Efficient algorithmic implementation of the Voigt/complex error function based on exponential series approximation

S.M. Abrarov [a,*], B.M. Quine [a,b]

[a] Department of Earth and Space Science and Engineering, York University, 4700 Keele Street, Toronto, Canada M3J 1P3
[b] Department of Physics and Astronomy, York University, 4700 Keele Street, Toronto, Canada M3J 1P3

ARTICLE INFO

ABSTRACT

We show that a Fourier expansion of the exponential multiplier yields an exponential series that can compute high-accuracy values of the complex error function in a rapid algorithm. Numerical error analysis and computational test reveal that with essentially higher accuracy it is as fast as FFT-based Weideman's algorithm at a regular size of the input array and considerably faster at an extended size of the input array. As this exponential series approximation is based only on elementary functions, the algorithm can be implemented utilizing freely available functions from the standard libraries of most programming languages. Due to its simplicity, rapidness, high-accuracy and coverage of the entire complex plane, the algorithm is efficient and practically convenient in numerical methods related to the spectral line broadening and other applications requiring error-function evaluation over extended input arrays.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

The complex probability function is defined by the following integral [4,10]

$$W(z) = \frac{i}{\pi} \int_{-\infty}^{\infty} \frac{\exp\left(-t^2\right)}{z - t} dt,$$

where $z = x + iy$. It is convenient to represent it as a superposition of the real and imaginary parts

$$W(z) = K(x, y) + iL(x, y),$$

where

$$K(x, y) = \frac{y}{\pi} \int_{-\infty}^{\infty} \frac{\exp\left(-t^2\right)}{y^2 + (x - t)^2} dt \tag{1}$$

and

$$L(x, y) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{(x - t) \exp\left(-t^2\right)}{y^2 + (x - t)^2} dt. \tag{2}$$

---

* Corresponding author.
  E-mail address: abrarov@yorku.ca (S.M. Abrarov).

The function $K(x, y)$ is widely applied in radiative transfer as it describes the broadening process in absorption and emission of the atomic and molecular gases on the Earth and solar system planetary atmospheres [19]. It is commonly known as the Voigt function where the input parameters are

$$x = \sqrt{\ln 2}(v - v_0)/\alpha_D$$

and

$$y = \sqrt{\ln 2}\,\alpha_L/\alpha_D,$$

v is the frequency, v_0 is the frequency at the line center, α_D and α_L are the Doppler and Lorentz half widths at half maximum. The complex part $L(x, y)$ is also widely used as it can describe spectral behaviour of the index of refraction in various dispersive materials [5,6].

Applying the Fourier transform, the real (1) and imaginary (2) parts of the complex probability function can be rewritten as [4]

$$K(x, y) = \frac{1}{\sqrt{\pi}} \int_0^\infty \exp\left(-\frac{\tau^2}{4} - y\tau\right) \cos(x\tau)dt, \quad (-\infty < x < \infty, y > 0) \tag{3}$$

and

$$L(x, y) = \frac{1}{\sqrt{\pi}} \int_0^\infty \exp\left(-\frac{\tau^2}{4} - y\tau\right) \sin(x\tau)dt, \quad (-\infty < x < \infty, y > 0), \tag{4}$$

respectively. It was shown earlier that at $y > 0$ the complex probability function can be reduced to complex error function [4,8]

$$w(z) = \exp(-z^2)\mathrm{erfc}(-iz), \tag{5}$$

where $\mathrm{erfc}(\alpha) = 1 - \mathrm{erf}(\alpha)$ is the complimentary error function of complex argument α.

None of the integrals above have analytical solutions. However, the last two integrals (3) and (4) may be advantageous for numerical methods and many algorithms, developed for accurate calculation of the Voigt and complex error functions, are derived on their basis (see e.g. Srivastava–Miller model proposed in Ref. [22]).

The most widely used approximations of the Voigt and complex error functions utilize rational polynomials [10–12,14,21,23,24]. Although rational polynomials provide the accuracy sufficient for most practical tasks, the higher accuracy may sometimes also be required. However, approximations that provide high-accuracy remain problematic since an improvement in accuracy drastically decelerates the computation. This becomes particularly evident when massive input arrays are used for high-spectral resolution in spectral line broadening applications.

While for a moderate spectral resolution few millions computed points may be sufficient, a high spectral resolution requires them up to hundreds of millions or even more than a billion. Therefore it is essential to develop efficient algorithm providing highly accurate as well as rapid results.

In this paper we describe an algorithm based on an exponential series approximation for the rapid and high-accuracy calculation of the Voigt and complex error functions. Compared to the approximations of $K(x, y)$ and $L(x, y)$ functions involving the real [2] and complex [3] rational functions, the proposed exponential series has considerable advantages for algorithmic implementation.

We applied MATLAB 7.9.0 supporting array programming features. A typical desktop computer Intel(R) Quad CPU with RAM 4.00 GB was utilized.

2. Approximation methods

2.1. High-accuracy approximations at smaller y

The problem related to calculation of the Voigt and complex error functions at smaller y is very common perhaps in the most of known approximations. This problem can be resolved for instance by partitioning a complex plane along certain level of y below which, say at $y \leqslant 10^{-3}$, an appropriate approximation can be applied. In this section we will consider two approximations of this kind.

