 doi:10.20944/preprints202208.0404.v1

Article

A Two-Domain MATLAB Implementation for Efficient Computation of the Voigt/Complex Error Function

Sanjar M. Abrarov [1,2,3]*, Rehan Siddiqui [2,3], Rajinder K. Jagpal [2,3] and Brendan M. Quine [1,3]

1 Thoth Technology Inc., Algonquin Radio Observatory, Achray Rd, RR6, Pembroke, Canada, K8A 6W7; sanjar@thothx.com (S.M.A.), ben@thothx.com (B.M.Q.)
2 Epic College of Technology, 5670 McAdam Rd., Mississauga, Canada, L4Z 1T2; rehan@epiccollege.ca (R.S.), raj@epiccollege.ca (R.K.J.)
3 Dept. Earth and Space Science, York University, 4700 Keele St., Toronto, Canada, M3J 1P3 ; bquine@yorku.ca (B.M.Q.)
* Correspondence: sanjar@thothx.com

Abstract: In this work we develop a new algorithm for efficient computation of the Voigt/complex error function. In particular, in this approach we propose a two-domain scheme where number of the grid-points is dependent on the input parameter y. The error analysis we performed shows that the MATLAB implementation meets the requirements for radiative transfer applications involving the HITRAN spectroscopic database. The run-time test shows that this MATLAB implementation provides rapid computation especially at smaller range of the parameter x.

Keywords: complex error function; Faddeeva function; Voigt function; interpolation

1. Introduction

The complex error function, also known as the Faddeeva function, is defined as [1–4]

$$w(z) = e^{-z^2}\left(1 + \frac{2i}{\sqrt{\pi}}\int\limits_0^z e^{t^2}\,dt\right), \tag{1}$$

where $z = x + iy$ is a complex argument. The complex error function $w(z)$ is closely related to the complex probability function [2]

$$W(z) = PV\frac{i}{\pi}\int\limits_{-\infty}^{\infty}\frac{e^{-t^2}}{z-t}dt,$$

where the principal value implies that this integral remains valid at $t = z$.

The complex probability function can be written in terms of its real and imaginary parts [2]

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

such that

$$K(x,y) = \mathrm{Re}[W(z)] = PV\frac{y}{\pi}\int\limits_{-\infty}^{\infty}\frac{e^{-t^2}}{y^2+(x-t)^2}dt \tag{2}$$

and

$$L(x,y) = \mathrm{Im}[W(z)] = PV\frac{1}{\pi}\int\limits_{-\infty}^{\infty}\frac{e^{-t^2}(x-t)}{y^2+(x-t)^2}dt. \tag{3}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1

Both functions $w(z)$ and $W(z)$ are equal to each other on the upper half of the complex plane, when $y = \mathrm{Im}[z] \geq 0$ [2]. Consequently, it follows that

$$w(z) = K(x, y) + iL(x, y), \qquad y \geq 0.$$

Further we will imply that the parameter $y = \mathrm{Im}[z]$ is always equal or greater than zero.

The real part of the complex error function $K(x, y)$ is known as the Voigt function [1–4] that is widely used in Atmospheric Science to describe emission and absorption of the photons by atmospheric molecules [5,7,8]. Specifically, the Voigt function is used to compute wavelength dependent absorption coefficients by using the HITRAN molecular spectroscopic database [12]. The imaginary part of the complex error function $L(x, y)$ is also used in many applications. For example, it can describe spectral behavior of the index of refraction in the various materials [13,14].

Despite simple representations, the integrals (1), (2) and (3) do not have analytical solutions and must be computed numerically. There are many approximations are available in scientific literature [15–24,26–28]. Although this problem is known for many decades, derivation of new approximations for the complex error function $w(z)$ and developing their efficient algorithms still remains an interesting topic [29–36].

In our previous publication we proposed an algorithm for efficient computation of the complex error function based on a single-domain implementation with vectorized interpolation [32]. However, despite rapid performance it has some limitations including a limited range for the parameter x as well as restriction for input array type. As a further development, in this work we present a new MATLAB implementation without those drawbacks. The numerical analysis and computational tests we performed show that the proposed algorithmic implementation meets all the requirements in terms of accuracy and the run-time performance for efficient computation of the Voigt/complex error function in the radiative transfer applications.

2. Approximations

2.1. Sampling based approximation

In our previous publication [28] we have proposed a new sampling method based on incomplete cosine expansion of the sinc function. In particular, it is shown that using a new sampling method based on incomplete cosine expansion of the sinc function we can obtain the following approximation

$$w(z) \approx \sum_{m=1}^{M} \frac{a_m + b_m(z + i\varsigma/2)}{c_m^2 - (z + i\varsigma/2)^2}, \tag{4}$$

where the expansion coefficients are given by

$$a_m = \frac{\sqrt{\pi}(m - 1/2)}{2M^2 h} \sum_{n=-N}^{N} e^{\varsigma^2/4 - n^2 h^2} \sin\left(\frac{\pi(m - 1/2)(nh + \varsigma/2)}{Mh}\right),$$

$$b_m = -\frac{i}{M\sqrt{\pi}} \sum_{n=-N}^{N} e^{\varsigma^2/4 - n^2 h^2} \cos\left(\frac{\pi(m - 1/2)(nh + \varsigma/2)}{Mh}\right),$$

$$c_m = \frac{\pi(m - 1/2)}{2M\pi},$$

with parameters N, M, h and ς that can be taken as 23, 23, 0.25 and 2.75, respectively.

