Regular Article

A 4-Term Exponential-Quadratic Approximation for Gaussian Q or Error Functions Accurate to 1.65×10^{-4}

Graham W. Pulford

BandGap AI, Toulouse, France

Correspondence: Graham W. Pulford, gpulford@ieee.org

Communication: received 13 July 2021, revised 10 October 2022, accepted 30 October 2022
Online publication: 7 November 2022, Digital Object Identifier: 10.21553/rev-jec.276
The associate editor coordinating the review of this article and recommending it for publication was Prof. Vo Nguyen Quoc Bao.

Abstract– **Integrals on $[0,\infty)$ where the integrand is of the form $Q^n(a\sqrt{x})\,\mathrm{p}(x)$, where Q is the Gaussian Q function, $\mathrm{p}(\cdot)$ a Gamma PDF, n a positive integer and $a > 0$; or of the form $\mathrm{erf}^n(ax+b)\,x^r\exp(-c^2x^2+2dx)$, where $\mathrm{erf}(x)$ is the error function, with integers $r \geq 0$, $n > 0$, arise in performance modelling of communication and machine learning systems. Such integrals cannot be evaluated analytically in general, but they are reducible to a set of key integrals whose integrand is $\mathrm{erf}^n(ax+b)\,\mathrm{N}(x;m,s)$ where $\mathrm{N}()$ is a Gaussian PDF with mean m and variance s. Seeking an efficient and accurate evaluation method, we develop a new 4-term exponential quadratic approximator (EQA) for the error function that includes both linear and quadratic terms in its exponents. The EQA minimises a sum-of-squares cost function with two "spline-type" constraints, i.e., constraints on the function value and its first derivative. This constrained optimisation problem is reduced to an unconstrained one by inverting a 4-D linear system, then solved by gradient descent. The resulting approximator has a maximum absolute error of 1.65×10^{-4} on the real line, and outperforms many other exponential sum approximators for $\mathrm{erf}(x)$ on $x \in [0, 1.5]$ and for $Q(x)$ on $x \in [0, 2]$. Moreover, due to its functional form, the EQA leads to an analytical formula for the set of key integrals, which, in the $n = 1$ case, is accurate to 3 to 4 significant figures while being orders of magnitude more efficient than Monte Carlo integration. The EQA can equally be used to obtain closed forms for the average symbol error probability of various modulation schemes on Rayleigh fading channels.**

Keywords– **Approximation methods, Error function, Error function integral, Error probability, Exponential approximation, Exponential quadratic approximation, Gradient descent, Q-function, Rayleigh channel, Symbolic algebra.**

1 Introduction

1.1 Background & Motivation

The error function $\mathrm{erf}(x)$ and the closely related $Q(x)$ function, giving the tail probability of the Gaussian distribution, satisfy the functional relationships $Q(x) = \frac{1}{2}(1 - \mathrm{erf}(x/\sqrt{2}))$ and $\mathrm{erf}(x) = 1 - 2Q(x\sqrt{2})$. These functions have been extensively studied as they arise in fields of science and technology that involve statistical modelling of uncertainty under Gaussian noise assumptions, and, not least, in the performance analysis of communication systems. Neither of these functions has a closed form in terms of elementary functions. Numerical library routines for the error function have been based on minimax approximations, which minimize the maximum error, for absolute error bounds dating back to 1955 [1]. Clenshaw [2] presented a pair of rational Chebyshev expansions, one for $|x| \leq 4$ and one for $x > 4$, accurate to 20 places in 1962. Rational Chebyshev approximations for optimising a relative error bound, accurate to 6×10^{-19}, were developed in 1969 by Cody [3]. Schonfelder details Chebyshev expansions for $\mathrm{erf}(x)$ and $Q(x)$, and their complements, that are accurate to 30 decimal places and were used in the 1978 Numerical Algorithms Group (NAG) library [4], which form the basis for various numerical software packages. More recent approaches are also based on power series expansions, rather than rational expansions, which can be used to produce approximations to arbitrary precision by bounding the truncation error.

While numerical libraries include highly accurate implementations of the error function, there is a need in many applications to approximate these functions in an analytical form. This is the case where functional dependency needs to be studied (for tuning and optimisation), or as a component of a more complicated system performance model. Series implementations often need a large number of terms to achieve a desired accuracy. For instance, for a standard series implementation (see later equation (6)), achieving a target precision of $2^{-16} \approx 1.5E-5$ in 32-bit arithmetic would require hundreds of terms [5]. This is clearly not a practical approach for obtaining an analytical expression, which would itself include high-order factorial or combinatorial terms arising from the binomial expansion. There is thus considerable interest in studying function-based approximations for the error function and Q function, which are extensively used in performance analysis of communication systems; and consequently one finds an extensive body of literature on this subject.

In this article, we focus on two main families of 1-dimensional definite integrals on the non-negative real line $[0,\infty)$. The first of these is of the form:

$$\mathbf{J}_n(a) \triangleq \mathrm{E}_Z[Q^n(a\sqrt{z})] = \int_0^\infty Q^n(a\sqrt{z})\,\mathrm{p}_Z(z)\,dz, \quad (1)$$

where $a,\ n > 0$, z is the instantaneous received SNR, with probability density function (PDF) $\mathrm{p}_Z(z)$, which is assumed to be a Gamma distribution. This type of integral arises in computing the symbol error probability under various modulation schemes [6–9]. The integral (1) can be expressed as infinite series via the confluent hypergeometric function (CHGF). A change of variables with $z = x^2$ reexpresses the integral as:

$$\mathbf{J}_n(a) = \int_0^\infty Q^n(ax)\,\mathrm{p}_X(x)\,dx = 2^{-n}\int_0^\infty \mathrm{erfc}^n(\frac{ax}{\sqrt{2}})\,\mathrm{p}_X(x)\,dx, \tag{2}$$

where $\mathrm{p}_X(x)$ is the PDF of $x = \sqrt{z}$ and $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$ is the complementary error function.

The second family of integrals is of the form:

$$\int_0^\infty \mathrm{erf}^n(ax+b)\,x^r\,\mathrm{e}^{-c^2x^2+2dx}\,dx,\ n,a>0,\ r,c\geq 0. \tag{3}$$

Integrals of this type arise in modelling certain adaptive machine learning systems such as generative adversarial networks [10]. They are reducible to explicit functions involving exponentials and error functions and integrals $\mathbf{I}^{(n)}(a,b,m,s)$, for integers $n > 0$ and reals $a,\ s > 0$, of the form:

$$\mathbf{I}^{(n)}(a,b,m,s) = \frac{1}{\sqrt{2\pi s}}\int_0^\infty \Phi(ax+b)^n \exp\left[\frac{-(x-m)^2}{2s}\right]dx, \tag{4}$$

where $\Phi(x)$ is shorthand for $\mathrm{erf}(x)$, an abbreviation that we will use throughout the paper, and which should not be confused with the normal cumulative distribution function. Clearly $\mathbf{I}^{(n)}(a,b,m,s) = \int_0^\infty \Phi(ax+b)^n\,\mathrm{N}(x;m,s)dx$ where $\mathrm{N}(x;m,s)$ is a Gaussian PDF in x with mean m and variance s. Since $|\Phi(x)| \leq 1$, it follows that $|\mathbf{I}^{(n)}(a,b,m,s)| \leq 1$. The integral evaluates to [11]: $\Phi((b+ma)/\sqrt{1+2a^2s})$ if the range of integration is extended to $(-\infty,\infty)$ in the special case $n = 1$.