The first approximation of the complex error function can be readily obtained by substituting well-known equation of the error function of complex argument [1,15]

$$\mathrm{erf}(z) \approx \mathrm{erf}(x) + \frac{\exp(-x^2)}{\pi}\left\{\frac{1 - \exp(-2ixy)}{2x} + 2\sum_{n=1}^N \frac{\exp(-n^2/4)}{n^2 + 4x^2}[2x - \exp(-2ixy)(2x\cosh(ny) - in\sinh(ny))]\right\} \tag{6}$$

into identity (5). In order to perform the error analysis, we can apply the following equations

$$\Delta_K = \frac{|K(x,y) - K_{ref.}(x,y)|}{K_{ref.}(x,y)}$$

and

$$\Delta_L = \frac{|L(x,y) - L_{ref.}(x,y)|}{L_{ref.}(x,y)},$$

where Δ_K and Δ_L are the relative errors, $K_{ref.}(x,y)$ and $L_{ref.}(x,y)$ are references for the real and imaginary parts of the complex error function, respectively. As references for $K_{ref.}(x,y)$ and $L_{ref.}(x,y)$ functions we applied Algorithm 680 providing highly accurate results that may be limited by machine accuracy [18]. Despite moderate calculation speed, the accuracy of Algorithm 680 is nearly perfect and in excellent agreement with other results reported recently elsewhere [16,25].

Fig. 1 shows the logarithms of relative errors for the real (upper panel) and imaginary (lower panel) parts of the complex error function obtained by substituting Eq. (6) into (5). The colour bars on the right sides of two panels indicate the order of accuracy. As we can see from the both panels of Fig. 1, at $y < 4$ this approximation works perfectly well; the blue and dark-blue colours in the contour plots indicate the highly accurate numbers that can match up to the last decimal digits with those obtained by Algorithm 680 [18]. However, the accuracy drastically decreases with increasing y. As a result, this approximation is limited from upper side by value $y \approx 5$ indicating an accuracy 10^{-5} that corresponds to orange colour.

Another interesting approximation of the error function of complex argument that can also be obtained by Fourier expansion (see Appendix A) is given by

$$\operatorname{erf}(z) \approx \frac{2t_m}{\pi^{3/2}} \sum_{n=1}^{N} \frac{a_n}{n} \sin\left(\frac{n\pi}{t_m} z\right) + \frac{a_0}{\sqrt{\pi}} z, \tag{7}$$

where

$$a_n \approx \frac{\sqrt{\pi}}{t_m} \exp\left(-\frac{n^2 \pi^2}{4 t_m^2}\right). \tag{8}$$

The approximation (7) has very rapid convergence that can be estimated from Fig. 2. Particularly, at $N = 19$ the difference $\varepsilon(x)$ between "true" error function and its approximation (7) is smaller than 4.5×10^{-14} (the domain $\Delta t/2 = 6$ has been chosen according to Milone et al. [17]). As this approximation is highly accurate for the real argument x, it is reasonable to expect the high-accuracy at smaller values of y as well. Therefore the approximation (7) may also be practically useful for complex error function approximation.

Substituting approximation (7) into identity (5) we get

$$w(x,y) \approx e^{-(x+iy)^2} \left[1 - \frac{2t_m}{\pi^{3/2}} \sum_{n=1}^{N} \frac{a_n}{n} \sin\left(\frac{n\pi}{t_m}(-ix+y)\right) - \frac{a_0}{\sqrt{\pi}}(-ix+y) \right]. \tag{9}$$

The accuracy of the real and imaginary parts of approximation (9) can be seen from Fig. 3 in the upper and lower panels, respectively. At $y \leqslant 4$ the approximation (9) provides the highly accurate results that can be better than 10^{-8}. However,
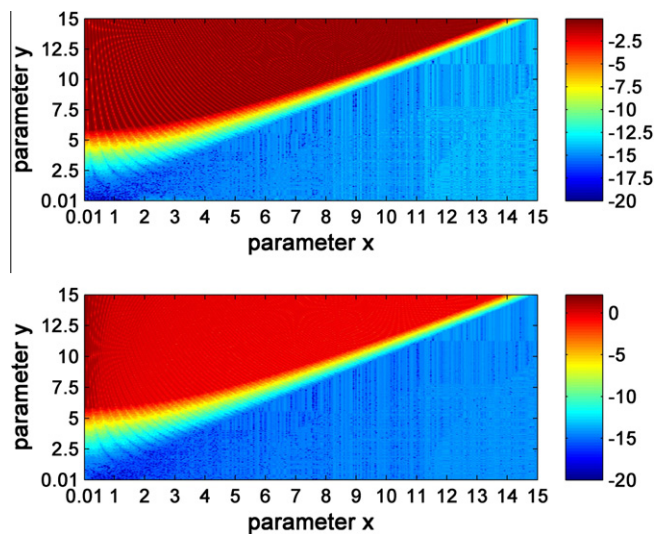