For more rapid performance in algorithmic implementation, it is reasonable to define the function

$$\Omega(z) \triangleq \sum_{m=1}^{M} \frac{a_m + b_m z}{c_m^2 - z^2},$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1

such that the equation (4) can be represented as [29]

$$w(z) \approx \Omega(z + i\varsigma/2). \tag{5}$$

Overall, approximation (5) is highly accurate. However, its accuracy deteriorates with decreasing parameter y. In order to resolve this problem we can use the following approximation [29]

$$w(z) \approx e^{-z^2} + z \sum_{m=1}^{M+2} \frac{\alpha_m - \beta_m z^2}{\gamma_m - \theta_m z^2 + z^4}, \tag{6}$$

where the expansion coefficients are

$$\alpha_m = b_m \left[c_m^2 - \left(\frac{\varsigma^2}{2} \right)^2 \right] + i\alpha_m \varsigma,$$

$$\beta_m = b_m,$$

$$\gamma_m = c_m^4 + \frac{c_m^2 \varsigma^2}{2} + \frac{\varsigma^4}{16}$$

and

$$\theta_m = 2c_m^2 - \frac{\varsigma^2}{2},$$

It is interesting to note that this approximation of the complex error function is obtained by substituting approximation (4) into the right side of the identity (see [29] for details in derivation)

$$w(z) = e^{-z^2} + \frac{w(z) - w(-z)}{2}.$$

For $|z| > 8$ one of the best choices is the approximation based on the Laplace continued fraction [1,3]. In particular, in our algorithm we used

$$w(z) \approx \cfrac{(i/\sqrt{\pi})}{z - \cfrac{1/2}{z - \cfrac{1}{z - \cfrac{3/2}{z - \cfrac{2}{z - \cfrac{5/2}{z - \cfrac{3}{z - \cfrac{7/2}{z - \cfrac{4}{z - \cfrac{9/2}{z - \cfrac{5}{z - \cfrac{11/2}{z}}}}}}}}}}}}. \tag{7}$$

This algorithm is implemented in MATLAB as a script file *fadsamp.m* that utilizes three approximations (5), (6) and (7) as follows [29]

$$w(z) \approx \begin{cases} \text{eq.}(5), & \text{if} |x + iy| \le 8 \cap y > 0.05x, \\ \text{eq.}(6), & \text{if} |x + iy| \le 8 \cap y \le 0.05x, \\ \text{eq.}(7), & \text{otherwise.} \end{cases} \tag{8}$$

As we have shown in our publication [29], approximation (8) provides highly accurate and rapid computation of the complex error function without poles and can be used to cover the entire complex plane.

2.2. Modified trapezoidal rule

In 1945, English mathematician and cryptanalyst Alan Turing, who succeeded to decrypt sophisticated machine codes of the Enigma during the Second World War [37], published an interesting paper where he proposed an elegant method of numerical integration for some class of integrals [38]. Nowadays, his method of the numerical integration that involves some advanced techniques of the residue calculus is regarded as the modified

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1

trapezoidal rule [39] or the generalized trapezoidal rule [36]. The comprehensive and detailed description of the Turing's method of integration may be found in literature [39].

In 1949, Goodwin showed how to implement Turing's idea to the integrals of kind [40]

$$\int_{-\infty}^{\infty} f(x)e^{-x^2}dx.$$

Using the method described by Goodwin, Chiarella and Reichel in their work [41] derived the series expansion for the following integral equation

$$\Psi(x,t) = U(x,t) + iV(x,t) = \frac{\Omega(x,t)}{(4\pi t)^{1/2}} \int_{-\infty}^{\infty} \frac{e^{-u^2}}{u^2 + \Omega^2(x,t)}du,$$

where $\Omega(x,t) = (1 - ix)/\left(2t^{1/2}\right)$. In particular, they showed that that the function $\Psi(x,t)$ can be approximated as a series

$$\begin{aligned}\Psi(x,t) \approx &\frac{h}{\Omega(x,t)(4\pi t)^{1/2}} + \frac{2h\Omega(x,t)}{(4\pi t)^{1/2}} \sum_{n=1}^{\infty} \frac{e^{-n^2h^2}}{\Omega^2(x,t) + n^2h^2} \\ &+ \frac{\pi e^{\Omega^2(x,t)}}{(\pi t)^{1/2}\left(1 - e^{2\pi\Omega(x,t)/h}\right)} H\left(t - \frac{h^2}{\pi^2}\right),\end{aligned} \tag{9}$$

where h is a small fitting parameter and $H(t)$ is the Heaviside step function defined as

$$H(t) = \begin{cases} 0, & \text{if } t < 0, \\ 1/2, & \text{if } t = 0, \\ 1, & \text{if } t > 0. \end{cases}$$

Thus, due to Heaviside step function the equation above can be separated into three parts

$$\Psi(x,t) \approx \frac{h}{\Omega(x,t)(4\pi t)^{1/2}} + \frac{2h\Omega(x,t)}{(4\pi t)^{1/2}} \sum_{n=1}^{\infty} \frac{e^{-n^2h^2}}{\Omega^2(x,t) + n^2h^2}, \quad t < \frac{h^2}{\pi^2}, \tag{10}$$

$$\begin{aligned}\Psi(x,t) \approx &\frac{h}{\Omega(x,t)(4\pi t)^{1/2}} + \frac{2h\Omega(x,t)}{(4\pi t)^{1/2}} \sum_{n=1}^{\infty} \frac{e^{-n^2h^2}}{\Omega^2(x,t) + n^2h^2} \\ &+ \frac{\pi e^{\Omega^2(x,t)}}{2(\pi t)^{1/2}\left(1 - e^{2\pi\Omega(x,t)/h}\right)}, \qquad t = \frac{h^2}{\pi^2},\end{aligned} \tag{11}$$

$$\begin{aligned}\Psi(x,t) \approx &\frac{h}{\Omega(x,t)(4\pi t)^{1/2}} + \frac{2h\Omega(x,t)}{(4\pi t)^{1/2}} \sum_{n=1}^{\infty} \frac{e^{-n^2h^2}}{\Omega^2(x,t) + n^2h^2} \\ &+ \frac{\pi e^{\Omega^2(x,t)}}{(\pi t)^{1/2}\left(1 - e^{2\pi\Omega(x,t)/h}\right)}, \qquad t > \frac{h^2}{\pi^2},\end{aligned} \tag{12}$$

Equation (11) deals only with a single point h^2/π^2 for the parameter t and does not represent any practical interest. Therefore, further we will consider only two equations (10) and (12).