When the PDF $\mathrm{p}_Z(z)$ in (1) is a Gamma distribution, (2) has a quadratic exponential argument (e.g., Rayleigh and Maxwell cases). Hence (1) is also reducible to explicit functions (exponentials and error functions) and integrals of the form $\mathbf{I}^{(n)}(a,b,m,s)$. So we can view $\{\mathbf{I}^{(n)}(a,b,m,s)\}$ as a set of *key integrals* on which both (2) and (3) depend. Although in certain special cases $\mathbf{I}^{(n)}(a,b,m,s)$ can be evaluated analytically, these integrals generally have no closed form solution, even for $n = 1$, and particularly if $b \neq 0$. The key integrals may be expressed as infinite series of closed-form integrals via the series expansions for the error function (5) and (6), but the subsequent expressions often exhibit poor convergence and are not suitable for efficient numerical computation. Similar observations concerning a definite integral similar to (4) were made by Fayed and Atiya [12], who obtained a series in Hermite polynomials and Gamma functions, valid for $n = 1$, $m = 0$, $s = \frac{1}{2}$ and $|a| < 1$.

Practical approximations, that is ones having both reasonable accuracy and computational cost, for the error function, complementary error function and Q-function have been dealt with in [13–15]. In terms of functional approximations with a small number of terms, which will be of interest here, Bao et al. [16] distinguished two classes: (i) Mills' ratio based forms; (ii) sums of exponentials. (A third class of polynomial forms is not suitable due to its complexity.) These approximations are presented in more detail in section 2.

Our approach to the evaluation of integrals of the form (2) and (3) in this paper is by finding a practical approximation for the evaluation of the key integrals (4). We seek a method that yields a closed form (analytical) formula that is both computationally efficient and accurate. Due to the form of their denominators, none of the Mills ratio approximations are suitable. The same applies to rational-times-exponential approximations. None of these functional forms yields a closed-form when substituted in (4). In contrast, sums of exponential approximations yield combinations of exponentials and error functions when substituted in (4), and are potential candidates. Such approximations are useful for asymptotic performance models in communication systems because they accurately approximate the large-argument behaviour of the Q and error functions. However, as we will see, the existing approximations based on sums of exponentials typically fail to model accurately the medium and small argument behaviour of these functions. We are thus led to consider modifications of sums of exponential functions that provide an accurate approximation to $Q(x)$ and $\mathrm{erf}(x)$ on the half real line $x \geq 0$ and, by extension, on the *entire real line*.

1.2 Structure and Contributions

Following a review of known approximations to $Q(x)$ and $\mathrm{erf}(x)$ in section 2, we consider, in section 3.1, a conventional M-term exponential-quadratic approximator (EQA) for the error function of the form:

$$\mathrm{erf}(x) \approx 1 - \sum_{i=1}^{M} c_i \mathrm{e}^{-a_i x^2},\ x \in R,$$

where $\{a_i > 0\}$ and $\{c_i\}$ are parameters to be estimated. Minimising a sum-of-squares cost function by gradient descent leads to the expected result: an approximation that provides a poor fit to $\mathrm{erf}(x)$ for small and medium arguments. In section 3.2 we consider a $M = 3$ term EQA where we impose 3 spline-like constraints: function values at $x = 0$ and $x = 1$ and a first derivative constraint at $x = 1$. The 3 equality constraints are used to solve a linear system that reduces the constrained 6-parameter estimation problem to an unconstrained 3-parameter optimisation problem. In a different context, the idea of using a completely determined linear system has also been applied to obtain the minimax approximation to the Q-function using sums of exponential-quadratic functions in [17].

While the approximation in section 3.2 is still poor for small arguments, this example serves as a basis for the more complicated example in section 3.3. In this latter section, we develop a 4-term EQA involving additional linear terms in the exponentials for $\mathrm{erf}(x)$ of the form:

$$\mathrm{erf}(x) \approx 1 - \sum_{i=1}^{4} c_i \mathrm{e}^{-a_i x^2 + 2b_i x},\ x \in R.$$

The approximator is subject to 4 constraints: function values and first derivatives at $x = 0$ and $x = u > 0$. The presence of the linear terms $\{b_i\}$ allows the matching of the derivative at $x = 0$ but at the expense of solving a 12-parameter constrained optimisation problem. The reformulation of this 12-D constrained optimisation problem as an unconstrained 8-D optimisation problem is the main theoretical result of the paper. The recasting allows the approximation parameters to be easily obtained via gradient descent optimisation. Due to the complexity of the calculations, which require symbolic algebra, only the method is presented in this section.

The results of the gradient descent optimisation formulated in section 3.3 are presented in section 4. This section also contains a 10-way performance comparison giving the absolute relative error in approximating the error and Q functions. The 4-term EQA for $\mathrm{erf}(x)$ is shown to be superior to the other approximators for $x \in [0,\ 1.43]$ and is applicable on the entire real line. Section 5 gives a number of application examples of the 4-term EQA, the first of which is in approximating the key integral (4) when $n = 1$. The numerical accuracy of the approximation is tested on some special cases (for which $\mathbf{I}^{(1)}(a, b, m, s)$ is known) and on more general cases, where the value of the integral is obtained by Monte Carlo integration. The results indicate that the 4-term EQA is accurate to 3 to 4 significant figures, while being at least 6 orders of magnitude more efficient than Monte Carlo (MC) integration performed to a similar accuracy. In section 5.3, we demonstrate how the 4-term EQA for $\mathrm{erf}(x)$ can be applied to evaluate the integral (2) used to model the average symbol error probability in the $m = 1, 2, 3$ cases. We draw conclusions and discuss further directions for the work in section 6. A listing of the symbolic algebra code (in Maple/Matlab format) appears in the Appendix.

2 Known Approximations for $Q(x)$ and $\mathrm{erf}(x)$

The two standard power series for the error function contained in Abramowitz & Stegun [18] are:

$$\mathrm{erf}(x) = \frac{2x}{\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n+1)n!}; \tag{5}$$

$$\mathrm{erf}(x) = \frac{2x\mathrm{e}^{-x^2}}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(2x^2)^n}{(2n+1)!!}, \tag{6}$$

where $(2n+1)!! = 1 \cdot 3 \cdot 5 \cdots (2n+1)$. There is also an asymptotic expansion for the complementary error function $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$:

$$\mathrm{erfc}(x) = \frac{\mathrm{e}^{-x^2}}{x\sqrt{\pi}} \left(1 + \sum_{n=1}^{N-1} (-1)^n \frac{(2n-1)!!}{(2x^2)^n}\right) + \epsilon_N(x), \tag{7}$$

where $\epsilon_N(x)$ is the remainder term, which is smaller in magnitude than the last term in the series and can be used to achieve a target precision. According to [5], series (5) is useful for small arguments but is ill conditioned for large arguments because it is alternating and suffers from "catastrophic cancellation," where the leading bits of adjacent terms cancel after subtraction. The asymptotic expansion is often the fastest to compute for a given target precision, when this is achievable.