Fig. 1. The contour plots for the real $\log_{10}\Delta_K$ (above) and imaginary $\log_{10}\Delta_L$ (below) parts of the complex error function approximation obtained by substitution of Eq. (6) into identity (5). The considered domain is $0.01 \leqslant x \leqslant 15$ and $0.01 \leqslant y \leqslant 15$. The value of the integer N dependent on x and y is given by Leutenegger, Ref. [15].
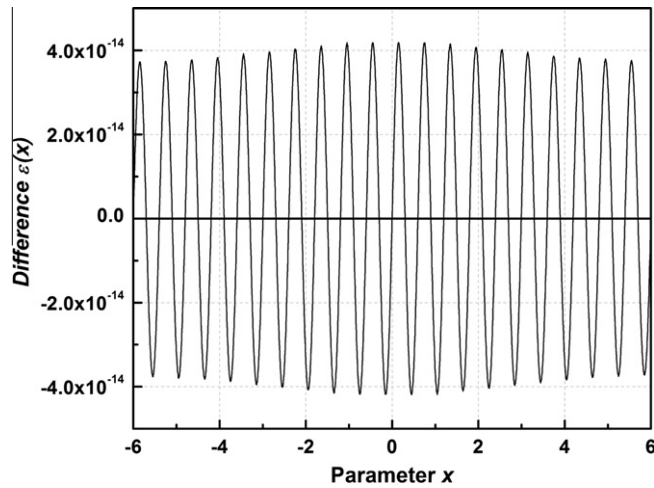
Fig. 2. The difference $\varepsilon(x)$ between error function and error function approximation (7) in domain $-6 \leqslant x \leqslant 6$ at $t_m = 6$ and $N = 19$.
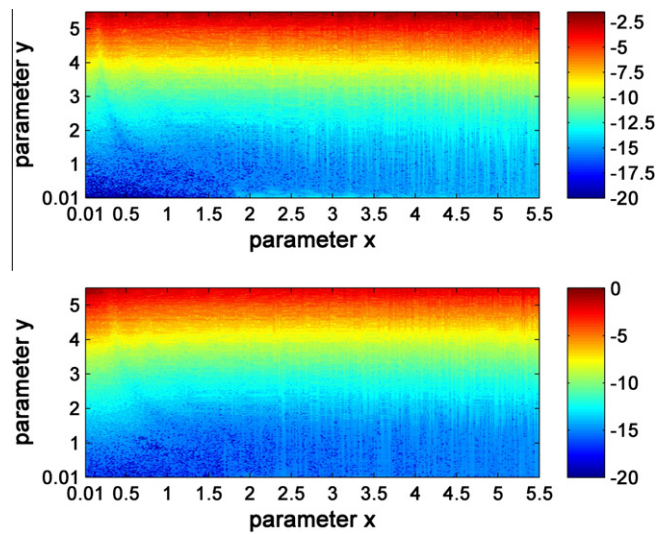

Fig. 3. The contour plots for the real $\log_{10}\Delta_K$ (above) and imaginary $\log_{10}\Delta_L$ (below) parts of complex error function approximation (7) in the domain $0.01 \leqslant x \leqslant 5.5$ and $0.01 \leqslant y \leqslant 5.5$ at $N = 45$.

the accuracy of approximation (9) also drastically decreases with increasing y. Once again taking as criterion an accuracy 10^{-5} (orange colour), the upper boundary of parameter y in approximation (9) can be found as 4.5. Despite the fact that this upper limit of the input parameter y is slightly slower than that of in the previous approximation, the use of Eq. (9) has an advantage in terms of calculation speed.

In Eq. (9) the sine is a "slowest" function in computation. The argument of the sine function depends upon index n. As a result, the sine function has to be determined over and over again N times for summation. However, there is no the need to calculate directly all $\sin(n\alpha)$ terms and acceleration can be achieved by using the identity

$$\sin(n\alpha) \equiv 2\cos\alpha\sin(\alpha(n-1)) - \sin(\alpha(n-2)),$$

where $\alpha = \pi(-ix + y)/t_m$. Hence, computing directly only arrays $\sin(\alpha)$ and $\sin(2\alpha)$, the remaining $N-2$ terms can be determined recursively according to this identity. Such a way, we can gain a significant acceleration in computation of the complex error function.

2.2. Boundary-free approximations

As an alternative to approach above, a boundary-free approximation can be effectively used in order to avoid undesirable division of the complex plane $z = x + iy$ in calculation of the Voigt and complex error functions.

One of the efficient boundary-free algorithms providing highly accurate and rapid computation is the Weideman's algorithm [23] based on the following approximation

$$w(z) \approx \frac{\pi^{-1/2}}{L - iz} + \frac{2}{(L - iz)^2} \sum_{n=0}^{N-1} a_{n+1} \left(\frac{L + iz}{L - iz} \right)^n, \tag{10}$$

where a_n are FFT-based expansion coefficients and $L = 2^{-1/4} N^{1/2}$. Although the corresponding algorithm is slower than the Humlíček algorithm, it provides the accuracy that is better by several orders of magnitude. Nowadays the Weiderman's algorithm is perhaps most rapid among those providing an accuracy better than 10^{-6}.

