, where $p_n = P(t_n)$, for each n.

Received March 13, 1972.

AMS (MOS) subject classifications (1970). Primary 65-00, 65D30; Secondary 65E05.

Key words and phrases. Computation of winding numbers, closed polygonal paths, interrogation of algebraic signs, computer program.

2. Mathematical Basis for the Algorithm. In this section, we consider the proposition which enables us to define the algorithm so that it employs primarily the interrogation of the signs of the real and imaginary parts of the turn-points of a given closed polygonal path. We recall that the winding number is invariant under translation of the coordinate axes. In particular, if we wish to find $W_C(z)$, for a given closed path C, we may translate the origin to z. Hence, without loss of generality, we always assume that the origin has been translated to the winding point and consider only $W_C(0)$. In order to emphasize the conditions from which the algorithm is derived, we make the following

Definition. Let $P: p_n = x_n + iy_n$, $n = 1, 2, \cdots, N$, be a closed polygonal path with $0 \notin P^*$. For each $n = 1, 2, \cdots, N-1$, let T_n be the point set consisting of the interior and boundary of the triangle determined by $p_n, x_{n+1} + iy_n, p_{n+1}$. Now, define integers K_n, L_n, for $n = 1, 2, \cdots, N-1$, as follows. If $0 \notin T_n$, then $K_n = 0$ and $L_n = 0$. If $0 \in T_n$, then

(2.1) $x_{n+1} = 0$ and $y_n = 0$ imply $L_n = 0$ and $K_n = 1$, provided x_n and y_{n+1} have the same sign, or $K_n = -1$, provided x_n and y_{n+1} differ in sign;

(2.2) $x_{n+1} = 0$ or $y_n = 0$, but not both, implies $K_n = 0$ and $L_n = 1$, provided x_n and y_{n+1} have the same sign, or $L_n = -1$, provided x_n and y_{n+1} differ in sign;

(2.3) $x_{n+1} \neq 0$ and $y_n \neq 0$ imply $K_n = 0$ and $L_n = 2$, provided x_{n+1} and $y_n - y_{n+1}$ have the same sign, or $L_n = -2$, provided x_{n+1} and $y_n - y_{n+1}$ differ in sign.

In addition, for each $n = 1, 2, \cdots, N-1$, put

(2.4) $I_n = (E(y_{n+1}, x_{n+1}) + E(x_n, y_n)) - (E(x_{n+1}, y_n) + E(y_n, x_{n+1}))$, where $E(a, b) =$ Arctan (a/b), if $b \neq 0$, or $E(a, b) = 0$, if $b = 0$. (Here Arctan indicates the principal branch of the inverse tangent relation.)

PROPOSITION. *Let $P: p_n = x_n + iy_n$, $n = 1, 2, \cdots, N$, be a closed polygonal path with $0 \notin P^*$. Then*

$$i^{-1} \int_P z^{-1}\, dz = \sum_{n=1}^{N-1} (I_n + \tfrac{1}{2}K_n\pi + L_n\pi).$$

Proof. Let $z = x + iy$, $z \neq 0$, $u(x, y) = x(x^2 + y^2)^{-1}$, $v(x, y) = -y(x^2 + y^2)^{-1}$, $F = u\,dy + v\,dx$. Then by [1, 10.10, p. 204], $i^{-1} \int_P z^{-1}\, dz = \int_P F$. Put $S_n = [p_n, p_{n+1}]$, for $n = 1, 2, \cdots, N-1$. Then $\int_P F = \sum_{n=1}^{N-1} \int_{S_n} F$, so it suffices to show that

$$\int_{S_n} F = I_n + \tfrac{1}{2}K_n\pi + L_n\pi,$$

for each $n = 1, 2, \cdots, N-1$.

Fix an arbitrary n, $1 \leqq n \leqq N-1$. If $0 \notin T_n$, put $q_n = x_{n+1} + iy_n$, $A_n = [p_n, q_n]$, $B_n = [q_n, p_{n+1}]$. Now, $\int_{S_n} F = \int_{A_n} F + \int_{B_n} F = I_n$, so we have the desired result, since, by definition, $K_n = 0$, $L_n = 0$, whenever $0 \notin T_n$. Thus, we assume $0 \in T_n$ and consider three cases: (1) $x_{n+1} = 0$ and $y_n = 0$; (2) $x_{n+1} = 0$ or $y_n = 0$, but not both; (3) $x_{n+1} \neq 0$ and $y_n \neq 0$.

