More Efficient Computation of the Complex Error Function

G. P. M. POPPE and C. M. J. WIJERS
Twente University

Gautschi has developed an algorithm that calculates the value of the Faddeeva function $w(z)$ for a given complex number z in the first quadrant, up to 10 significant digits. We show that by modifying the tuning of the algorithm and testing the relative rather than the absolute error we can improve the accuracy of this algorithm to 14 significant digits throughout almost the whole of the complex plane, as well as increase its speed significantly in most of the complex plane. The efficiency of the calculation is further enhanced by using a different approximation in the neighborhood of the origin, where the Gautschi algorithm becomes ineffective. Finally, we develop a criterion to test the reliability of the algorithm's results near the zeros of the function, which occur in the third and fourth quadrants.

Categories and Subject Descriptors: G.1.2 [**Numerical Analysis**]: Approximation—*rational approximation*; G.4 [**Mathematical Software**]: —*algorithm analysis*

General Terms: Algorithms

Additional Key Words and Phrases: Error function of complex argument, recursive computation, Voigt function

1. INTRODUCTION

The complex error function is an important tool in several areas of mathematics and physics (Fresnel integral, Dawson's integral, Voigt function . . .). Often frequent evaluation of this function is necessary, making the use of efficient and accurate algorithms very important. Thus far, Gautschi's method [3, 4] for evaluating the Faddeeva function [1] (closely related to the complex error function) is the most successful, its success resulting from the use of a single algorithm which yields an accuracy of up to 10 significant digits throughout the complex plane. Most of the program libraries contain Gautschi's algorithm (see, e.g., [5]).

In this paper we propose an improved version of Gautschi's algorithm. While retaining the algorithm itself in most of the complex plane, we have managed to increase the accuracy to 14 significant digits or better (in the first quadrant) by testing the relative error rather than the absolute one and by modifying the tuning. Furthermore, we were able to obtain a more efficient algorithm by drastically reducing the number of terms to be computed and by using a different

Authors' Address: Twente University, P.O. Box 217, 7500 AE Enschede, The Netherlands.

algorithm in the neighborhood of the origin, where Gautschi's algorithm becomes quite ineffective. This produces a considerable decrease in computing time.

2. COMPUTATIONAL PROCEDURE

2.1 Gautschi's Algorithm

The purpose of Gautschi's algorithm [3] is to evaluate the function

$$w(z) = e^{-z^2}\text{erfc}(-iz) \tag{2.1}$$

up to d correct decimal digits after the decimal point. Limiting ourselves at present to values of z in the first quadrant Q_1 of the complex plane (see, however, Section 3), two areas can be distinguished, separated by a contour Γ (see Figure 1). The first, the outer region Q (see Figure 2), in which the νth convergent of the Laplace continued fraction (which approximates $w(z)$ asymptotically in this region [4]) yields an accuracy of d decimal places;

$$w(z) \cong \frac{1}{z-}\,\frac{1/2}{z-}\,\frac{1}{z-}\,\cdots\,\frac{(\nu-1)/2}{z}. \tag{2.2}$$

The second, the inner region T = R + S (see Figure 2), in which the Faddeeva function can be evaluated (with the same accuracy) by approximating a truncated Taylor expansion,

$$w(z) \cong \sum_{n=0}^{N} \frac{w^{[n]}(z+ih)}{n!}(-ih)^n, \tag{2.3}$$

where $w^{[n]}(z)$ is the nth derivate of $w(z)$. In this expansion, the increment h as well as N, the number of terms, depend on the argument z at which the function is evaluated.

Now the main point in Gautschi's article is the fact that, as one approaches Γ from within T, h and N decrease until eventually (on the contour itself) both become zero, and the algorithm reduces to the evaluation of the Laplace continued fraction.