Another boundary-free approximation of the complex error function can be recast from [3]

$$K(x,y) \approx \frac{1}{2\sqrt{\pi}} \sum_{n=0}^{N} a_n \left[\frac{(in\pi\tau_m + \tau_m^2 y)(1 - e^{-(in\pi + \tau_m y)} \cos(\tau_m x)) + e^{-(in\pi + \tau_m y)} \tau_m^2 x \sin(\tau_m x)}{\tau_m^2 x^2 - (n\pi - i\tau_m y)^2} \right.$$
$$\left. - \frac{(in\pi\tau_m - \tau_m^2 y)(1 - e^{in\pi - \tau_m y} \cos(\tau_m x)) - e^{in\pi - \tau_m y} \tau_m^2 x \sin(\tau_m x)}{\tau_m^2 x^2 - (n\pi + i\tau_m y)^2} \right] - \frac{a_0[y - e^{-\tau_m y}(y\cos(\tau_m x) - x\sin(\tau_m x))]}{2\sqrt{\pi}(x^2 + y^2)} \tag{11}$$

and

$$L(x,y) \approx \frac{1}{2\sqrt{\pi}} \sum_{n=0}^{N} a_n \left[\frac{\tau_m^2 x - e^{-(in\pi + \tau_m y)}(\tau_m^2 x \cos(\tau_m x) + (in\pi\tau_m + \tau_m^2 y) \sin(\tau_m x))}{\tau_m^2 x^2 - (n\pi - i\tau_m y)^2} \right.$$
$$\left. + \frac{\tau_m^2 x - e^{in\pi - \tau_m y}(\tau_m^2 x \cos(\tau_m x) - (in\pi\tau_m - \tau_m^2 y) \sin(\tau_m x))}{\tau_m^2 x^2 - (n\pi + i\tau_m y)^2} \right] - \frac{a_0[x - e^{-\tau_m y}(x\cos(\tau_m x) + y\sin(\tau_m x))]}{2\sqrt{\pi}(x^2 + y^2)}, \tag{12}$$

where the Fourier expansion coefficients are

$$a_n \approx \frac{2\sqrt{\pi}}{\tau_m} \exp\left(-\frac{n^2 \pi^2}{\tau_m^2} \right). \tag{13}$$

In particular, it can be shown that the superposition $K(x,y) + iL(x,y)$ results in a simple representation (see Appendix B):

$$w(z) \approx \frac{i}{2\sqrt{\pi}} \left[\sum_{n=0}^{N} a_n \tau_m \left(\frac{1 - e^{i(n\pi + \tau_m z)}}{n\pi + \tau_m z} - \frac{1 - e^{i(-n\pi + \tau_m z)}}{n\pi - \tau_m z} \right) - a_0 \frac{1 - e^{i\tau_m z}}{z} \right]. \tag{14}$$

In contrast to approximations (11) and (12), the approximation (14) deals only with a single complex argument $z = x + iy$, not with its real x and imaginary y parts separately. This gives some advantages in algorithmic implementation. In addition, applying approximation (14) for the complex error function we can avoid a separate computation of the real and imaginary parts by Eqs. (11) and (12), respectively. Consequently, the use of the single approximation (14) saves time in computation by about factor of 2.

3. Results and discussion

3.1. Error analysis

The accuracy of approximation (14) is in excellent agreement with that provided by Algorithm 680. As we can see from Tables 1 and 2, the numbers generated by Algorithm 680 and approximation (14) within the domain $0.01 \leqslant x \leqslant 15$ and $0.01 \leqslant y \leqslant 15$ may match up to the last decimal digit.

Although the accuracy provided by approximation (10) is very high, it is generally dependent on the parameter y. Since the decrease in accuracy with decreasing parameter y is not drastic, it can be sustained by increasing integer N determining

Table 1
The numbers generated by Algorithm 680 and approximation (14) for the real part of the complex error function (the Voigt function) within the domain $0.01 \leqslant x \leqslant 15$ and $0.01 \leqslant y \leqslant 15$.

| Parameter x	Parameter y	Algorithm 680	Approximation (14)
0.01	0.01	9.887176929549547E−1	9.887176929549550E−1
0.5	0.5	5.331567079121750E−1	5.331567079121748E−1
1	1	3.047442052569128E−1	3.047442052569125E−1
2.5	2.5	1.167371250446503E−1	1.167371250446503E−1
5	5	5.696543988817697E−2	5.696543988817698E−2
7.5	7.5	3.777752935846000E−2	3.777752935845998E−2
10	10	2.827946745423246E−2	2.827946745423245E−2
12.5	12.5	2.260351678541391E−2	2.260351678541391E−2
15	15	1.882714532513676E−2	1.882714532513676E−2

Table 2
The numbers generated by Algorithm 680 and approximation (14) for the imaginary part of the complex error function within the domain $0.01 \leqslant x \leqslant 15$ and $0.01 \leqslant y \leqslant 15$.