Mata and Reichel [42] showed the relations

$$K(x,y) = \frac{1}{y\sqrt{\pi}} U\left(\frac{x}{y}, \frac{1}{4y^2}\right)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1

and

$$L(x,y) = \frac{1}{y\sqrt{\pi}} V\left(\frac{x}{y}, \frac{1}{4y^2}\right)$$

that link both functions $\Psi(x,t)$ and $w(x,y)$ with each other. Consequently, using these relations the series expansions (10) and (12) can be reformulated as

$$w(z) \approx \frac{2ihz}{\pi} \sum_{k=0}^{N} \frac{e^{-t_k^2}}{z^2 - t_k^2} \tag{13}$$

and

$$w(z) \approx \frac{2e^{-z^2}}{1 + e^{-2i\pi z/h}} + \frac{2ihz}{\pi} \sum_{k=0}^{N} \frac{e^{-t_k^2}}{z^2 - t_k^2}, \tag{14}$$

respectively, where $t_k = (k + 1/2)h$ and h can be chosen to be equal to $\sqrt{\pi/(N+1)}$ [36].

Consider an expansion series for the complementary error function that was reported by Hunter and Regan [43] (see also [36])

$$\mathrm{erfc}(z) \approx \frac{2hze^{-z^2}}{\pi} \sum_{k=1}^{N} \frac{e^{-(k-1/2)^2 h^2}}{z^2 + (k-1/2)^2 h^2} + \frac{2}{1 + e^{2\pi z/h}}, \qquad x < \frac{\pi}{h}.$$

In particular, using the identity relating complex error function and complementary error function [3]

$$w(z) = e^{-z^2}(1 - \mathrm{erfc}(-z)),$$

we get

$$w(z) \approx \frac{2e^{-z^2}}{1 + e^{-2i\pi z/h}} + \frac{ih}{\pi z} + \frac{2ihz}{\pi} \sum_{k=1}^{N} \frac{e^{-\tau_k^2}}{z^2 - \tau_k^2}, \tag{15}$$

where $\tau_k = kh$ [36].

The equations (13) and (14) have poles at $z = t_k$ while equation (15) has poles at $z = \tau_k$. Furthermore, each of these equations can cover with high accuracy only in corresponding domain. However, as recently proposed by Al Azah and Chandler-Wilde [36], the following approximation

$$w(z) \approx \begin{cases} \text{eq.}(13), & \text{if } y < x \cap 1/4 \le \varphi(x/h) \le 3/4, \\ \text{eq.}(14), & \text{if } y \ge \max(\pi, x), \\ \text{eq.}(15), & \text{otherwise,} \end{cases} \tag{16}$$

where $\varphi(t) = t - \lfloor t \rfloor \in [0,1)$, appeared to be very efficient since it can be used for rapid and highly accurate computation without poles at $N = 11$. This is possible to achieve since, according to approximation (16), the equations (13), (14) and (15) are used interchangeably depending on the domain over the entire complex plain.

3. Algorithmic implementation

Previously we have reported a new algorithm based on a vectorized interpolation over a single-domain [32]. Such an approach provides accuracy better than 10^{-6} at $y \ge 10^{-8}$ for the HITRAN [12] applications. However, this implementation has several limitations. In particular, there is a limitation $|x| \le 10^5$. Although it is possible to increase the range for more than 10^5, it requires to introduce more grid-points for precomputation. Furthermore, this MATLAB implementation accepts an input only as a vector $\boldsymbol{x} = \{x_1, x_2, x_2, \ldots x_{\max}\}$ or as a scalar.

 doi:10.20944/preprints202208.0404.v1

One of the efficient ways to implement efficient algorithm is to use an approximation based on two-domain scheme that we proposed in our earlier publication [22]

$$K(x,y) \approx \begin{cases} \text{interpolation}, & \dfrac{x^2}{27^2} + \dfrac{y^2}{15^2} \le 1 \\ \dfrac{a_1 + b_1 x^2}{a_2 + b_2 x^2 + x^4}, & \dfrac{x^2}{27^2} + \dfrac{y^2}{15^2} > 1, \end{cases} \tag{17a}$$

where the coefficients are [19]

$$\begin{aligned} a_1 &= y/\left(2\sqrt{\pi}\right) + y^3/\sqrt{\pi} \approx 0.2820948y + 0.5641896y^3 \\ b_1 &= y/\sqrt{\pi} \approx 0.5641896y \\ a_2 &= 0.25 + y^2 + y^4 \\ b_2 &= -1 + 2y^2 \end{aligned}$$

such that

$$\frac{a_1 + b_1 x^2}{a_2 + b_2 x^2 + x^4} = \mathrm{Re}\left\{ \frac{\left(i/\sqrt{\pi}\right)}{z - \frac{1/2}{z}} \right\}.$$

Consequently, the complex error function can also be approximated as [32]

$$w(z) = K(x,y) + iL(x,y) \approx \begin{cases} \text{interpolation}, & \dfrac{x^2}{27^2} + \dfrac{y^2}{15^2} \leqslant 1 \\ \dfrac{\left(i/\sqrt{\pi}\right)}{z - \frac{1/2}{z}}, & \dfrac{x^2}{27^2} + \dfrac{y^2}{15^2} > 1. \end{cases} \tag{17b}$$

Although the algorithms for the Voigt and complex error functions, built on equations (17a) and (17b) can provide rapid computations, their accuracies deteriorate at $y < 10^{-6}$.