In this way, only one numerical recursive method can be used for the two domains since the function $w(z)$ can be approximated everywhere by the quantity $\sigma_N^{[\nu]}(z, h)$,

$$w(z) \cong \sigma_N^{[\nu]}(z, h) = \begin{cases} 2/\sqrt{\pi} \cdot s_{-1}, & h > 0 \\ 2/\sqrt{\pi} \cdot r_{-1}, & h = 0 \end{cases} \tag{2.4}$$

where

$$\left.\begin{aligned} r_\nu &= 0, \qquad s_N = 0 \\ r_{n-1} &= \frac{1/2}{h - iz + (n+1)r_n} \\ s_{n-1} &= r_{n-1}[(2h)^n + s_n], \qquad \text{if} \quad n \le N \end{aligned}\right\} \quad n = \nu, \quad \nu - 1, \ldots, 0. \tag{2.5}$$

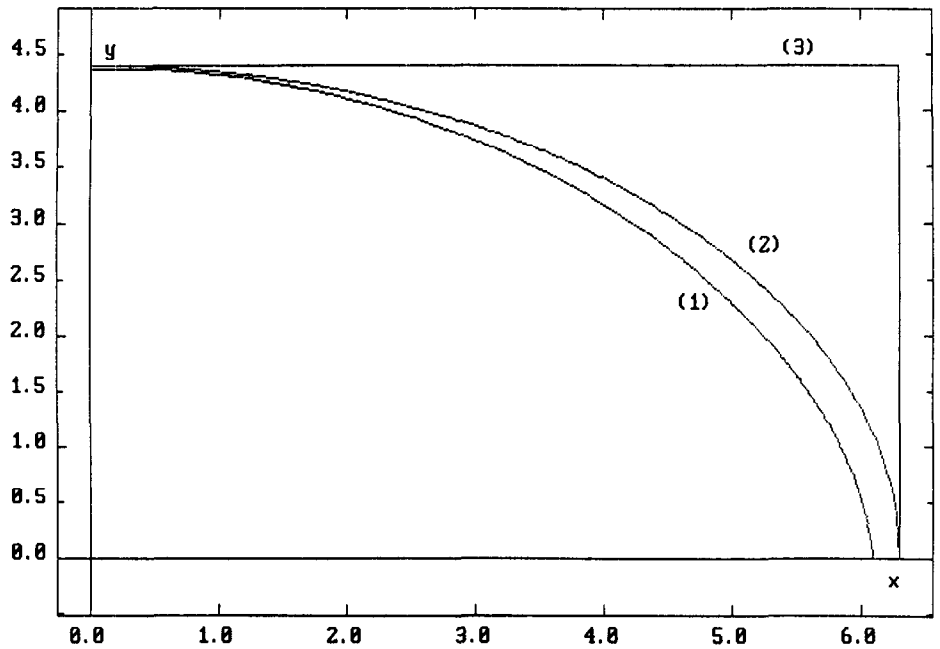

Fig. 1. Curve (1) bounds the region where 16 iterations of the Laplace continued fraction approximate $w(z)$ up to 14 significant digits. Curve (2) is the contour Γ used in this article. Curve (3) is a contour as used by Gautschi.

The only problem that remains is how to define the contour Γ, the value of ν in Q_1 and that of h and N in T, such that

$$\frac{|\sigma_N^{[\nu]}(z, h) - w(z)|}{|w(z)|} \leq \frac{1}{2} 10^{-d}, \tag{2.6}$$

where d is determined by the required accuracy.

2.2 Defining the Contour Γ

First of all we have to define the contour Γ which divides the inner and outer regions. As the Laplace continued fraction converges to the value of $w(z)$, it is clear that this contour will depend on d (the larger d, the larger T), ν (the larger ν, the smaller T), and on the decision whether one adopts an absolute or relative error criterion. As we want to ensure that the algorithm generates $w(z)$ to at least 14 significant digits in Q_1 (so $d = 14$, in our case), we choose to test on the relative error. The remaining parameter ν will be defined in order to minimize the inner area. One finds that at the value $\nu = 17$, the increase in computing time (increasing ν by 1 requires 11 additional floating-point operations) is no longer compensated by decrease of the inner region (where the evaluation of $w(z)$ requires considerably more computing operations than in the outer region, as can be seen from (2.4), (2.5)). We choose thus ν to be 16 on the contour.