For the Gaussian Q-function, examples of Mills' ratio approximations include [19, 20]:

$$Q_{\text{Borjesson 1}}(x) = \frac{1}{\sqrt{2\pi}} \frac{\mathrm{e}^{-x^2/2}}{0.661x + 0.339\sqrt{x^2 + 5.51}}; \tag{8}$$

$$Q_{\text{Borjesson 2}}(x) = \frac{1}{\sqrt{2\pi}} \frac{\mathrm{e}^{-x^2/2}}{\sqrt{x^2+1}}; \tag{9}$$

$$Q_{\text{Jang}}(x) = \frac{1}{\sqrt{2\pi}} \frac{\left(1 - \mathrm{e}^{-\sqrt{\frac{\pi}{2}}x}\right)}{x} \mathrm{e}^{-x^2/2}; \tag{10}$$

where each approximation has been subscripted to denote its authorship. Jang's approximation is closely related to an earlier one in [21]. Examples of sums of exponential approximations include [22–25]:

$$Q_{\text{Loskot 2}}(x) = 0.208\,\mathrm{e}^{-0.971x^2} + 0.147\,\mathrm{e}^{-0.525x^2}; \tag{11}$$

$$Q_{\text{Loskot 3}}(x) = 0.168\,\mathrm{e}^{-0.876x^2} + 0.144\,\mathrm{e}^{-0.525x^2} + 0.002\,\mathrm{e}^{-0.603x^2}; \tag{12}$$

$$Q_{\text{Benitez M}}(x) = \mathrm{e}^{-0.4920x^2 - 0.2887x - 1.1893}; \tag{13}$$

$$Q_{\text{Benitez S}}(x) = \mathrm{e}^{-0.3842x^2 - 0.7640x - 0.6964}; \tag{14}$$

$$Q_{\text{Chiani}}(x) = \tfrac{1}{12}\mathrm{e}^{-x^2/2} + \tfrac{1}{4}\mathrm{e}^{-2x^2/3}; \tag{15}$$

$$Q_{\text{Sofotasios}}(x) = 0.49\,\mathrm{e}^{-8x/13}\,\mathrm{e}^{-x^2/2}; \tag{16}$$

where "M" refers to minimisation of the maximum absolute error (MARE) and "S" refers to sum of squared error (SSE) in the Benitez approximation. Two-term exponential-quadratic approximations to $Q(x)$ similar to (11) and (15) have also been used to evaluate (1) by representing $Q(\sqrt{x})$ as a sum of two exponentials in a linear argument [26]. We mention two further, higher-order, exponential approximations [27, 28]:

$$Q_{\text{Dao}}(x) = \exp\left(\sum_{i=0}^{6} a_i x^i\right); \tag{17}$$

$$\mathrm{erf}_{\text{VanHalen}}(x) = 1 - \exp\left(\sum_{i=0}^{10} a_i x^i\right). \tag{18}$$

According to [16] (Figure 9), Dao et al.'s approximation for $Q(x)$ is accurate to around 1×10^{-4} on $x \in [0.2,\ 4.5]$ rising to about 6×10^{-4} at $x = 0$. Van Halen's approximation (18) to $\mathrm{erf}(x)$ is accurate to 1.6×10^{-9} for $x \geq 0$. The $\{a_i\}$ coefficient values for each expression are in the respective references.

A further class of approximations to the error function is based on sums products of either rational functions or polynomials and e^{-x^2}. The following rational-times-exponential approximation, mentioned in Abramowitz & Stegun, is well known:

$$\mathrm{erf}(x) \approx 1 - \left[\sum_{k=1}^{5} \frac{a_k}{(1 + p\,x)^k}\right] \mathrm{e}^{-x^2}, \tag{19}$$

and provides an absolute relative error bound of 8.09×10^{-6} for $x \geq 0$. Howard [29] applies splines in deriving a class of polynomial-times-exponential approximations of the form:

$$\mathrm{erf}_{\mathrm{H}}(x) = \frac{2}{\sqrt{\pi}} \sum_{k=1}^{n} c_k\, x^{k+1} \left[p_k(0) + (-1)^k p_k(x) \mathrm{e}^{-x^2} \right], \quad (20)$$

where n is the order of the approximation and $p_k(x)$ is a polynomial of degree $2n$ with only even powers. Although Howard's approximations have the desirable property that $\mathrm{erf}_{\mathrm{H}}(0) = 0$, an order of $n = 10$ (polynomial degree 20) is required to obtain a relative error of around 1×10^{-4} on $x \in [0,\ 4]$, with the error increasing for $x > 4$.

3 New Exponential Sum Approximations

3.1 Unconstrained Case

Of the approximations to the error function and Gaussian Q-function given in section 2, only sums of exponentials of second order polynomials are suitable candidates for the evaluation of the main types of integrals (2) and (3) considered in the introduction. However, the exponential quadratic functions in (11)–(16) have been developed for *asymptotic* approximation of the Q-function or error function and provide a poor match for small-to-medium arguments and in particular at the origin. Such EQAs are therefore unsuitable for approximation on the real line. Nonetheless, for what follows, a demonstration of how to obtain these approximations is necessary.

We therefore present, as an initial illustration of the method, a simple sum-of-exponentials approximation for $\mathrm{erf}(x)$, directly applicable to $\mathrm{Q}(x)$, and show how to optimise its parameters. This is followed by more complicated but more accurate approximations. As previously mentioned, and following [22], we consider an M-term Prony-type approximation for $\mathrm{erf}(x)$ of the form

$$\widehat{\Phi}_M(x) = 1 - \sum_{i=1}^{M} c_i \mathrm{e}^{-a_i x^2},\ x \in R, \quad (21)$$

where $\{c_i\}$ and $\{a_i > 0\}$, $i = 1, \ldots, M$, are $2M$ real parameters to be estimated. The first derivative of this function with respect to x is given by

$$\widehat{\Phi}'_M(x) = 2 \sum_{i=1}^{M} a_i c_i x \mathrm{e}^{-a_i x^2}, \quad (22)$$

Clearly, this function has the drawback that $\widehat{\Phi}'_M(0)$=0 regardless of the value of M, whereas we know that the slope of $\mathrm{erf}(x)$ at $x = 0$ is $2/\sqrt{\pi} \approx 1.12838$. Nonetheless, this functional form is convenient for demonstrative purposes and leads naturally to the more complicated cases.