Parameter x	Parameter y	Algorithm 680	Approximation (14)
0.01	0.01	1.108529605747726E−2	1.108529605747765E−2
0.5	0.5	2.304882313844584E−1	2.304882313844584E−1
1	1	2.082189382028316E−1	2.082189382028317E−1
2.5	2.5	1.079085859964814E−1	1.079085859964814E−1
5	5	5.583874277539103E−2	5.583874277539103E−2
7.5	7.5	3.744329372959514E−2	3.744329372959511E−2
10	10	2.813843327633690E−2	2.813843327633689E−2
12.5	12.5	2.253130329137736E−2	2.253130329137736E−2
15	15	1.878535427799565E−2	1.878535427799564E−2

the number of the summation terms. In particular, when the integer $N = 24$, the accuracy 10^{-6} is sustained at $y \geqslant 10^{-3}$ whereas to sustain the same accuracy at $y \geqslant 10^{-5}$ requires the integer $N = 32$.

The comparison shows that the approximation (14) is significantly more accurate than approximation (10). Fig. 4 illustrates the relative error contour plots within the domain $0.1 \leqslant x \leqslant 15$ and $10^{-5} \leqslant y \leqslant 10^{-4}$ for the real (left frames) and imaginary (right frames) parts of the approximations (10) and (14), respectively. Comparing Fig. 4a with b one can see that the accuracy of the real part of approximation (10) at $N = 32$ is 10^{-6} while the accuracy of the real part of approximation (14) at
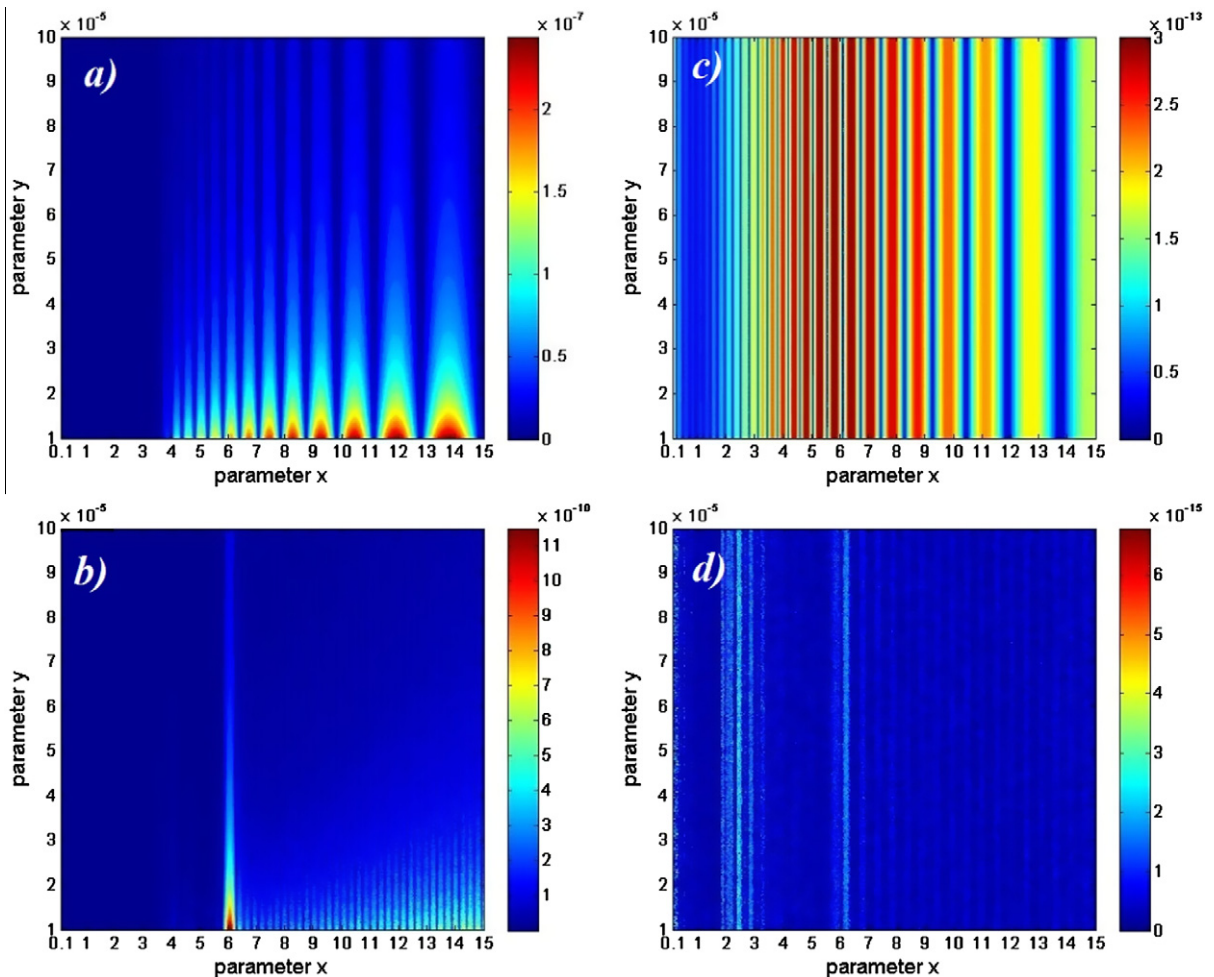
Fig. 4. The frames (a) and (b) illustrate the contour plots of the real Δ_K part of the complex error function in the domain $0.1 \leqslant x \leqslant 15$ and $10^{-5} \leqslant y \leqslant 10^{-4}$ corresponding to approximation (10) at $N = 32$ and approximation (14) at $N = 23$, respectively. The frames (c) and (d) are the contour plots of the imaginary Δ_L part of the complex error function in the domain $0.1 \leqslant x \leqslant 15$ and $10^{-5} \leqslant y \leqslant 10^{-4}$ corresponding to approximation (10) at $N = 32$ and approximation (14) at $N = 23$, respectively.