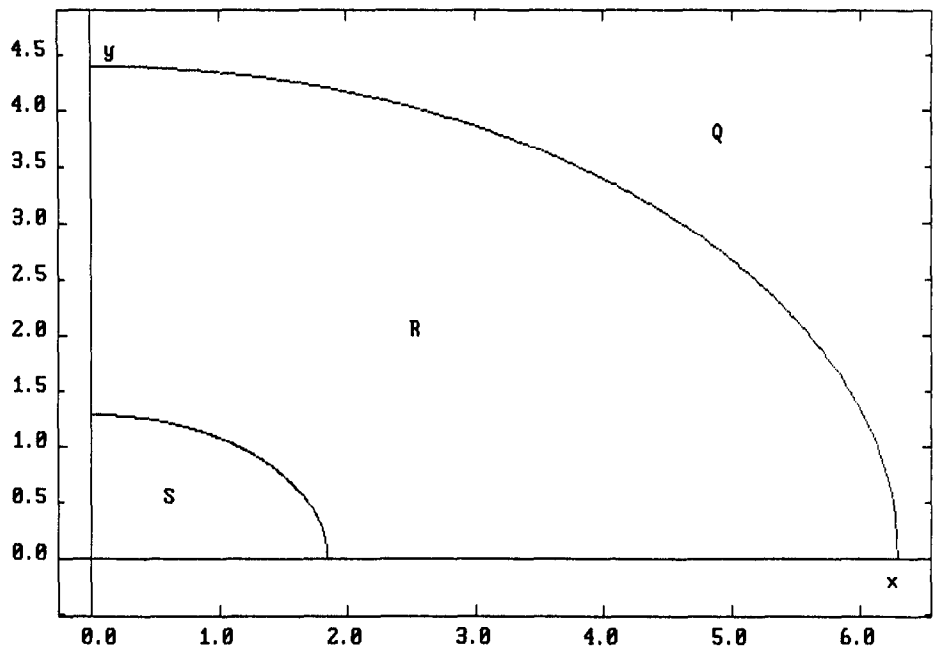

Fig. 2. In the region Q of the first quadrant of the complex plane, the Faddeeva function is approximated by the continued fraction (2.2). In the region R, this is done by the truncated Taylor expansion (2.3). In S, we find the value of the Faddeeva function by evaluating the series (2.15). The contour which separates the region S from R is Σ. The one which separates R from Q is Γ.

The contour Γ itself will be defined (differently from [4]) by the condition that points (x, y) on Γ have to obey the relation,

$$\rho(z) = \sqrt{\left(\frac{x}{x_0}\right)^2 + \left(\frac{y}{y_0}\right)^2} = 1, \qquad \text{with} \quad \begin{matrix} x_0 = 6.3 \\ y_0 = 4.4 \end{matrix}, \tag{2.7}$$

outside of which the continued fraction attains the required accuracy. This shape of Γ diminishes the area of T, compared to a rectangular region (as in [4]) by about 21 percent (see Figure 1).

2.3 The Outer Region Q

In Gautschi's article [4], ν in the outer region is chosen to be a constant. With his tuning, the optimal value of ν depends on the relative frequency with which the procedure is used in the inner and outer region of the first quadrant of the complex plane. Indeed, if a large value for ν in Q is chosen, the area of T is reduced but at the expense of having to compute a large number of iterations for every z in Q. If, on the other hand, ν is kept small, a large inner region is obtained where, as stated earlier, considerably more operations have to be performed to evaluate the Faddeeva function. So it becomes necessary to look for a reasonable compromise between those two effects, as long as ν is kept constant in the outer area.

This problem disappears however if we make $\nu_{\min}(z)$, the smallest integer satisfying (2.6) (with $N = h = 0$ as, according to Gautschi, should be the case in Q), a function of z in the outer region. Indeed, in this way we minimize the computing time in this region and at the same time we can reduce the area of T as much as possible, thus improving the speed of the algorithm in a twofold way.

Now it is sufficient to find a simple rational function to approach the value $\nu_{\min}(z)$ as close as possible for every z in this region. We found as a satisfactory solution the function,

$$\nu(z) = \left[3 + \frac{1442}{26\rho(z) + 77}\right], \qquad z \in Q \tag{2.8}$$

where the [ ] denotes that the integer part of the expression is to be used. This function approximates the value of $\nu_{\min}(z)$, found by explicit calculation, very closely. At most, ν obtained from (2.8) exceeds the true value $\nu_{\min}(z)$ by only 4 (see Figure 3).

2.4 The Inner Region R

The problem in the inner region is how to determine $h(z)$, $N(z)$ and $\nu(z)$ in such a way that $h = N = 0$ and $\nu = 16$ for z on the contour Γ, having at the same time values of $N(z)$ and $\nu(z)$ as small as possible throughout T. Similarly to Gautschi [4], we will set up h, N and ν tentatively in the form

$$h = h_0 s(z), \qquad N = \{N_1 + N_2 s(z)\}, \qquad \nu = \{\nu_1 + \nu_2 s(z)\} \tag{2.9}$$

for $z \in R$, where h_0, N_1, N_2, ν_1, ν_2 and $s(z)$ remain to be determined. The { } here denotes that the integer closest to the argument has to be used.