Example 1 (M terms, unconstrained). *Let the approximator be defined as in (21). Consider the sum-of-squares cost function defined for a sample of x-axis points ("knots")* $\mathbf{x} = [x_1, \ldots, x_N]$ *with corresponding exact function values* $[y_1, \ldots, y_N]$ *given by*

$$S(\mathbf{x}; \mathbf{a}, \mathbf{c}) = \tfrac{1}{2} \sum_{n=1}^{N} \left(y_n - \widehat{\Phi}_M(x_n) \right)^2, \quad (23)$$

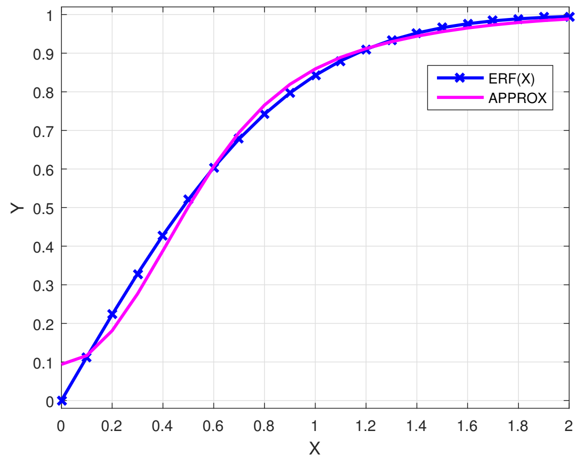

Figure 1. Unconstrained 3-term exponential-quadratic approximation to the error function.

where $\mathbf{a} = [a_1, \ldots, a_M]$ *and* $\mathbf{c} = [c_1, \ldots, c_M]$ *are the parameter vectors. In the case of gradient descent (GD) optimisation, which we assume here, the first derivatives of the cost function with respect to the parameters are required. These are furnished by:*

$$\frac{\partial S(\mathbf{x}; \mathbf{a}, \mathbf{c})}{\partial a_j} = -\sum_{n=1}^{N} \left(y_n - \widehat{\Phi}_M(x_n) \right) c_j x_n^2 \mathrm{e}^{-a_i x^2},$$

$$\frac{\partial S(\mathbf{x}; \mathbf{a}, \mathbf{c})}{\partial c_j} = \sum_{n=1}^{N} \left(y_n - \widehat{\Phi}_M(x_n) \right) \mathrm{e}^{-a_i x^2}.$$

The resulting gradient descent optimisation iteration (over k) is given for $j = 1, \ldots, M$ *by*

$$a_j(k+1) = a_j(k) - \eta \frac{\partial S(\mathbf{x}; \mathbf{a}(k), \mathbf{c}(k))}{\partial a_j},$$

$$c_j(k+1) = c_j(k) - \eta \frac{\partial S(\mathbf{x}; \mathbf{a}(k), \mathbf{c}(k))}{\partial c_j},$$

where $\eta > 0$ *is a step size parameter. The iteration is initialised with suitable set of initial parameter values* $\mathbf{a}(0)$ *and* $\mathbf{c}(0)$*. The GD equations are highly nonlinear and there is no guarantee of convergence or uniqueness of the solution. This is typical for parameter estimation of arbitrary sums of exponential functions estimated by gradient descent. (Note the approximator is not a Gaussian mixture, to which the expectation maximisation (EM) algorithm can be applied.)*

Figure 1 shows the behaviour of the unconstrained 3-term exponential quadratic approximator, carried out in Matlab. The knot points (blue crosses) were taken to be uniformly spaced on [0, 2] with spacing 0.1. The gradient descent algorithm was initialised with $\mathbf{c}(0) = [0.336,\ 0.288,\ 0.004]$ and $\mathbf{a}(0) = [1.752,\ 1.05,\ 1.206]$, corresponding to the 3-term Loskot exponential approximator (12). GD was run for 1000 iterations with constant step size $\eta = 0.2$, converging rapidly at first, then very gradually. While the optimisation is sensitive to the initial parameter settings, the approximation to the error function (shown in blue) is generally poor, particularly around the origin. Matters do not improve significantly by adding more knot points or by tweaking the optimisation parameters.

3.2 3 Terms, 3 Constraints

The preceding example shows that a simple sum-of-squares cost function that includes error terms at a finite set of knots can result in significant errors by failing to match the slope of the error function at the ends of the interval. The next example attempts to correct this deficiency by adding constraints on the approximator to match the function values at the end point and the first derivative at the right end point of the interval, which is taken as [0, 1] for simplicity. We choose $M = 3$ terms to match the number of constraints. This results in a simple linear system for the multipliers (c_1, c_2, c_3) that allows the constrained 6-parameter estimation problem to be rewritten as an unconstrained 3-parameter estimation problem to which gradient descent optimisation can be applied.

Example 2 ($M = 3$ terms, 3 constraints)**.** *Let the approximator be defined as in (21) with $M = 3$ terms and the sum-of-squares cost function defined in (23). Suppose further that the approximator satisfies the following constraints*

$$\begin{aligned}\widehat{\Phi}_3(0) &= \Phi(0) = 0,\\ \widehat{\Phi}_3(1) &= \Phi(1),\\ \widehat{\Phi}_3'(1) &= \Phi'(1).\end{aligned}$$

It is easy to show that these constraints are equivalent to the system of equations

$$\begin{bmatrix} 1 & 1 & 1 \\ e^{-a_1} & e^{-a_1} & e^{-a_1} \\ a_1 e^{-a_1} & a_2 e^{-a_1} & a_3 e^{-a_1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ q \\ r \end{bmatrix}, \tag{24}$$

where $q \stackrel{\triangle}{=} 1 - \Phi(1)$ and $r \stackrel{\triangle}{=} \frac{1}{2}\Phi'(1) = 1/(e\sqrt{\pi})$. We can solve the linear system to obtain the c_i in terms of $\mathbf{a} = (a_1, a_2, a_3)$. The result is

$$\begin{aligned}\tilde{c}_1(\mathbf{a}) &= e^{a_1}\left((qa_3 - r)e^{a_2} + (r - qa_2)e^{a_3} + a_2 - a_3\right),\\ \tilde{c}_2(\mathbf{a}) &= e^{a_2}\left((r - qa_3)e^{a_1} + (qa_1 - r)e^{a_3} + a_3 - a_1\right),\\ \tilde{c}_3(\mathbf{a}) &= e^{a_3}\left((qa_2 - r)e^{a_1} + (r - qa_1)e^{a_2} + a_1 - a_2\right),\\ \Delta(\mathbf{a}) &= (a_2 - a_3)e^{a_1} + (a_3 - a_1)e^{a_2} + (a_1 - a_2)e^{a_3},\\ c_i(\mathbf{a}) &\stackrel{\triangle}{=} \tilde{c}_i(\mathbf{a})/\Delta(\mathbf{a}),\ i = 1, 2, 3.\end{aligned}$$

Eliminating the c_i from (21) allows the approximator to be written as

$$\widehat{\Phi}_3(x; \mathbf{a}) = 1 - \textstyle\sum_{i=1}^{3} c_i(\mathbf{a}) e^{-a_i x^2}. \tag{25}$$

Estimating the remaining parameters (a_1, a_2, a_3) is straightforward since the equality constraints have been eliminated. Gradient descent can be applied as before once the partial derivatives of $\widehat{\Phi}_3(\mathbf{x}; \mathbf{a})$ with respect to the a_i are computed. This tedious calculation is best done using symbolic algebra (or other automatic differentiation software). As an example, the partial derivative of $\widehat{\Phi}_3(x; \mathbf{a})$ with respect to a_1, i.e., *$\partial\widehat{\Phi}_3(x; \mathbf{a})/\partial a_1$, can be shown to be:*

$$\begin{aligned}&\frac{C}{A}\left(\frac{B}{A} - 1 + x^2\right)\phi_1(x)\\ &+ \frac{1}{A}\left((qa_3 - r)(1 - \frac{B}{A})e^{a_1} - De^{a_3} + \frac{B}{A}(a_3 - a_1)\right)\phi_2(x)\\ &+ \frac{1}{A}\left((qa_2 - r)(\frac{B}{A} - 1)e^{a_1} + De^{a_2} + \frac{B}{A}(a_1 - a_2)\right)\phi_3(x),\end{aligned}$$

where $\phi_i(x) = \exp(a_i(1 - x^2))$, $i = 1, 2, 3$, and the terms A, B, C, D are defined by

$$\begin{aligned}A &= (a_2 - a_3)e^{a_1} + (a_3 - a_1)e^{a_2} + (a_1 - a_2)e^{a_3},\\ B &= (a_2 - a_3)e^{a_1} + e^{a_3} - e^{a_2},\\ C &= (qa_3 - r)e^{a_2} + (r - qa_2)e^{a_3} + a_2 - a_3,\\ D &= q + B(r - qa_1)/A.\end{aligned}$$