$N = 23$ is 10^{-9}. The accuracy of the imaginary part of the approximation (10) is essentially better and equals to 10^{-12} as shown in Fig. 4c. However, the accuracy of the imaginary part of the approximation (14) is better by two orders of magnitudes – it is equal to 10^{-14} as Fig. 4d shows.

3.2. Computational optimization and test

The speed optimization in approximation (14) is achieved mostly due to exclusion the repetitive computation of the exponential multiplier $e^{i\tau_m z}$. Specifically, the expressions $e^{i(n\pi + \tau_m z)}$ and $e^{i(-n\pi + \tau_m z)}$ in the nominators inside parenthesis of approximation (14) can be rewritten as the multiplications $e^{in\pi} e^{i\tau_m z}$ and $e^{-in\pi} e^{i\tau_m z}$ respectively. Now we can see more explicitly that only three arrays $e^{i\tau_m z}$, $\tau_m z$ and z are involved in computation. All other summation terms and multipliers in approximation (14) like $e^{in\pi} e^{in\pi}$, $n\pi$, etc. are just scalars that being computed momentarily does not contribute at all for computation time.

In approximation (14) the exponential multiplier $e^{i\tau_m z}$ is the "slowest" array for computation. There are different ways to accelerate exponential function by rational polynomial approximation methods (see *e.g.* Ref. [20]). However, this is absolutely unnecessary. Since $e^{i\tau_m z}$ is independent of n, there are no reasons to repeat the computation over and over again $N + 1$ times – it can be computed only a single time. Once computed, the resultant array is substituted into summation. Due to this technique, the computation of Eq. (14) becomes *de facto* as fast as a rational approximation.

In addition, the optimization is also gained by appropriate implementation of the array programming method. It is well-known that in the most practical cases the array programming is a key factor for effective computation. However, the improper choice of the array dimension can decelerate the computation. We have found empirically that 1D array programming (i.e. vector programming) provides the best result in terms of the speed optimization. The increase of the array dimension does not necessarily improve the efficiency. By comparison, the programming with 2D arrays results in deceleration by two orders of the magnitude. This might occur due to unnecessary allocation of the extended intermediate data in the computer memory. It should be noted that further gain in speed can be obtained by compiling an executable file with help of Matlab Compiler [7].

Fig. 5 shows the computation acceleration factors at different values of N. At $N = 14$ the performance of the algorithm based on approximation (14) is slightly faster by about 8% and at $N = 23$ it becomes slightly slower by about 12%. Unlike approximation (10), the approximation (14) does not need an increase of integer N for more than 23 due to its rapid convergence. As follows from Fig. 4a, for the required accuracy the integer N in approximation (10) should be equal at least to 32. Since the required integer N in approximation (14) is smaller, the acceleration factor increases by about 7%. All calculations reflected in Fig. 5 were performed by averaging calculation time for input array z consisting of five million elements.

Another important aspect is related to size of the input array that can be effectively handled by an algorithm. From this point of view the implementation of the approximation (14) is considerably advantageous. The increase of the computation time with increasing size of the input array is usually non-linear. Such non-linearity is more intrinsic to the Weiderman's algorithm (see Matlab code in page 1511, Ref. [23]) that decelerates the computation at extended input array size. Due to this phenomenon, the acceleration of the algorithm based on approximation (14) becomes considerably greater – by factor more than 2.5 when the input array exceeds 30 million elements.

The approximation (10) requires specific function files performing FFT operations to get the reduced matrix of expansion coefficients. While these function files are typically available in the most of the Matlab distributives, other traditional
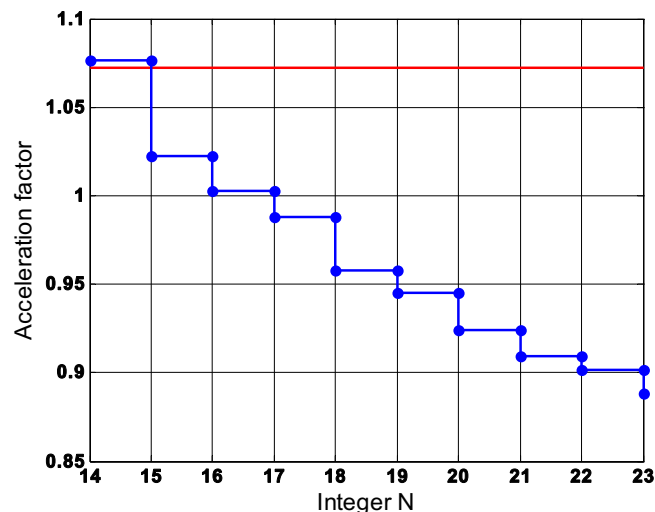

Fig. 5. Acceleration factor of the complex error function approximation (14) at different values of integer N. For comparison, the red line shows the acceleration factor at $N = 23$ in approximation (14) and $N = 32$ in approximation (10).

languages like Fortran and C/C++ require them separately since none of these function files are available in their standard libraries. The approximation (14) utilises only elementary functions that are freely available in any standard library. Therefore the utilization of approximation (14) may be practical and convenient in various numerical applications.

