

# On the Fourier expansion method for highly accurate computation of the Voigt/complex error function in a rapid algorithm

S. M. Abrarov\* and B. M. Quine\*†

Posted to arXiv on May 7, 2012

## Abstract

In our recent publication [1] we presented an exponential series approximation suitable for highly accurate computation of the complex error function in a rapid algorithm. In this Short Communication we describe how a simplified representation of the proposed complex error function approximation makes possible further algorithmic optimization resulting in a considerable computational acceleration without compromise on accuracy.

**Keywords:** Complex error function; Voigt function; Faddeeva function; complex probability function; plasma dispersion function; spectral line broadening

Consider the exponential approximation of the complex error function, obtained by Fourier expansion method [1]

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

---

\*Dept. Earth and Space Science and Engineering, York University, Toronto, Canada, M3J 1P3.

†Dept. Physics and Astronomy, York University, Toronto, Canada, M3J 1P3.

where  $z = x + iy$  is a complex argument and

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

is a set of the Fourier expansion coefficients.

Let us take a term corresponding to index  $n = 0$  apart from the summation sign in approximation series (1). This leads to the following representation

$$w(z) \approx i \frac{1 - e^{i\tau_m z}}{\tau_m z} + \frac{i\tau_m}{2\sqrt{\pi}} \sum_{n=1}^N a_n \left( \frac{1 - e^{in\pi} e^{i\tau_m z}}{n\pi + \tau_m z} - \frac{1 - e^{-in\pi} e^{i\tau_m z}}{n\pi - \tau_m z} \right). \quad (2)$$

The expression inside parentheses of the approximation (2) can be rearranged as

$$\frac{1 - e^{in\pi} e^{i\tau_m z}}{n\pi + \tau_m z} - \frac{1 - e^{-in\pi} e^{i\tau_m z}}{n\pi - \tau_m z} = e^{i\tau_m z} \frac{n\pi (e^{in\pi} - e^{-in\pi}) + \tau_m z (e^{in\pi} + e^{-in\pi})}{n^2\pi^2 - \tau_m^2 z^2} - \frac{2\tau_m z}{n^2\pi^2 - \tau_m^2 z^2}.$$

Since  $(e^{in\pi} - e^{-in\pi}) / (2i) = \sin(n\pi) \equiv 0$  and  $(e^{in\pi} + e^{-in\pi}) / 2 = \cos(n\pi) \equiv (-1)^n$ , we can rewrite it in form

$$\frac{1 - e^{in\pi} e^{i\tau_m z}}{n\pi + \tau_m z} - \frac{1 - e^{-in\pi} e^{i\tau_m z}}{n\pi - \tau_m z} = 2\tau_m z \frac{(-1)^n e^{i\tau_m z} - 1}{n^2\pi^2 - \tau_m^2 z^2}.$$

Ultimately, applying this relation in approximation (2) yields

$$w(z) \approx i \frac{1 - e^{i\tau_m z}}{\tau_m z} + i \frac{\tau_m^2 z}{\sqrt{\pi}} \sum_{n=1}^N a_n \frac{(-1)^n e^{i\tau_m z} - 1}{n^2\pi^2 - \tau_m^2 z^2}. \quad (3)$$

Comparing the approximation representations (1) and (3) with each other, we can see that the second one is substantially simplified and we can take advantage of it for computational optimization.

In order to describe shortly and conveniently how the simplified representation of the complex error function (3) can be appropriately implemented in a rapid algorithm, we will use the standard Matlab notations for element-by-element array multiplication `.*` and division `./` with regular and bold letters for scalars and arrays, respectively. Thus the algorithmic implementation based on approximation (3) can be symbolically conformed to

$$\mathbf{w} := i(1 - \mathbf{B}) ./ \mathbf{A} + i \frac{\tau_m}{\sqrt{\pi}} \mathbf{A} .* \sum_{n=1}^N a_n [(-1)^n \mathbf{B} - 1] ./ (n^2 \pi^2 - \mathbf{C}),$$

where  $\mathbf{A} := \tau_m \mathbf{z}$ ,  $\mathbf{B} := e^{i\mathbf{A}}$ , and  $\mathbf{C} := \mathbf{A} * \mathbf{A}$  are the only three arrays involved in computation (after initialization of  $\mathbf{A}$ , the array  $\mathbf{z}$  is never used anymore and can be immediately cleaned from the computer memory). Since none of these arrays are dependent on index  $n$ , each of them are computed just a single time. As follows from this scheme, only the array  $\mathbf{B}$  requires the exponentiation. All other operations are trivial such as summation, multiplication and division. Since the calculation of the array  $\mathbf{B}$  is made only once, it has a very minor contribution (at  $\tau_m = 12$  and  $N = 23$  it is less than 6%) for the total computation time. This explains why the calculation of the Voigt/complex error function is *de facto* as rapid as a rational approximation.

The notation  $e^{i\mathbf{A}}$  implies element-wise exponential operation over the array  $\mathbf{A}$  that returns the array  $\mathbf{B}$  with same dimension; such element-wise functional operations are typical in array programming languages like Matlab/GNU Octave or modern FORTRAN.

It should be noted that if the high-accuracy is not needed, the acceleration can be additionally gained by reducing the values of  $\tau_m$  and  $N$ , say to 9 and 12, respectively. The approximation of the exponential function  $e^{i\tau_m z}$  by a rational function is not required as its contribution to the total computation time remains insignificant even in this case.

The advantage of the approximation (3) implementation becomes particularly evident at an extended size of the input array. Specifically, the computational test reveals that with size of input array  $\mathbf{z}$  exceeding 30 million elements, the algorithm based on approximation (3) is more than twice faster than the one based on approximation (1) and, compared to the Weideman's algorithm (we used Matlab code in page 1511, Ref. [2]), it becomes faster by a factor greater than 5 (see also Ref. [1] for comparison).

Similar to approximation (1) the numbers generated by approximation (3) match up to the last decimal digits with those generated by well-known Algorithm 680 [1, 3]. This signifies that due to simplified form of the approximation (3) the considerable acceleration in computation is achieved without compromise on accuracy.

All calculations were performed with help of Matlab 7.9.0 on a typical desktop computer Intel(R) Quad CPU with RAM 4.00 GB.

## References

- [1] S. M. Abrarov, B. M. Quine, Efficient algorithmic implementation of the Voigt/complex error function based on exponential series approximation, *Appl. Math. Comp.*, 218 (2011) 1894-1902. <http://dx.doi.org/10.1016/j.amc.2011.06.072>
- [2] J. A. C. Weideman, Computation of the complex error function, *SIAM J. Numerical Analysis*, 31 (1994) 1497-1518.
- [3] G. P. M. Poppe, C. M. J. Wijers, Algorithm 680: evaluation of the complex error function, *ACM Trans. Math. Softw.*, 161 (1990) 47.

---

Submitted to journal on October 18, 2011.