First of all we will determine h_0 so as to minimize machine time, using the gauging routine of Gautschi [4] (which, given z, d and h, returns nearly optimal values of N and ν; i.e., values which give a $\sigma_N^{[\nu]}(z, h)$ compatible with (2.6)) with the exception that we will, again, test the relative rather than the absolute error (but since $|z| \approx 1$ in this region, this should not make too much difference). For this we need, of course, $s(z)$, but since we have to test only in the neighborhood of $\mathrm{Im}(z) \to 0$ (where the algorithm is most sensitive with respect to the value of h) only the x-dependent part of $s(z)$ must be known at this point. Exploratory computations show that this should have the form,

$$s(x) \sim \sqrt{1 - \left(\frac{x}{x_0}\right)^2}, \tag{2.10}$$

Other possible choices yield very large values of N and ν along the x-axis or require too many operations. We then get the optimal value, $h_0 = 1.88$.

Now equation (2.10) has to be extended to the whole complex plane. As (2.10) is the expression for $s(z)$ when $y = 0$, we know that $s(0) = 1$ and furthermore, we want $s(z)$ to equal 0 on the contour Γ (for there, h has to be 0, as required at the beginning of this section). It appears that the expression

$$s(z) = \left(1 - \frac{y}{y_0}\right)\sqrt{1 - \rho^2(z)} \tag{2.11}$$

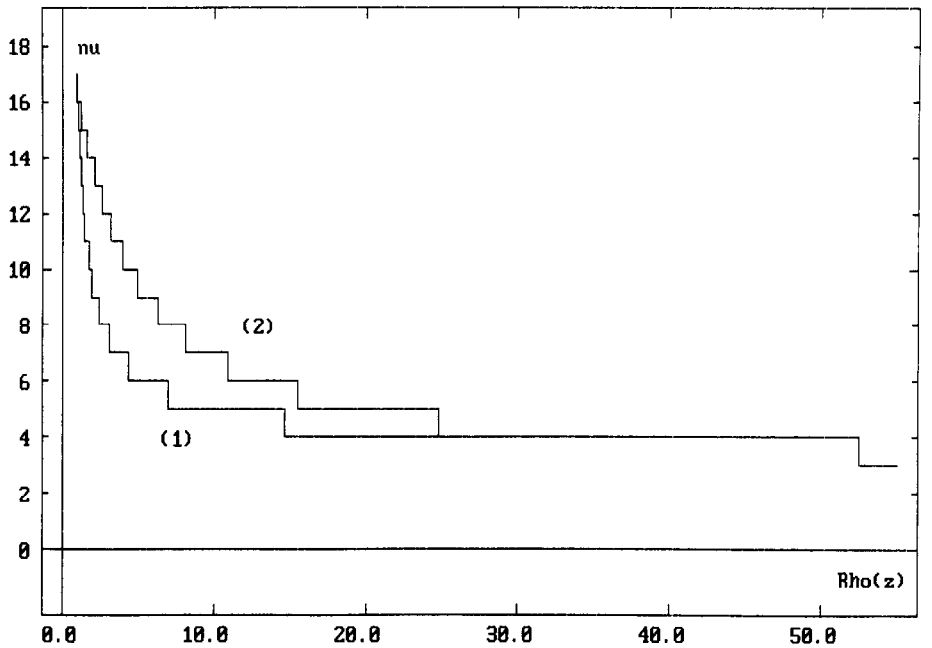

Fig. 3. Curve (1) is the function $\nu_{\min}(z)$. Curve (2) is the function $\nu(z)$ we use in the paper (see (2.8)).

fits these requirements and gives the best (i.e., smallest possible) values of N and ν.

With this knowledge we can obtain the remaining unknown parameters N_1, N_2, ν_1, ν_2 from $s(0) = 1$ and $s(z) = 0$, $z \in \Gamma$. We get

$$\begin{cases} N_1 = 7; & N_2 = 34 \\ \nu_1 = 16; & \nu_2 = 26 \end{cases} \tag{2.12}$$

As these values should at least be equal to those given by the gauging routine, extensive tests were made throughout T, confirming the adequateness of equations (2.11) and (2.12).