In order to resolve this problem we developed a new algorithm that utilizes the internal MATLAB built-up features. Unlike interpolation algorithms shown in [22] and [32], the proposed approach implies that the number of the grid-points required for precomputation is not constant and dependent on the input parameter y such that

$$N_{gp} = \frac{1}{\sqrt{y}} + \delta, \qquad y \ge 0, \tag{18}$$

where the values of r and δ are radius and offset that were found experimentally to be 35 and 3×10^4, respectively. The corresponding grid-points range is given by $[-r, r]$. These grid-points are distributed non-equidistantly in logarithmic scale to increase density of the grid-points towards origin along x axis.

The algorithm utilizes two domains, internal and external that are bounded by a circle of radius $|x + iy| = r$. Internal domain is situated within circle while external domain is situated outside it.

All points within internal domain $|x + iy| \le r$ are computed by MATLAB built-in interpolation function *interp1* though grid-points that are computed by using the function file *fadsamp.m* provided in our article [29]. Spline method is found to be the best for interpolation.

All points outside external domain $|x + iy| > r$ are computed by the following approximation

$$w(z) \approx \frac{\left(i/\sqrt{\pi}\right)}{z - \frac{1/2}{z - \frac{1}{z - \frac{3/2}{z}}}}$$

that represents a simplified version of the equation (7) above.

As it has been mentioned above, the computation of the grid-points in its original version of the function file *w2dom.m* is performed by external function *fadsamp.m*. However,

 doi:10.20944/preprints202208.0404.v1

any other MATLAB function file that can provide highly accurate computation of the complex error function may also be used for computation of the grid-points. For example, the function files like *fadf.m* [26] and *fadfunc.m* [30] can also be used as an alternatives. The script of the function file *w2dom.m* is given in Appendix A.

4. Error analysis

In order to exclude the rounding and truncation errors in computation by using the most recent HITRAN database [12], the values of the $K(x, y)$ and $L(x, y)$ functions with 6 or more accurate decimal digits in their mantissas are required. Therefore, in radiative transfer applications involving the HITRAN database, the accuracy of computation of these functions has to be better than 10^{-6}.

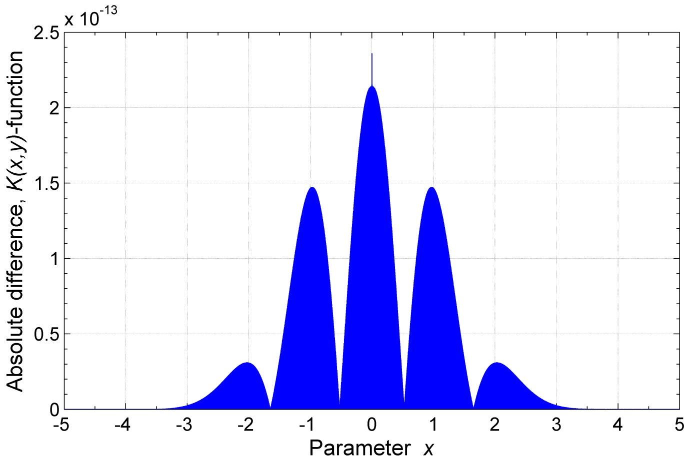

Figure 1. Absolute difference for the real part $K(x, y)$ of the complex error function.

Figure 1 shows the absolute difference $\left|K_{ref.}(x, y) - K(x, y)\right|$, where $K_{ref.}(x, y)$ is highly accurate reference, in the range $-5 \le x \le 5$ at $y = 10^{-8}$. As we can see, the absolute difference does not exceed 2.5×10^{-13}.

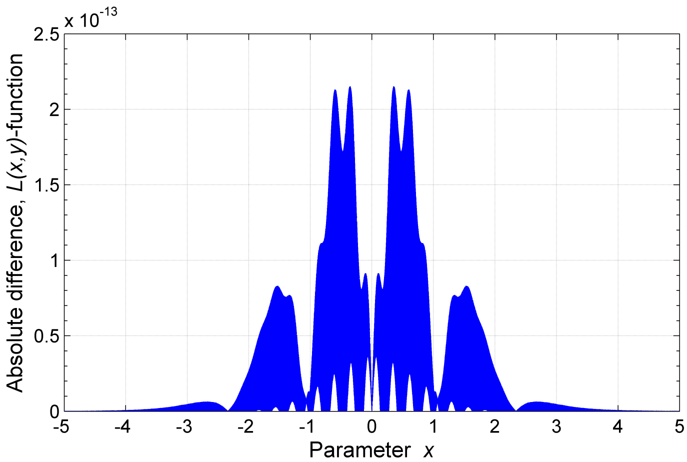

Figure 2. Absolute difference for the imaginary part $L(x, y)$ of the complex error function.