Suppose case (1) holds. Now, by [2, Theorem V, p. 437], $\int_{S_n} F = a$, where $|a|$ is the angle subtended at the origin by S_n. Since in this case $x_{n+1} = 0$ and $y_n = 0$, we have $|a| = \frac{1}{2}\pi$. We observe that $0 < a$, if x_n, y_{n+1} have the same sign, or $a < 0$, if x_n, y_{n+1} differ in sign. Thus, K_n is defined properly by (2.1) and we have $\int_{S_n} F = \frac{1}{2}K_n\pi$. Moreover, $x_{n+1} = 0$, $y_n = 0$ imply $I_n = 0$ and the definition $L_n = 0$ given in (2.1) is correct, so the Proposition is satisfied for case (1).

For case (2), first assume $x_{n+1} \neq 0$ and $y_n = 0$. Choose a path C having initial point p_n and terminal point $q_n = x_{n+1} + iy_n$ such that C^* lies in the upper or lower half of the plane if p_{n+1} lies in the upper or lower half of the plane, resp. Also, put $D = [q_n, p_{n+1}]$. Then, we have $\int_{S_n} F = \int_C F + \int_D F$. Now by [2, Theorem V, p. 437], $\int_C F = a$, where $|a| = \pi$, and by antidifferentiation, $\int_D F = I_n$. Clearly, $0 < a$, if x_n, y_{n+1} have the same sign, or $a < 0$, if x_n, y_{n+1} differ in sign. Thus, K_n, L_n are properly defined by (2.2), and we have

$$\int_{S_n} F = I_n + \tfrac{1}{2}K_n\pi + L_n\pi,$$

as required. On the other hand, if $x_{n+1} = 0$ and $y_n \neq 0$, we select a path C having initial point q_n and terminal point p_{n+1} such that C^* lies in the left or right half of the plane if p_n lies in the left or right half of the plane, resp. Let $D = [p_n, q_n]$. The desired result is obtained in a fashion similar to the one used above. Hence, we reach the desired conclusion in either event if case (2) holds.

Finally, suppose $x_{n+1} \neq 0$ and $y_n \neq 0$. Let $q_n = x_{n+1} + iy_n$, $C = [q_n, p_{n+1}]$, $D = [p_n, q_n]$. Now $\int_{S_n} F \neq \int_C F + \int_D F$, since independence of path does not hold. However, if $r_n = x_n + iy_{n+1}$, $A = [p_n, r_n]$, $B = [r_n, p_{n+1}]$, then we have $\int_{S_n} F = \int_A F + \int_B F$. Define a path $G = A + B + \bar{C} + \bar{D}$. Now G is a simple closed path around the origin, so $\int_G F = a$, where $a = 2\pi$ if G is traversed counterclockwise, or $a = -2\pi$ if G is traversed clockwise. We observe that the direction of G can be determined easily: G is traversed counterclockwise if $x_{n+1}, y_n - y_{n+1}$ have the same sign, or is traversed clockwise if $x_{n+1}, y_n - y_{n+1}$ differ in sign. Consequently, we see that, by (2.3), L_n is defined properly, so we have

$$L_n\pi = \int_G F = \int_A F + \int_B F + \int_{\bar{C}} F + \int_{\bar{D}} F.$$

The desired result now follows easily and the proof of the Proposition is complete.

Now, for a given n, $1 \leqq n \leqq N - 1$, observe that $E(y_{n+1}, x_{n+1})$ is a summand in I_n, and $E(x_{n+1}, y_{n+1})$ is a summand in I_{n+1}. Hence, if $x_{n+1} \neq 0$, $y_{n+1} \neq 0$, $M_n = 2\pi^{-1}(E(y_{n+1}, x_{n+1}) + E(x_{n+1}, y_{n+1}))$, then $M_n = 1$, provided x_{n+1}, y_{n+1} have the same sign, or $M_n = -1$, provided x_{n+1}, y_{n+1} differ in sign. If $x_{n+1} = 0$ or $y_{n+1} = 0$, then $M_n = 0$. In addition, $E(x_{n+1}, y_n) + E(y_n, x_{n+1})$ appears in I_n. Thus, if

$$M_n' = 2\pi^{-1}(E(x_{n+1}, y_n) + E(y_n, x_{n+1})),$$

then $M_n' = 1, -1$, or 0, depending on x_{n+1}, y_n in the same fashion. Therefore, we can determine the contribution made to $i^{-1}\int_P z^{-1}\,dz$ by $\sum_{n=1}^{N-1} I_n$ without explicitly using the inverse tangent; i.e., $\sum_{n=1}^{N-1} I_n = \frac{1}{2}\pi \sum_{n=1}^{N-1}(M_n - M_n')$.

Thus, given a closed polygonal path P with $0 \notin P^*$, we have that $W_P(0) = \frac{1}{4}\sum_{n=1}^{N-1}(M_n - M_n' + K_n + 2L_n)$. This follows easily from the Proposition and the above observations.