Summarizing, we conclude that, for a required accuracy of 14 significant digits in Q_1, the parameters are most optimally tuned in the following way:

$$\left.\begin{aligned} \nu(z) &= \left[3 + \frac{1442}{26\rho(z) + 77}\right] \\ N(z) &= 0 \\ h(z) &= 0 \end{aligned}\right\} \quad \text{for} \quad z \text{ in } Q \tag{2.13}$$

and

$$\left.\begin{aligned} \nu(z) &= \{16 + 26s(z)\} \\ N(z) &= \{7 + 34s(z)\} \\ h(z) &= 1.88s(z) \end{aligned}\right\} \quad \text{for} \quad z \text{ in T.} \tag{2.14}$$

where $s(z)$ is given by (2.11).

2.5 The Region Around the Origin S

As can easily be seen from equation 2.12, the Gautschi algorithm tends to become very inefficient in the neighborhood of the origin. Because of this, it would be convenient to replace the Gautschi algorithm in this region by a series which is more efficient there:

$$w(z) = e^{-z^2}\left(1 + \frac{2i}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{z^{2n+1}}{(2n+1)\,n!}\right). \tag{2.15}$$

The contour Σ (see Figure 2) that separates the region in which this series is used from that in which we use the modified Gautschi algorithm is chosen so as to minimize the computing time.

We found that in the region S of Q_1 determined by

$$0 \le \rho(z) \le 0.292$$

(with $\rho(z)$ given by (2.7)), the series

$$w(z) = e^{-z^2}\left(1 + \frac{2i}{\sqrt{\pi}} \sum_{n=0}^{N} \frac{z^{2n+1}}{(2n+1)\,n!}\right) \tag{2.16}$$

attains a relative accuracy of 14 significant digits for

$$N = \{6 + 72s'(z)\} \tag{2.17}$$

where

$$s'(z) = \left(1 - 0.85\,\frac{y}{y_0}\right)\rho(z) \tag{2.18}$$

and y_0 given by (2.7).

3. THE OTHER QUADRANTS

Until now we have only considered the Faddeeva function in the first quadrant Q_1, where its value is given by (2.1), and where the algorithm is tuned to a relative accuracy of 14 significant digits. The values of $w(z)$ for z in one of the other quadrants can be calculated using

$$w(\bar{z}) = \overline{w(-z)}, \tag{3.1}$$

$$w(-z) = 2e^{-z^2} - w(z). \tag{3.2}$$

As can be seen from (3.2), the value of $w(z)$ in the 3rd and 4th quadrant (Q_3 and Q_4) is obtained by subtraction of two values. Because of this, loss of accuracy will occur in the lower half of the complex plane, a loss that becomes significant in the neighborhood of the zeros of $w(z)$. A table of the first 100 of these zeros has been published in Fettis [2].

As the absolute error (A.E.) of $w(z)$ in the 3rd quadrant is bounded by

$$\text{A.E.}(w(-z)) \le \tfrac{1}{2} \cdot 10^{-14} \sqrt{|\,w(z)\,|^2 + |\,2e^{-z^2}\,|^2}, \qquad z \in Q_1, \tag{3.3}$$

(both $w(z)$ and e^{-z^2} have a relative error (R.E.) $\leq \frac{1}{2} \cdot 10^{-14}$) and $|w(z)|$ has to equal $|e^{-z^2}|$ at each of its zeros, we get for z near one of the zeros:

$$\text{A.E.}(w(-z)) \approx \tfrac{1}{2} \cdot 10^{-14} \sqrt{2\,|w(z)|^2} \leq \tfrac{1}{2} \cdot 10^{-14} \cdot \sqrt{2} \tag{3.4}$$

Furthermore, in a small region around z_i, one of the zeros, we can approximate $w(z)$ linearly:

$$w(z_i + dz) \approx w(z_i) + w'(z_i) \cdot dz. \tag{3.5}$$

From [1, p. 9] we find that $w'(z) = 2i/\sqrt{\pi} - 2zw(z)$, so

$$w'(z_i) = \frac{2i}{\sqrt{\pi}}, \tag{3.6}$$

and

$$w(z_i + dz) \approx \frac{2i}{\sqrt{\pi}} \cdot dz. \tag{3.7}$$

Combining (3.4) and (3.7) yields

$$|\,\text{R.E.}(w(z_i + dz))\,| \approx \frac{0.63\,10^{-14}}{|\,dz\,|}. \tag{3.8}$$

From this we see that the algorithm, used in Q_3 and Q_4 to calculate a single value of $w(z)$, yields an accuracy of about 13 significant digits outside a circular region with radius $|\,dz\,| \sim 0.126$ around a zero of the function. Inside that region, we have a constant A.E. (see (3.4)) and a rapidly increasing R.E. as $|\,dz\,| \Rightarrow 0$ (see (3.8)).