Figure 2 shows the absolute difference $\left|L_{ref.}(x, y) - L(x, y)\right|$, where $L_{ref.}(x, y)$ is highly accurate reference, in the range $-5 \le x \le 5$ also at $y = 10^{-8}$. We can see that for imaginary part the absolute difference also does not exceed 2.5×10^{-13}.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1

For more rigorous error analysis, we can apply the following relative errors

$$\Delta_{\text{Re}} = \frac{\left|K_{ref.}(x,y) - K(x,y)\right|}{K_{ref.}(x,y)}$$

and

$$\Delta_{\text{Im}} = \frac{\left|L_{ref.}(x,y) - L(x,y)\right|}{L_{ref.}(x,y)}$$

for the real and imaginary parts of the complex error function $w(z)$, respectively. 95

Figure 3 depicts the relative error for the real part of the complex error function 96 $\text{Re}[z] = K(x,y)$ in the domain $0 \le x \le 15$ and $0 \le y \le 10^2$. As we can see from this figure, 97 the relative error does not exceed $\sim 10^{-10}$. 98

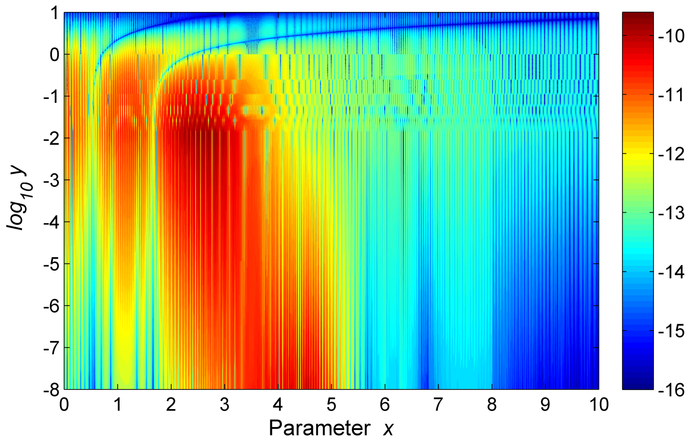

Figure 3. Relative error for the real part $K(x,y)$ of the complex error function.

Figure 4 illustrates the relative error for the imaginary part of the complex error 99 function $\text{Im}[z] = L(x,y)$ in the domain $0 \le x \le 15$ and $0 \le y \le 10^2$. As we can see from 100 this figure, the relative error is generally lower in imaginary part and does not exceed 101 $\sim 10^{-11}$. 102

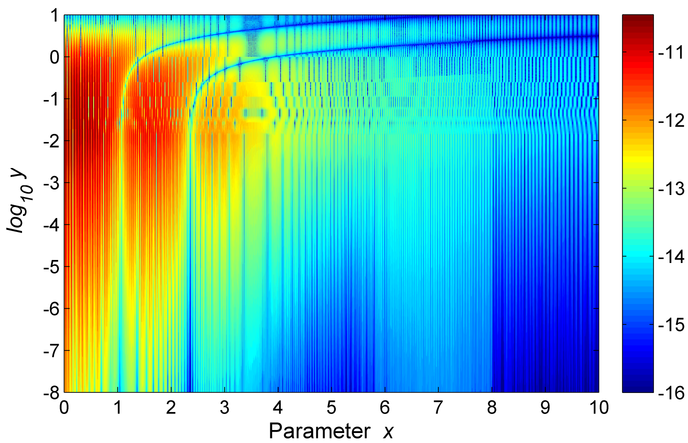

Figure 4. Relative error for the imaginary part $L(x,y)$ of the complex error function.

It is commonly known that the accuracy of $K(x,y)$ and $L(x,y)$ functions tend to 103 deteriorate when y decreases [2,15,20,21,27]. However, from the Figs. 3 and 4 we can see 104 that the decrease of the parameter y does not deteriorate the accuracy. This is possible to 105

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1

achieve since in accordance with equation (18) the number of the grid-points N_{gp} increases with decreasing y. Therefore, this technique enables us to resolve efficiently this problem in computation of the complex error function $w(z)$.

5. Run-time test

The run-time test was performed with MATLAB function files *wTrap.m* [36], *fadsamp.m* [30] and *w2dom.m* at equidistantly distributed 10 million grid-points for the parameter x. The results of the run-time test are shown in the Table 1.

Algorithm	Run-time in seconds		
	$x \in [-10, 10]$	$x \in [-10^2, 10^2]$	$x \in [-10^3, 10^3]$
wTrap.m	2.41	2.55	2.45
fadsamp.m	4.14	1.78	1.54
w2dom.m	1.23	1.04	0.98

Table 1. Run-time of algorithms for 10 million points at three different ranges.

It has been reported that both algorithms *wTrap.m* and *fadsamp.m* are highly accurate in computation [36]. In particular, the maximum values in relative errors for the algorithms *wTrap.m* and *fadsamp.m* are found to be $\sim 10^{-15}$ and $\sim 10^{-14}$, respectively. However, the computational speed of these two algorithms differs depending on the range for the input parameter x. In particular, Table 1 shows that at smaller range for the parameter x the function file *wTrap.m* performs computation faster than the function file *fadsamp.m*. However, as the range for the parameter x increases, our algorithm *fadsamp.m* becomes faster. The run-time test also reveals that the algorithm *w2dom.m* is always faster regardless the chosen range. This is particularly evident when the range for the input parameter x is smaller.

Algorithm	Run-time in seconds		
	$x \in [-10, 10]$	$x \in [-10^2, 10^2]$	$x \in [-10^3, 10^3]$
wTrap	8.38	8.33	8.42
fadsamp	14.37	6.02	5.21
w2dom	3.78	3.07	2.86

Table 2. Run-time of algorithms for 30 million points at three different ranges.