The derivative of the cost function S given by (23), using $\widehat{\Phi}_3(x; \mathbf{a})$ in place of $\widehat{\Phi}_M(x)$, is:

$$\frac{\partial S(\mathbf{x}; \mathbf{a})}{\partial a_j} = -\sum_{n=1}^{N}\left(y_n - \widehat{\Phi}_3(x_n; \mathbf{a})\right)\frac{\partial\widehat{\Phi}_3(x_n; \mathbf{a})}{\partial a_j}.$$

The resulting GD iteration (over k) is given for $j = 1, 2, 3$ by

$$a_j(k+1) = a_j(k) - \eta\frac{\partial S(\mathbf{x}; \mathbf{a}(k))}{\partial a_j},\ \eta > 0,\ k = 0, 1, 2, \ldots$$

The results of a numerical simulation for the constrained 3-term exponential approximator in Example 2 are shown in Figures 2–4. The gradient descent algorithm was initialised with $\mathbf{a}(0) = [1.752,\ 1.05,\ 1.206]$. The knot points x_n, $n = 1, \ldots, N = 101$ were taken to be uniformly spaced on [0, 1] with spacing 0.01. The GD algorithm was run for 1000 iterations with constant step size $\eta = 0.2$. From this initial configuration, convergence of the cost function to 10^{-11} was observed after 100 iterations. The estimated values for the exponential parameters were $\mathbf{a} = [3.4441,\ 2.7421,\ 2.8981]$ with corresponding weights $\mathbf{c} = [18.119,\ 56.133,\ -73.252]$. Along with the 3-term approximation to the error function, several other approximations are shown in Figure 2. All of these are based on single or a sum of two exponentials of quadratic arguments. The 3-term approximator exactly matches the error function at the constraint points 0 and 1 but there is a significant mismatch in the slope at $x = 0$, although the fit is better than the Loskot 3-term and Chiani approximations for $x \in [0,\ 1]$. The best fit for this range of x values is the Benitez estimator with Sofotasios in second place. This conclusion is clearer from the relative absolute error (ARE) plot in Figure 3, which also shows the large argument behaviour: all estimators considered achieve 10^{-6} ARE for $x \geq 3.5$. The same 3-term approximator can be directly used to approximate the Q function. The corresponding ARE plots appear in Figure 4. The 3-term approximator performs worst for large arguments of the Q function, which is unsurprising given the small fitting interval of [0, 1].

3.3 4 Terms, 4 Constraints

With the preceding two examples in hand, it is is natural to impose a fourth constraint on the approximator to match both the function values and the first derivatives at each end of the interval. This is, of course, the principle of the smoothing spline [30], with the principal difference being the approximation function is not polynomial. As mentioned in section 3.1, the approximator in (21) has zero derivative at $x = 0$, i.e, $\widehat{\Phi}_M'(0) = 0$, and cannot match the derivative of the

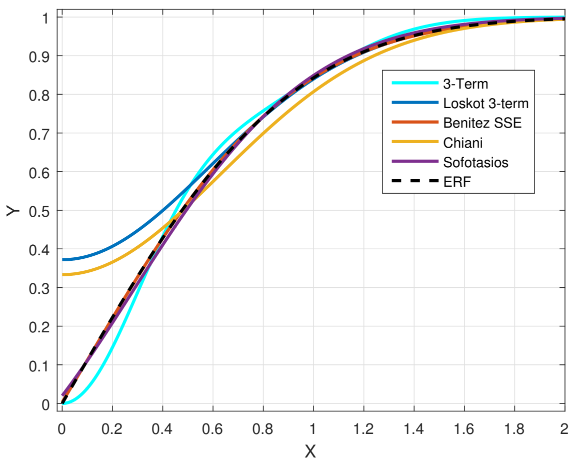

Figure 2. Various exponential-quadratic approximations to the error function.

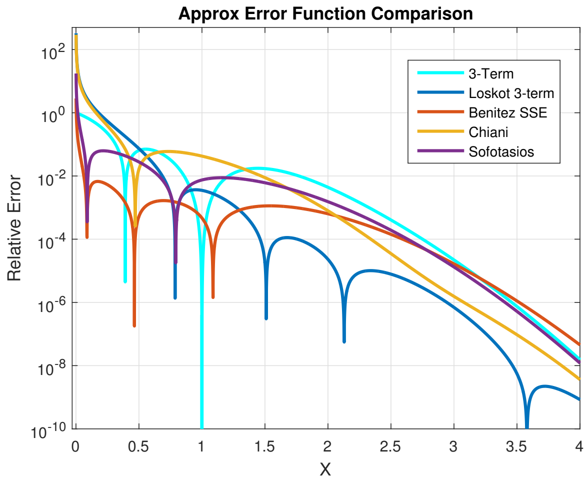

Figure 3. Absolute relative error for exponential-quadratic approximations to error function.

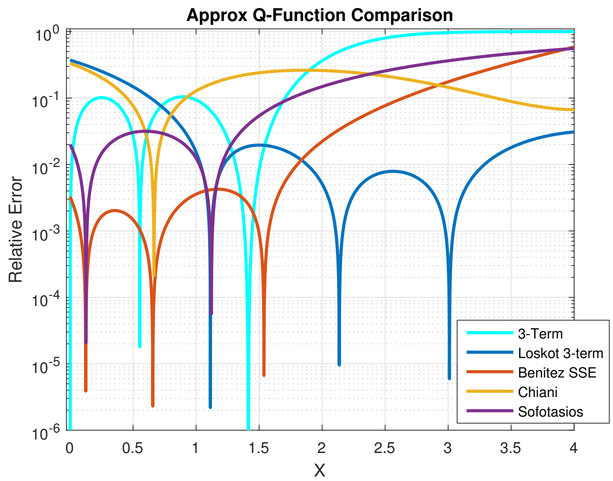

Figure 4. Absolute relative error for exponential-quadratic approximations to Q-function.

error function (or the Q function) at $x = 0$. We therefore mildly generalise the exponential to include a linear term, considering approximators of the form:

$$\widehat{\Phi}_M(x;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = 1 - \sum_{i=1}^{M} c_i e^{-a_i x^2 + 2b_i x},\ x \in R. \quad (26)$$

which has $3M$ parameters to be estimated, namely the $\{a_i, b_i, c_i\}$, $i = 1, \ldots, M$ where, as before we assume that the $a_i > 0\ \forall i$. The added degrees of freedom can then be used to match the first derivative at both ends of the interval, which we also extend to $[0,\ u]$ where $u \geq 1$. In practice, as we will see, u can be used to set a rough upper bound on the approximation error for large arguments of the error function or Q function since these functions are asymptotically equal to 1 and 0 respectively. The previous method for converting the constrained nonlinear optimisation problem into an unconstrained one is applicable if we take $M = 4$ terms, equal to the number of constraints. This 4-term exponential-quadratic approximator, which we refer to as the 4-term EQA, is dealt with in the following theorem. For reasons of algebraic complexity, certain explicit expressions have been omitted.