3. The Algorithm. Let $R: r_n = s_n + it_n$, $n = 1, 2, \cdots, N$, be a closed polygonal path, $a + ib$ a complex number for which we are to compute $W_R(a + ib)$, or determine that $a + ib \in R^*$. For each $n = 1, 2, \cdots, N$, put $x_n = s_n - a$, $y_n = t_n - b$. Then, $P: p_n = x_n + iy_n$, $n = 1, 2, \cdots, N$, is a closed polygonal path and we are concerned

with computing $W_P(0)$, since $W_P(0) = W_R(a + ib)$, or determining that $0 \in P^*$. For each $n = 1, 2, \cdots, N - 1$, let S_n, T_n be defined as in the preceding section.

The algorithm consists of carrying out the four steps listed below for each $n = 1, 2, \cdots, N - 1$, defining K_n, L_n, M_n, M_n' as indicated.

Step 1. Determine if $0 \in S_n$; if so, terminate the process; if not, carry out Step 2 whenever $0 \notin T_n$, or Step 3 whenever $0 \in T_n$.

Step 2. Put $K_n = 0$, $L_n = 0$. Now carry out Step 4.

Step 3. Define $K_n = 0$, 1, or -1 and $L_n = 0$, 1, -1, 2, or -2, according to which of the conditions (2.1), (2.2), or (2.3) is applicable. Now, carry out Step 4, except when (2.1) is applicable; in this case, put $M_n = 0$, $M_n' = 0$ and start the process again at Step 1 with $n + 1$.

Step 4. Put $M_n = 0$, if $x_{n+1} = 0$ or $y_{n+1} = 0$; $M_n = 1$, if x_{n+1}, y_{n+1} are nonzero and have the same sign; $M_n = -1$, if x_{n+1}, y_{n+1} are nonzero and differ in sign. Put $M_n' = 0$, if $x_n = 0$ or $y_{n+1} = 0$; $M_n' = 1$, if x_n, y_{n+1} are nonzero and have the same sign; $M_n' = -1$, if x_n, y_{n+1} are nonzero and differ in sign.

If $0 \notin P^*$, then all the K_n, L_n, M_n, M_n' are defined after going through the algorithm. By the Proposition and the concluding remarks of the preceding section, we have

$$W_P(0) = \tfrac{1}{4} \sum_{n=1}^{N-1} M_n - M_n' + K_n + 2L_n.$$

Since $W_P(0)$ is obtained by division by 4, a power of 2, division can be avoided in a computer program by using a shift of two bits on the sum. (Most optimizing FORTRAN compilers expand shifts as in-line code, as opposed to treating them as subprograms, hence shift operations are very efficient.) Also, it is trivial to determine if $0 \in S_n$, $0 \in T_n$ without using division. Therefore, division need not be employed at all in a computer program based on the algorithm. This is desirable since division is a relatively slow operation in a computer. Moreover, if $a + ib$, r_n, $n = 1, 2, \cdots, N$, are all Gaussian integers, then it is evident that a computer program based on the algorithm can be written completely in fixed-point mode. This too is desirable, since fixed-point operations are faster in a computer than floating-point operations.

4. The Computer Program. A FORTRAN IV program based on the algorithm has been tested for efficiency on several cases. The program was compiled using the highest level of optimization available at the installation used by the author. A typical test case consisted of a closed polygonal path P having 25 turn-points and 22,000 points for which the program computed the winding numbers or determined that points were on P^*. All points (including the turn-points of P) were Gaussian integers. Of the 22,000 points, 5,000 were on P^*, 5,000 were inside P^*, and 12,000 were outside P^*. Hence, on this test case, the program was required to carry out the algorithm completely 17,000 times, but had to carry out the algorithm only partially for the 5,000 points which were on P^*. The program required 21.52 seconds of task time on an IBM 360/65 to process this test case.

As a basis for comparison, a program was written in FORTRAN IV which computed the winding number by evaluating the required integral using antidifferentiation. (The FORTRAN inverse tangent subprogram was employed.) This program (compiled using the highest level of optimization) required 149.54 seconds of task time on the same machine to process the above test case.

Similar results were observed on all other test cases attempted. The program based on the algorithm processed all test cases at least seven times faster than the other program, and in some cases, ten times faster.

Design Automation Department
Texas Instruments, Inc.
Dallas, Texas 75222

1. W. RUDIN, *Real and Complex Analysis*, McGraw-Hill Series in Higher Math., McGraw-Hill, New York, 1966. MR 35 #1420.
2. A. E. TAYLOR, *Advanced Calculus*, Ginn, New York, 1955.