Table 2 shows the results of run-time test at 30 million equidistantly distributed points. As we can see, the algorithm *w2dom.m* remain more rapid proportionally at extended size of the input array as compared to *wTrap.m* and *fadsamp.m* algorithms.

The MATLAB code for the run-time test is provided in the Appendix B. The scripts of function files *wTrap.m* and *fadsamp.m* can be accessed from the cited literature [36] and [29], respectively.

6. Conclusion

A new algorithm for efficient computation of the Voigt/complex error function is shown. We propose a two-domain scheme where number of the grid-points N_{gp} is dependent on the parameter y. The error analysis shows that our MATLAB implementation meets the requirements for radiative transfer applications utilizing the HITRAN spectroscopic database. The run-time test we performed shows that this MATLAB implementation can provide rapid computation especially at smaller range of the parameter x.

Author Contributions: S.M.A. developed software, performed data analysis and wrote the manuscript. R.S., R.K.J. and B.M.Q. developed software and performed data analysis. All authors have read and agreed to publish the manuscript.

Funding: This research received no external funding.

 doi:10.20944/preprints202208.0404.v1

Acknowledgments: This work is supported by National Research Council Canada, Thoth Technology Inc., York University, Epic College of Technology and Epic Climate Green (ECG) Inc.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

```
function FF = w2dom(x,y,opt)

% SYNOPSIS:
%          x   - array or scalar
%          y   - scalar
%          opt - option for the real and imaginary parts
%
% opt = 1 returns value(s) for the Voigt function, K(x,y)-function
% opt = 2 returns value(s) for the L(x,y)-function
% opt = 3 returns value(s) for the complex error function
%
% This code is primarily intended to work in the range y >= 1e-8 for the
% HITRAN applications for accelerated computation of the Voigt/complex
% error function.
%
% ------------------------------------------------------------------------
% Example:
%          x = linspace(-10,10,1e7); y = 1e-8; tic; w2dom(x,y); toc
% ------------------------------------------------------------------------
if ~isscalar(y)
    disp('Input parameter y must be a scalar')
    return % terminate computation if y is not a scalar
elseif y < 1e-8
    FF = fadsamp(x+1i*y);
    return
end

if nargin == 2
    opt = 3;
%     disp('Default value opt = 3 is assigned.')
end

if opt ~= 1 && opt ~= 2 && opt ~= 3
    % disp(['Wrong parameter opt = ',num2str(opt),'! Use 1, 2 or 3.'])
    return
end

FF = zeros(size(x));
radius = 35; % define radius
ind = abs(x + 1i*y) <= radius;

gp =[-radius,radius]; % define grid-points
if ~isempty(x(ind))

    offset = 5*1e3; % assign offset
    nump = 1/sqrt(y) + offset; % assign number of points
    % For better accuracy use for example nump = 2/sqrt(y) + 3*offset;

    gp = radius*(logspace(log10(1 + eps),log10(2),nump) ...
    - 1); % notice the log scale

    gp = [flip(-gp),gp];
end

switch opt
    case 1
        FF(ind) = real(internD(x(ind),y,gp));
        FF(~ind) = real(externD(x(~ind) + 1i*y));
    case 2
        FF(ind) = imag(internD(x(ind),y,gp));
        FF(~ind) = imag(externD(x(~ind) + 1i*y));
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 August 2022 doi:10.20944/preprints202208.0404.v1


```
    otherwise
        FF(ind) = internD(x(ind),y,gp);
        FF(~ind) = externD(x(~ind) + 1i*y);
end

    function IntD = internD(x,y,gp) % internal domain
        IntD = interp1(gp,fadsamp(gp + 1i*y),x, ...
            'spline'); % interpolated values
    end

    function ExtD = externD(z) % external domain

        num = 1:4; % define a row vector
        num = num/2;

        ExtD = num(end)./z; % start computing from the end
        for m = 1:length(num) - 1
            ExtD = num(end - m)./(z - ExtD);
        end
        ExtD = 1i/sqrt(pi)./(z - ExtD);
    end
end
```

Appendix B

```
clear
clc

% Table for run-time in seconds
tab{1,1}='Algorithm';
tab{1,2}='-10 to 10'; % range 1
tab{1,3}='-10^2 to 10^2'; % range 2
tab{1,4}='-10^3 to 10^3'; % range 3

tab{2,1}='wTrap';
tab{3,1}='fadsamp';
tab{4,1}='w2dom';

y=1e-8; % smallest value for the parameter y
maxN=10; % max number of cycles

for k=[1,3] % k is a factor
    for m=1:3 % three ranges
        x0=10^m; x=linspace(-x0,x0,k*1e7); % 1 and 3 are for 1e7 and 3*1e7 ...
                                           % points, respectively

        if k==1
            disp(['10 million points in range ',num2str(m)])
        else
            disp(['30 million points in range ',num2str(m)])
        end
        disp('Computing, please wait!')