Often, Faddeeva's function or functions derived from it are used in a series. In that case the usual square root expression using A.E.'s has to be applied. It depends on the value of the sum of the series whether the negative influence of the zeros on its accuracy is significant or not.

So we conclude that, also in the other quadrants, the algorithm can be used with sufficiently high relative accuracy, except for some small circular regions in Q_3 and Q_4 around the zeros of w, whereas the absolute accuracy of 14 decimal digits remains preserved throughout.

4. TESTS AND CONCLUSION

Since we want a direct test between our algorithm (P) and the original Gautschi algorithm (G), we tuned the latter precisely according to the original article using $d = 14$. This means that for G, use was made of $x_0 = 6.3$ and $y_0 = 4.4$ for defining the rectangular contour (see Figure 1). For the two regions, the tuning parameters in G were:

$$\begin{array}{lll} \text{in} & \text{Q:} & \nu = 16, \\ \text{in} & \text{T:} & h = 1.88, \\ & & N = \{7 + 34s(z)\}, \\ & & \nu = \{16 + 26s(z)\}. \end{array}$$

The performance has been improved in a two-fold way. For points in the region S of the first quadrant the evaluation of the Faddeeva function by P is faster than by G by a factor of 3.9. For each point within R the time of computation for G and P is about the same. However, the P algorithm still has a 28% smaller inner region R. Outside R, already for $\rho = 1.2$ the P is 10% faster than the G algorithm; for $\rho = 7.1$, the P algorithm is 2.2 times faster and at infinity, it runs more than 3 times as fast.

These tests involved the speed of the algorithm. In order to gain further confidence in the accuracy claimed, we have compared the results of algorithm P with those obtained by G, and this for 10^4 points distributed throughout Q_1. We found that the largest deviation for the real part is 6.4×10^{-15} and for the imaginary part 3.1×10^{-14}.

We next tested the algorithm's accuracy by comparing the results with those obtained from independent methods for about 200 arbitrary values of y along the imaginary axis. For $x \in Q_1$ and $0 \leq y \leq 1$ the following series can be used:

$$\operatorname{Erfc}(z) = 1 - \frac{2e^{-z^2}}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(2)^n z^{2n+1}}{1.3 \cdots (2n+1)}, \tag{4.1}$$

and for large values of y ($10 \leq y$) we have the asymptotic expansion:

$$\operatorname{Erfc}(z) \sim \frac{e^{-z^2}}{\sqrt{\pi} z} \left(1 + \sum_{n=1}^{\infty} \frac{(-1)^n 1.3 \cdots (2n-1)}{(2z^2)^n}\right). \tag{4.2}$$

We found that the largest relative deviation, when the results of the P algorithm were compared to those of (4.1), was 2.66×10^{-15}, and with (4.2), it was 3.5×10^{-16}. This suggests that a relative accuracy of 14 decimal digits has indeed been attained.

REFERENCES

1. FADDEEVA, V. N., AND TERENT'EV, N. N. Tables of values of the function $w(z) = e^{-z^2}(1 + 2i/\sqrt{\pi} \int_0^z e^{t^2}\, dt)$ for complex argument. *Gosud. Izdat. Teh.-Teor. Lit.*, Moscow, 1954; English transl., Pergamon Press, New York, 1961.
2. FETTIS, H. E., CASLIN, J. C., AND CRAMER, K. R. Complex zeros of the error function and of the complex error function. *Math Comput. 27* (1973), 401–407.
3. GAUTSCHI, W. Algorithm 363—Complex error function. *Commun. ACM 12* (1969), 635.
4. GAUTSCHI, W. Efficient computation of the complex error function. *SIAM J. Numer. Anal. 7* (1970), 187–198.
5. *IMSL Library (Reference Manual)*, Vol. 3, Chap M. IMSL Inc., June 1982.

Received February 1988; revised January 1989; accepted March 1989