Theorem 3.1 ($M = 4$ terms, 4 constraints)**.** *Let the approximator be defined as in (26) with $M = 4$ terms and the sum-of-squares cost function defined below. The constrained nonlinear optimisation problem for $\{a_i, b_i, c_i\}$, $i = 1, \ldots, 4$ given by*

$$\min_{\{\mathbf{a},\mathbf{b},\mathbf{c}\}} S(\mathbf{x};\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = \tfrac{1}{2} \sum_{n=1}^{N} \Big(y_n - \widehat{\Phi}_4(x_n;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) \Big)^2. \quad (27)$$

subject to the constraints

$$(C1)\ \widehat{\Phi}_4(0;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = \Phi(0),$$
$$(C2)\ \widehat{\Phi}_4(u;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = \Phi(u),$$
$$(C3)\ \widehat{\Phi}'_4(0;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = \Phi'(0),$$
$$(C4)\ \widehat{\Phi}'_4(u;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = \Phi'(u),$$

where $u > 0$ is the upper limit of the knot values $\mathbf{x} = [x_1, \ldots, x_N]$, with $x_i \in [0, u]$, can be reexpressed as an unconstrained nonlinear optimisation problem for $\{a_i, b_i\}$, $i = 1, \ldots, 4$, as:

$$\min_{\{\mathbf{a},\mathbf{b}\}} S(\mathbf{x};\, \mathbf{a}, \mathbf{b}, \mathbf{c}(\mathbf{a}, \mathbf{b})) = \tfrac{1}{2} \sum_{n=1}^{N} \Big(y_n - \widehat{\Phi}_4(x_n;\, \mathbf{a}, \mathbf{b}, \mathbf{c}(\mathbf{a}, \mathbf{b})) \Big)^2, \quad (28)$$

where $\mathbf{c}(\mathbf{a}, \mathbf{b}) = [c_1(\mathbf{a}, \mathbf{b}), \ldots, c_4(\mathbf{a}, \mathbf{b})]^T$ is the unique solution (when it exists) of the 4-dimensional linear system

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ d_1 & d_2 & d_3 & d_4 \\ a_1 & a_2 & a_3 & a_4 \\ e_1 & e_2 & e_3 & e_4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 \\ q \\ s \\ r \end{bmatrix}, \quad (29)$$

where $q \triangleq 1 - \Phi(u)$, $r \triangleq \frac{1}{2}\Phi'(u) = \exp(-u^2)/\sqrt{\pi}$, and $s \triangleq -\frac{1}{2}\Phi'(0) = -1/\sqrt{\pi}$, and for $i = 1, \ldots, 4$ the d_i and e_i are defined by $d_i = \exp(-a_i u^2 + 2b_i u)$ and $e_i = (a_i u - b_i) d_i$.

The corresponding gradient descent algorithm is given for parameter index $j = 1, \ldots, 4$, step size $\eta > 0$ and iteration $k = 0, 1, 2, \ldots$ by

$$a_j(k+1) = a_j(k) + \eta\, [\mathbf{y} - \Psi_k(\mathbf{x})]^T \frac{\partial}{\partial a_j} \Psi_k(\mathbf{x}), \quad (30)$$

$$b_j(k+1) = b_j(k) + \eta\, [\mathbf{y} - \Psi_k(\mathbf{x})]^T \frac{\partial}{\partial b_j} \Psi_k(\mathbf{x}), \quad (31)$$

where $\mathbf{y}$ *is the N-vector of exact function values* $[\Phi(x_1), \ldots, \Phi(x_N)]^T$ *corresponding to the knots, and* $\Psi_k(\mathbf{x})$ *is given by*

$$\Psi_k(\mathbf{x}) = \begin{bmatrix} \widehat{\Phi}_4(x_1;\, \mathbf{a}(k), \mathbf{b}(k), \mathbf{c}(\mathbf{a}(k), \mathbf{b}(k))) \\ \vdots \\ \widehat{\Phi}_4(x_N;\, \mathbf{a}(k), \mathbf{b}(k), \mathbf{c}(\mathbf{a}(k), \mathbf{b}(k))) \end{bmatrix}.$$

Proof: Treating the $M = 4$ case, constraint (C1) gives

$$1 - \sum_{i=1}^{4} c_i = 0, \tag{32}$$

while constraint (C2) gives

$$1 - \sum_{i=1}^{4} c_i \exp(-a_i u^2 + 2b_i u) = \Phi(u), \tag{33}$$

or $\sum_{i=1}^{4} d_i c_i = q$ according to the notation in the theorem. For the last two constraints, we need the derivative of the approximator function:

$$\widehat{\Phi}'_M(x;\, \mathbf{a}, \mathbf{b}, \mathbf{c}) = 2\sum_{i=1}^{M} (a_i x - b_i) c_i e^{-a_i x^2 + 2b_i x}. \tag{34}$$

So that constraint (C3) leads to

$$-2\sum_{i=1}^{4} a_i c_i = \Phi'(0) = \frac{2}{\sqrt{\pi}}, \tag{35}$$

while constraint (C4) leads to a constraint at $\Phi'(u)$:

$$2\sum_{i=1}^{4} (a_i u - b_i) c_i \exp(-a_i u^2 + 2b_i u) = \frac{2e^{-u^2}}{\sqrt{\pi}}, \tag{36}$$

or $\sum_{i=1}^{4} e_i c_i = r$ according to the definitions in the theorem. Equations (32)-(36) therefore satisfy the linear system (29).

We then solve the linear system to obtain the c_i in terms of $\mathbf{a} = (a_1, a_2, a_3, a_4)$ and $\mathbf{b} = (b_1, b_2, b_3, b_4)$. This straightforward but tedious calculation is best handled using symbolic algebra. This is followed by substitution to replace $\mathbf{c}$ by $\mathbf{c}(\mathbf{a}, \mathbf{b})$ in (26), yielding the components of the vector $\Psi(x)$. The gradient descent algorithm in (30) is formally obtained by the chain rule from the cost function definition in (28). ■

Remarks

1) The partial derivatives are very complicated, each containing more than 1000 exponential terms in unoptimised form. They can be obtained and converted to C code using standard functionalities from the Matlab Symbolic Toolbox (based on the Maple kernel). It is not practical to reproduce these equations here due to their length.
2) The Matlab Symbolic Toolbox commands for generating the derivatives code for the gradient descent algorithm for the 4-term approximator can be found in the Appendix. This code requires only minor rework (search and replace) to produce working Matlab script.
3) The gist of the gradient descent algorithm here is similar to the deep learning paradigm, which uses algorithmic differentiation to compute derivatives of the standard functional blocks that comprise the convolutional neural network during the "learning phase", i.e., parameter estimation stage.