        tic; for n=1:maxN; wTrap(x+1i*y,11); end; tab{2,m+1}=toc/maxN;
        tic; for n=1:maxN; fadsamp(x+1i*y); end; tab{3,m+1}=toc/maxN;
        tic; for n=1:maxN; w2dom(x,y); end; tab{4,m+1}=toc/maxN;

        clc
    end
    if k==1; tab1=tab; else tab2=tab; end % assign tab1 or tab2
end

disp('Displaing Table 1:')
disp(tab1)

disp('Displaing Table 2:')
disp(tab2)
```

References

1. Faddeyeva, V.N.; Terent'ev, N.M. *Tables of the probability integral* $w(z) = e^{-z^2}\left(1 + \frac{2i}{\sqrt{\pi}}\int_0^z e^{t^2}dt\right)$ *for complex argument*; Publisher: Pergamon Press, Oxford, 1961.
2. Armstrong, B.H. Spectrum line profiles: The Voigt function. *J. Quant. Spectrosc. Radiat. Transfer*, **1967**, *7(1)*, 61–88. https://doi.org/10.1016/0022-4073(67)90057-X
3. Gautschi, W. Efficient computation of the complex error function. *SIAM J. Numer. Anal.*, **1970**, *7(1)*, 187–198. https://doi.org/10.1137/0707012
4. Abramowitz, M.; Stegun, I.A. *Error function and Fresnel integrals. Handbook of mathematical functions with formulas, graphs, and mathematical tables*, 9th ed.; New York 1972, pp. 297–309.
5. Quine, B.M.; Drummond, J.R. GENSPECT: a line-by-line code with selectable interpolation error tolerance. *J. Quantit. Spectrosc. Radiat. Transfer*, **2002**, *74(2)*, 147–165. https://doi.org/10.1016/S0022-4073(01)00193-5
6. Jagpal, R.K.; Quine, B.M.; Chesser, H.; Abrarov, S.; Lee, R. Calibration and in-orbit performance of the Argus 1000 spectrometer - the Canadian pollution monitor. *J. Appl. Remote Sens.*, **2010**, *4(1)*, 049501. https://doi.org/10.1117/1.3302405
7. Berk. A; Hawes, F. Validation of MODTRAN®6 and its line-by-line algorithm. *J. Quant. Spectrosc. Radiat. Transfer*, **2017**, *203*, 542–556. https://doi.org/10.1016/j.jqsrt.2017.03.004
8. Pliutau, D.; Roslyakov, K. Bytran -|- spectral calculations for portable devices using the HITRAN database. *Earth Sci. Inform.*, **2017**, *10(3)*, 395–404. https://doi.org/10.1007/s12145-017-0288-4
9. Siddiqui, R.; Jagpal, R.; Quine, B.M. Short wave upwelling radiative flux (SWupRF) within near infrared (NIR) wavelength bands of O_2, H_2O, CO_2, and CH_4 by Argus 1000 along with GENSPECT line by line radiative transfer model. *Canad. J. Remote Sens.*, **2017**, *43(4)*, 330–344. https://doi.org/10.1080/07038992.2017.1346467
10. Siddiqui, R.; Jagpal, R.K.; Abrarov, S.M.; Quine, B.M. Efficient application of the Radiance Enhancement method for detection of the forest fires due to combustion-originated reflectance. *J. Environ. Protect.*, **2021**, *12(10)*, 717–733. https://doi.org/10.4236/jep.2021.1210043
11. Jagpal, R.K.; Siddiqui, R.; Abrarov, S.M.; Quine, B.M. Effect of the instrument slit function on upwelling radiance from a wavelength dependent surface reflectance. **Natur. Sci.**, **2022**, *14(3)*, 133–147. https://doi.org/10.4236/ns.2022.143013
12. Hill, C.; Gordon, I.E.; Kochanov, R.V.; Barrett, L.; Wilzewski, J.S.; Rothman, L.S. HITRAN*online*: An online interface and the flexible representation of spectroscopic data in the HITRAN database. *J. Quant. Spectrosc. Radiat. Transfer*, **2016**, *177*, 4–14. https://doi.org/10.1016/j.jqsrt.2015.12.012
13. Balazs, N.L.; Tobias, I. Semiclassical dispersion theory of lasers. *Phil. Trans. Roy. Soc. London A.*, **1969**, *264*, 1–29. https://doi.org/10.1098/rsta.1969.0002
14. Chan, L.K.P. Equation of atomic resonance for solid-state optics. *Appl. Opt.*, **1986**, *25*, 1728–1730. https://opg.optica.org/ao/abstract.cfm?uri=ao-28-24-5208
15. Humlíček, J. Optimized computation of the Voigt and complex probability functions. *J. Quant. Spectrosc. Radiat. Transfer*, **1982**, *27(4)*, 437–444. https://doi.org/10.1016/0022-4073(82)90078-4
16. Drummond, J.R.; Steckner, M. Voigt-function evaluation using a two-dimensional interpolation scheme. *J. Quant. Spectrosc. Radiat. Transfer*, **1985**, *34(6)*, 517–521. https://doi.org/10.1016/0022-4073(85)90145-1
17. Poppe, G.P.M.; Wijers, C.M.J. More efficient computation of the complex error function. *ACM Transact. Math. Software*, **1990**, *16(1)*, 38–46. https://doi.org/10.1145/77626.77629
18. Schreier, F. The Voigt and complex error function: A comparison of computational methods. *J. Quant. Spectrosc. Radiat. Transfer*, **1992**, *48(5–6)*, 743–762. https://doi.org/10.1016/0022-4073(92)90139-U
19. Kuntz, M. A new implementation of the Humlicek algorithm for calculation of the Voigt profile function. *J. Quant. Spectrosc. Radiat. Transfer*, **1997**, *51(6)*, 819–824. https://doi.org/10.1016/S0022-4073(96)00162-8
20. Wells, R.J. Rapid approximation to the Voigt/Faddeeva function and its derivatives. *J. Quant. Spectrosc. Radiat. Transfer*, **1999**, *62(1)*, 29–48. https://doi.org/10.1016/S0022-4073(97)00231-8