4. Conclusion

The error analysis shows that the algorithm based on approximation (14) provides highly accurate results for the Voigt and complex error functions. The computational test reveals that it is as fast as FFT-based Weideman's algorithm at a regular size of the input array and faster by factor more than 2.5 at an extended size of the input array. This algorithm is based on elementary functions that are freely available in a standard library of most programming languages. Due to its simplicity, rapidness, high accuracy, and possibility to cover the entire complex plane, this approximation is efficient and practically convenient in the spectral line broadening applications.

Acknowledgements

Appendix A

Let us represent the exponential function $\exp(-t^2)$ as a trigonometric series by Fourier expansion, valid within the domain $[-t_m, t_m]$:

$$\exp(-t^2) = \lim_{t_m \to \infty} \sum_{n=0}^{\infty} a_n \cos\left(\frac{n\pi}{t_m}t\right) + b_n \sin\left(\frac{n\pi}{t_m}t\right) - \frac{a_0}{2}, \tag{A.1}$$

where the expansion coefficients a_n and b_n are given by

$$a_n = \frac{1}{t_m}\int_{-t_m}^{t_m} \exp(-t^2)\cos\left(\frac{n\pi}{t_m}t\right)dt \tag{A.2}$$

and

$$b_n = \frac{1}{t_m}\int_{-t_m}^{t_m} \exp(-t^2)\sin\left(\frac{n\pi}{t_m}t\right)dt = 0.$$

The coefficients b_n are equal to zero since $\exp(-t^2)$ is an even function. Any integral in Eq. (A.2) can be found analytically as

$$a_n = -\frac{\sqrt{\pi}}{2t_m}\exp\left(-\frac{n^2\pi^2}{4t_m^2}\right)\text{Im}\left[\text{erfi}\left(\frac{n\pi}{2t_m}-it_m\right)-\text{erfi}\left(\frac{n\pi}{2t_m}+it_m\right)\right],$$

that can be simplified at $t_m \to \infty$ in form of approximation (8).

The definition of the error function of complex argument is

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z \exp(-t^2)dt, \tag{A.3}$$

where $z = x + iy$. Substituting the Fourier expansion series (A.1) into Eq. (A.3) yields

$$\text{erf}(z) = \lim_{t_m \to \infty} \frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty} a_n \int_0^z \cos\left(\frac{n\pi}{t_m}t\right)dt - \frac{2}{\sqrt{\pi}}\int_0^z \frac{a_0}{2}dt. \tag{A.4}$$

Since

$$\int \cos\left(\frac{n\pi}{t_m}t\right)dt = \frac{t_m}{n\pi}\sin\left(\frac{n\pi}{t_m}t\right),$$

the Eq. (A.4) can be rewritten as approximation series (7).

Appendix B

Combining the real and imaginary parts (3) and (4) and applying an identity $\cos(\alpha) = (e^{i\alpha} + e^{-i\alpha})/2$, we can express the complex error function as

$$w(x,y) = \frac{1}{\sqrt{\pi}}\int_0^{\infty} \exp\left(-\frac{\tau^2}{4}\right)\exp[i(x+iy)\tau]d\tau. \tag{B.1}$$

It was shown previously [2,3] that in order to obtain high accuracy, the exponential multiplier in Eqs. (3) and (4) can be expanded into Fourier series

$$\exp(-\tau^2/4) \approx \sum_{n=0}^{N} \frac{a_n}{2}\left[\exp\left(\frac{i\pi n}{\tau_m}\tau\right) + \exp\left(-\frac{i\pi n}{\tau_m}\tau\right)\right] - \frac{a_0}{2}, \quad -\tau_m \leqslant \tau \leqslant \tau_m, \tag{B.2}$$

where the corresponding expansion coefficients

$$a_n = -\sqrt{\pi}\,\mathrm{Im}\left[\frac{1}{\tau_m}\exp\left(-\frac{n^2\pi^2}{\tau_m^2}\right)\left(\mathrm{erfi}\left(\frac{n\pi}{\tau_m} - i\frac{\tau_m}{2}\right) - \mathrm{erfi}\left(\frac{n\pi}{\tau_m} + i\frac{\tau_m}{2}\right)\right)\right],$$

can be simplified as given by approximation (13) when the condition

$$n \leqslant \left\lceil\frac{\tau_m^2}{2\pi}\right\rceil$$

is satisfied [2,3]. Substituting the Fourier expansion series (B.2) into (B.1) and replacing the upper limit by the margin value $\tau_m \geqslant 12$ [2,3,9,13] yields the following approximation of the complex error function

$$w(x,y) \approx \frac{1}{2\sqrt{\pi}}\int_0^{\tau_m}\left(\sum_{n=0}^{N}a_n\left[\exp\left(\frac{in\pi}{\tau_m}\tau\right) + \exp\left(-\frac{in\pi}{\tau_m}\tau\right)\right] - a_0\right)\exp[i(x+iy)\tau]d\tau. \tag{B.3}$$

In Eq. (B.3) each integral term can be found analytically. This leads to the approximation (14) of the complex error function.