4 Optimisation & Performance Comparison

The numerical experiments for the constrained 4-term exponential-quadratic approximator in Theorem 3.1 are dealt with in this section. These results were obtained after around 100 trials with different initialisation settings for $\mathbf{a}(0)$, $\mathbf{b}(0)$, different upper limits u, different numbers of knots and different knot values. Although the step size $\eta = 0.2$ was not varied, differing number of steps were tried. For each algorithm configuration, the maximum absolute error (MAE) and the maximising x-value were calculated on a grid of values from 0 to 5 with spacing $1E-05$. The minimum value of the MAE was used to pinpoint the "best" algorithm configuration, whose parameters are now described.

The gradient descent algorithm was initialised with $\mathbf{a}(0) = [1.3,\ 0.8,\ 1,\ 0.5]$ and $\mathbf{b}(0) = [\ -0.1,\ -0.1,\ 0,\ 0.4]$. The knot points x_n were taken to be $[0.1, 0.2, 0.3, 0.6, 0.8, 1.2, 1.5]$ with upper limit $u = 4$. The GD algorithm was run for 11000 iterations with constant step size $\eta = 0.2$. In this case, the MAE was 0.00016499 (increasing slightly to 0.00016601 at 12000 iterations). The estimated values for the exponential parameters $\mathbf{a}$, $\mathbf{b}$ and weights $\mathbf{c}$ appear in Table I. The simulation results appear in Figures 5–7. Each one of these figures gives the results for the 4-term EQA and 9 other known approximations to the Q function or error function (as appropriate). The additional derivative constraint at $x = 0$ greatly improves the quality of the approximation

Table I
Constrained 4-Term EQA Parameters

	term 1	term 2	term 3	term 4
a	+1.102149	+0.602149	+0.802149	+0.302149
b	−0.738479	−0.738479	−0.638479	−0.238479
c	−0.656344	$-8.65439E-2$	+1.742885	$2.31093E-6$

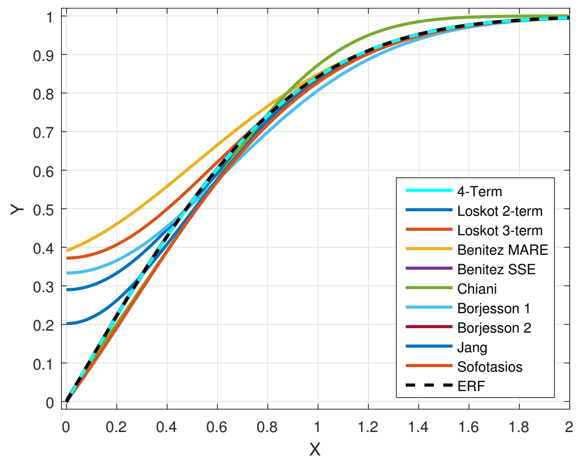

Figure 5. Comparison of approximations to the error function.

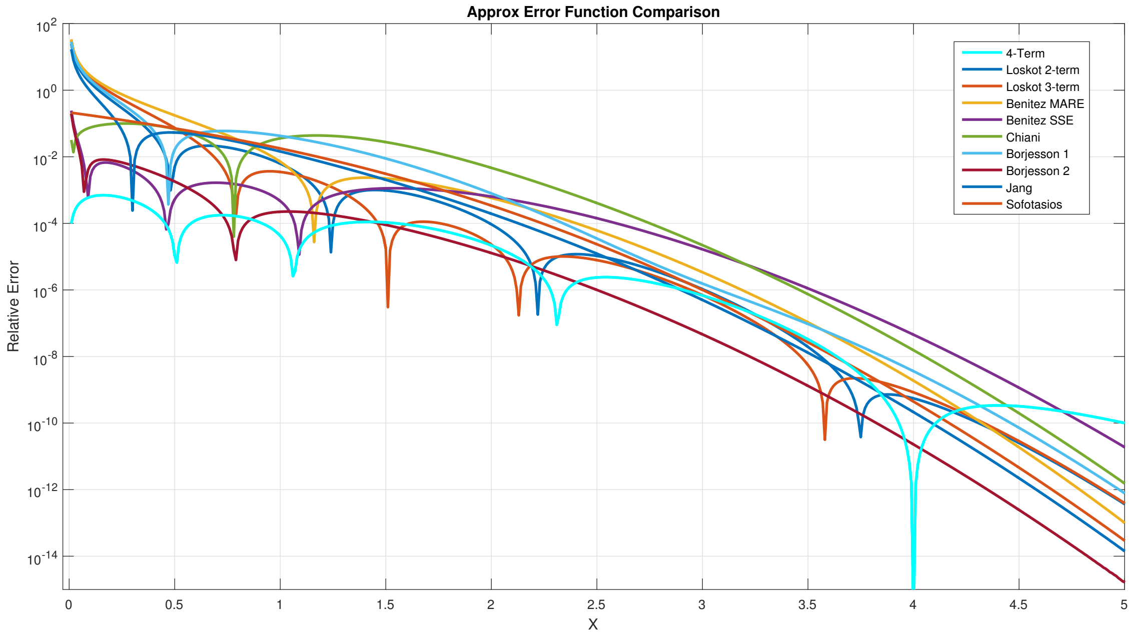

Figure 6. Absolute relative error for approximations to error function.

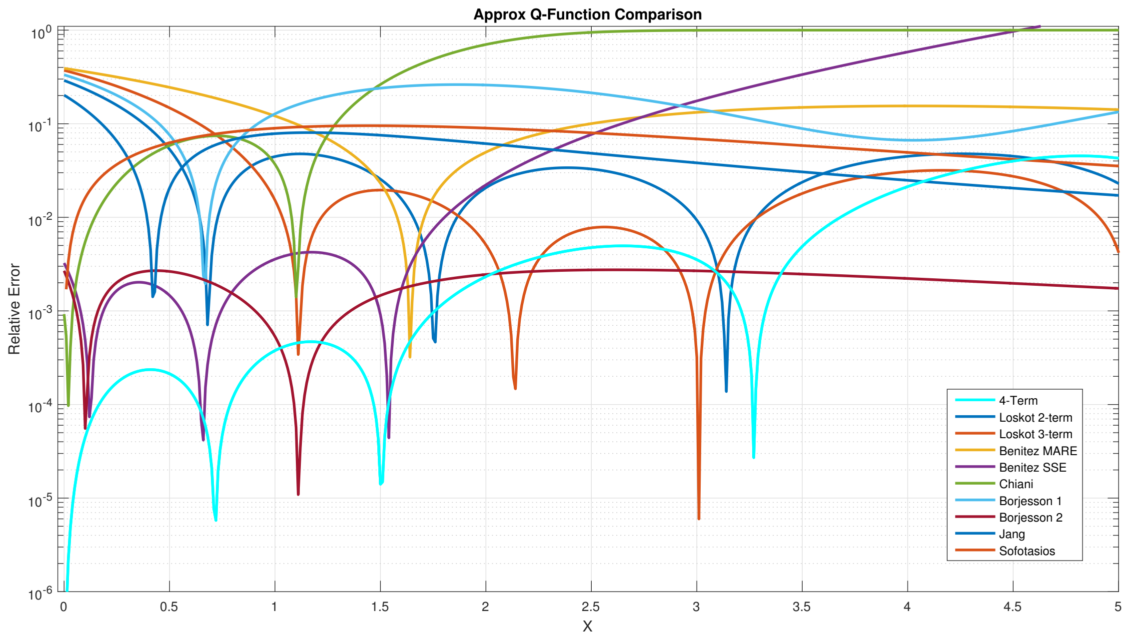

Figure 7. Absolute relative error for approximations to Q-function.

over the 3-term approximator in Example 2, making the 4-term approximator competitive with the other methods for x values up to the upper limit of $u = 4$, where (according to Matlab) the error function differs from unity by only $1.5417E-08$. The approximation quality is particularly good for smaller values in the range $0 \leq x \leq 1.5$ for the error function and $0 \leq x \leq 2$ for the Q function. The approximation can also be used for negative arguments since $\text{erf}(x)$ is an odd function, which makes it valid on the entire real line.