21. Letchworth K.L.; Benner, D.C. Rapid and accurate calculation of the Voigt function. *J. Quant. Spectrosc. Radiat. Transfer*, **2007**, *107(1)*, 173–192. https://doi.org/10.1016/j.jqsrt.2007.01.052
22. Abrarov, S.M.; Quine B.M.; Jagpal, R.K. A simple interpolating algorithm for the rapid and accurate calculation of the Voigt function. *J. Quant. Spectrosc. Radiat. Transfer*, **2009**, em 110(6–7), 376–383. https://doi.org/10.1016/j.jqsrt.2009.01.003
23. Pagnini, G.; Mainardi, F. Evolution equations for the probabilistic generalization of the Voigt profile function. *J. Comput. Appl. Math.*, **2010**, *233(6)*, 1590–1595. https://doi.org/10.1016/j.cam.2008.04.040
24. Imai, K.; Suzuki, M.; Takahashi, C. Evaluation of Voigt algorithms for the ISS/JEM/SMILES L2 data processing system. *Adv. Space Res.*, *2010*, *45(5)*, 669–675. https://doi.org/10.1016/j.asr.2009.11.005
25. Zaghloul. M.R.; Ali, A.N. Algorithm 916: Computing the Faddeyeva and Voigt Functions. *ACM Trans. Math. Software*, **2011**, *38(2)*, 15. https://doi.org/10.1145/2049673.2049679
26. Abrarov, S.M.; Quine, B.M. Efficient algorithmic implementation of the Voigt/complex error function based on exponential series approximation. *Appl. Math. Comput.*, **2011**, *218(5)*, 1894–1902. https://doi.org/10.1016/j.amc.2011.06.072

27. Amamou, H.; Ferhat, B.; Bois, A. Calculation of the Voigt Function in the region of very small values of the parameter *a* where the calculation is notoriously difficult. *Amer. J. Anal. Chem.*, **2013**, *4(12)*, 725–731. https://doi.org/10.4236/ajac.2013.412087
28. Abrarov, S.M.; Quine, B.M. Sampling by incomplete cosine expansion of the sinc function: Application to the Voigt/complex error function. *Appl. Math. Comput.*, **2015**, *258*, 425–435. https://doi.org/10.1016/j.amc.2015.01.072
29. Abrarov, S.M., Quine, B.M.; Jagpal, R.K. A sampling-based approximation of the complex error function and its implementation without poles. *Appl. Num. Math.*, **2018**, *129*, 181–191. https://doi.org/10.1016/j.apnum.2018.03.009
30. Abrarov, S.M.; Quine, B.M. A rational approximation of the Dawson's integral for efficient computation of the complex error function. *Appl. Math. Comput.*, **2018**, *321*, 526–543. https://doi.org/10.1016/j.amc.2017.10.032
31. Wang, S.; Huang, S. Evaluation of the numerical algorithms of the plasma dispersion function. *J. Quant. Spectrosc. Radiat. Transfer*, **2019**, *234*, 64–70. https://doi.org/10.1016/j.jqsrt.2019.06.006
32. Abrarov, S.M.; Quine, B.M.; Siddiqui, R.; Jagpal, R.K. A single-domain implementation of the Voigt/complex error function by vectorized interpolation. *Earth Sci. Res.*, **2019**, *8(2)*, 52–63. https://doi.org/10.5539/esr.v8n2p52
33. Kumar, R. The generalized modified Bessel function and its connection with Voigt line profile and Humbert functions. *Adv. Appl. Math.*, **2020**, *114*, 101986. https://doi.org/10.1016/j.aam.2019.101986
34. Nordebo, S. Uniform error bounds for fast calculation of approximate Voigt profiles. *J. Quant. Spectrosc, Radiat. Transfer*, **2021**, *270*, 107715. https://doi.org/10.1016/j.jqsrt.2021.107715
35. Pliutau, D. Combined "Abrarov/Quine-Schreier-Kuntz (AQSK)" algorithm for the calculation of the Voigt function. *J. Quant. Spectrosc. Radiat. Transfer*, **2021**, *272*, 107797. https://doi.org/10.1016/j.jqsrt.2021.107797
36. Al Azah, M.; Chadler-Wilde, S.N. Computation of the complex error function using modified trapezoidal rules. *SIAM J. Numer. Anal.*, **2021**, *59(5)*, 2346–2367. https://doi.org/10.1137/20M1373037
37. Copeland, B.J. *The essential turing: seminal writings in computing, logic, philosophy, artificial intelligence, and artificial life: plus the secrets of Enigma*; Oxford University Press Inc., New York 2004.
38. Turing, A.M. A method for the calculation of the zeta-function. *Proc. London Math. Soc.*, **1945**, *s2-48(1)*, 180–197. https://doi.org/10.1112/plms/s2-48.1.180
39. Trefethen, L.N.; Weideman, J. The exponentially convergent trapezoidal rule. *SIAM Review*, **2014**, *56(3)*, 385–458. https://www.jstor.org/stable/24248471
40. Goodwin. E. The evaluation of integrals of the form $\int_{-\infty}^{\infty} f(x)e^{-x^2}dx$. *Math. Proc. Cambridge Phil. Soc.*, **1949**, *45(2)*, 241–245. https://doi.org/10.1017/S0305004100024786
41. Chiarella, C.; Reichel, A. On the evaluation of integrals related to the error function. *Math. Comput.*, **1968**, *22(101)*, 137–143. http://www.jstor.org/stable/2004771
42. Matta, F.; Reichel, A. Uniform computation of the error function and other related functions. *Math. Comp.*, **1971**, *25(144)*, 339–344. https://doi.org/10.2307/2004930
43. Hunter, D.; Regan, T. A note on the evaluation of the complementary error function. *Math. Comp.*, **1972**, *26*, 539–541. https://doi.org/10.2307/2005181