It should be noted that approximation (14) can also be obtained directly from (11) and (12) by superposition as $K(x,y) + iL(x,y)$. Although such approach is trivial, it is very tedious for derivation.

References

[1] M. Abramowitz, I.A. Stegun, Error Function and Fresnel Integrals, ninth ed., Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover, New York, 1972. pp. 297–309.
[2] S.M. Abrarov, B.M. Quine, R.K. Jagpal, Rapidly convergent series for high-accuracy calculation of the Voigt function, J. Quant. Spectrosc. Radiat. Transfer 111 (2010) 372–375.
[3] S.M. Abrarov, B.M. Quine, R.K. Jagpal, High-accuracy approximation of the complex probability function by Fourier expansion of exponential multiplier, Comput. Phys. Commun. 181 (2010) 876–882.
[4] B.H. Armstrong, B.W. Nicholls, Emission, Absorption and Transfer of Radiation in Heated Atmospheres, Pergamon Press, New York, 1972. pp. 229–234.
[5] N.L. Balazs, I. Tobias, Semiclassical dispersion theory of lasers, Phil. Trans. Roy. Soc. London A. 264 (1969) 1–29.
[6] L.K.P. Chan, Equation of atomic resonance for solid-state optics, Appl. Opt. 25 (1986) 1728–1730.
[7] S.J. Chapman, MATLAB Programming for Engineers, Thomson Learning, 2008.
[8] V.N. Faddeyeva, N.M. Terent'ev, Tables of the Probability Integral for Complex Argument, Pergamon Press, Oxford, 1961.
[9] R.S. Geetha, R.S. Keshavamurthy, R. Harish, Temperature and energy derivatives of Doppler broadening functions by Steffensen's inequality technique, Ann. Nucl. Energy 37 (2010) 985–990.
[10] A.K. Hui, B.H. Armstrong, A.A. Wray, Rapid computation of the Voigt and complex error functions, J. Quant. Spectrosc. Radiat. Transfer 19 (1978) 509–516.
[11] J. Humlíček, Optimized computation of the Voigt and complex probability functions, J. Quant. Spectrosc. Radiat. Transfer 27 (1982) 437–444.
[12] K. Imai, M. Suzuki, Ch. Takahashi, Evaluation of Voigt algorithms for the ISS/JEM/SMILES L2 data processing system, Adv. Space Res. 5 (2010) 669–675.
[13] R.S. Keshavamurthy, R.S. Geetha, New properties and approximations of Doppler broadening function using Steffensen's inequality, Nucl. Sci. Eng. 162 (2009) 192–199.
[14] M. Kuntz, A new implementation of the Humlíček algorithm for the calculation of the Voigt profile function, J. Quant. Spectrosc. Radiat. Transfer 57 (1997) 819–824.
[15] M. Leutenegger, Error function of complex numbers, Matlab Cetnral, file ID# 18312, MATLAB code and the article with detailed description in calculation of the error function of complex number can be downloaded from Matlab Central website: <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=18312>.
[16] B.M. Mamedov, Analytical evaluation of the Voigt function using binomial coefficients and incomplete gamma functions, Mon. Not. R. Astron. Soc. 387 (2008) 1622–1626.
[17] A.A.E. Milone, L.A. Milone, G.E. Bobato, Numerical evaluation of the line broadening function $H(a,v)$, Astrophys. Space Sci. 147 (1988) 229–234.
[18] G.P.M. Poppe, C.M.J. Wijers, Algorithm 680: evaluation of the complex error function, ACM Trans. Math. Softw. 161 (1990) 47.
[19] B.M. Quine, J.R. Drummond, GENSPECT: a line-by-line code with selectable interpolation error tolerance, J. Quant. Spectrosc. Radiat. Transfer 74 (2002) 147–165.
[20] A. Ralston, P. Rabinowitz, A First Course in Numerical Analysis, second ed., McGraw-Hill, 1978.
[21] W. Ruyten, Comment on "a new implementation of the Humlicek algorithm for the calculation of the Voigt profile function", J. Quant. Spectrosc. Radiat. Transfer 86 (2004) 231–233.
[22] H.M. Srivastava, E.A. Miller, A unified presentation of the Voigt functions, Astrophys. Space Sci. 135 (1987) 111–118.
[23] J.A.C. Weideman, Computation of the complex error function, SIAM J. Numer. Anal. 31 (1994) 1497–1518.
[24] R.J. Wells, Rapid approximation to the Voigt/Faddeeva function and its derivatives, J. Quant. Spectrosc, Radiat. Transfer 62 (1999) 29–48.
[25] M.R. Zaghloul, On the calculation of the Voigt line profile: a single proper integral with a damped sine integrand, Mon. Not. R. Astron. Soc. 375 (2007) 1043–1048.