Figure 5 overlays 10 different approximations to the error function including the 4-term EQA is shown in cyan. The error function computed by Matlab's built-in library function appears as a black dashed line. The latter is used as "ground truth" for the error function to obtain absolute relative error estimates in Figures 6 and 7, which display, respectively, the ARE for the error function approximations and for the Q function. The same colour coding is used in these plots. The nine other approximations are: (1) Loskot 2-term; (2) Loskot 3-term; (3) Benitez MARE; (4) Benitez SSE; (5) Chiani; (6) Borjesson 1; (7) Borjesson 2; (8) Jang; (9) Sofotasios; where the approximator labels have the same meanings as in equations (8)–(16) and also as in Bao et al.'s paper [16] and so are directly comparable. The 4-term EQA provides a very good fit to the error function over the range $0 \leq x \leq 4$, particularly for small arguments $x < 0.6$ where five other approximations (Benitez

MARE, Loskot 3-term, Borjesson 1, Loskot 2-term and Jang) give a poor fit. The Chiani approximation is a poor fit for $0.8 < x < 2$. For small arguments, the best fitting approximations are (in order of best to worst): 4-term EQA, Benitez SSE, Borjesson 2, Chiani, Sofotasios, with the first 3 of these visually close in Figure 5. The clear superiority of the 4-term EQA for $x < 1.43$ can be seen in the ARE curves in Figure 6; its peak ARE is just under $7E{-}4$, occurring close to $x = 0.16$. It is also the only approximator of those tested that exactly matches (to around $2E{-}15$) the zero-value of the error function at $x = 0$. The next closest is Chiani, which is $9.244E{-}4$ at $x = 0$. For large arguments ($x > 4.7$) the 4-term EQA provides less accuracy in ARE than all the others, bearing in mind that the peak ARE is already under $3.4E{-}10$ for $x > 4$. Similar statements can be made for the Q function approximations in Figure 7: the 4-term EQA is better than the other nine approximations for $0 \le x \le 2.025$ (approximately) and better than 8 others up to $x = 4.07$, where it loses out to Jang. The Borjesson 2 is also closer to the Q function in ARE in a narrow range around $x = 1.1$.

5 Application Examples

5.1 Approximation of the Key Integral for $n = 1$

We now return to the motivation given in the introduction, namely, the evaluation of integrals of the form (2) and (3) via the integrals $\mathbf{I}^{(n)}(a, b, m, s)$ in equation (4). In the absence of a numerically well behaved, analytical formula for $\mathbf{I}^{(n)}(a, b, m, s)$, the brute-force method is to use Monte Carlo integration, which works by averaging randomly generated values of the integrand in its support region. The integrand decreases rapidly for $x > m + p\sqrt{s}$ where p is the number of standard deviations away from the mean m, so the support for the integral can be taken as $[0, m + 5\sqrt{s}]$. Nonetheless, a low variance Monte Carlo estimate of (4) might still require at least 10^6 random samples, which is not efficient, and may also suffer from bias due to imperfections in the random number generator or due to truncation of the infinite domain of integration. Although the integral is bounded, the integrand is unbounded when the Gaussian density is close to singular, i.e., when $s \to 0$.

Theorem 3.1 provides an accurate numerical approximation to the error function based only on exponentials of terms up to second order. This 4-term EQA is specifically tailored to the analytical approximation of integrals of the form (1) and (3). In the general $n > 0$ case, a multinomial expansion of (26) is required to compute the n-th power of the approximation of $\operatorname{erf}^n(x)$. There is no difficulty in principle, since all resulting terms will still be of quadratic exponential type, but their number increases rapidly with n. In this application example, we focus on the $n = 1$ case only, writing $\mathbf{I}(a, b, m, s)$ in place of $\mathbf{I}^{(1)}(a, b, m, s)$, and similarly for its approximation.

Replacing the error function in (4) by its 4-term EQA from (26) with $M = 4$, we define an approximation to the former for $a,\ s > 0$ by

$$\widehat{\mathbf{I}}(a, b, m, s) = \frac{1}{\sqrt{2\pi s}} \int_0^\infty \widehat{\Phi}_4(ax + b) \exp\left[\frac{-(x - m)^2}{2s}\right] dx. \tag{37}$$

Consider the integrand

$$\widehat{\Phi}_4(ax + b) \exp\left[\frac{-(x - m)^2}{2s}\right] = \mathrm{e}^{\frac{-(x-m)^2}{2s}} - \sum_{i=1}^{4} c_i \mathrm{e}^{F_i(x)},$$

where $F_i(x)$ is defined as

$$-(a_i a^2 + \frac{1}{2s})x^2 + 2(ab_i - aba_i + \frac{m}{2s})x - a_i b^2 + 2bb_i - \frac{m^2}{2s}.$$

We have implicitly assumed that $ax + b \ge 0$. This is the same as $b \ge 0$ since $x \ge 0$ and $a > 0$. Defining new variables as

$$\begin{aligned}
\alpha_i^2 &= a_i a^2 + \frac{1}{2s}, \\
\beta_i^+ &= ab_i - aba_i + \frac{m}{2s}, \\
\gamma_i^+ &= c_i \exp\left(2bb_i - a_i b^2 - \frac{m^2}{2s}\right),
\end{aligned}$$

where the $+$ superscript denotes the $b \ge 0$ case, we can write

$$\widehat{\mathbf{I}}^+(a, b, m, s) = \int_0^\infty \mathrm{N}(x; m, s)\, dx - \frac{1}{\sqrt{2\pi s}} \sum_{i=1}^{4} \gamma_i^+ \int_0^\infty \mathrm{e}^{-\alpha_i^2 x^2 + 2\beta_i^+ x}\, dx.$$

It is possible to show, due to the odd symmetry of the error function, that for $s > 0$ and $m \in R$

$$\int_0^\infty \mathrm{N}(x; m, s)\, dx = \tfrac{1}{2}\left(1 + \Phi(\frac{m}{\sqrt{2s}})\right),$$

and further (see [31]), that, with $\alpha > 0$

$$\int_0^\infty \mathrm{e}^{-\alpha^2 x^2 + 2\beta x}\, dx = \frac{\sqrt{\pi}}{2\alpha} \mathrm{e}^{(\beta/\alpha)^2} \left(1 + \Phi(\frac{\beta}{\alpha})\right).$$

Combining these results, we then have (for $b \ge 0$)

$$\widehat{\mathbf{I}}^+(a, b, m, s) = \tfrac{1}{2}\left(1 + \Phi(\frac{m}{\sqrt{2s}})\right) - \sum_{i=1}^{4} \frac{\gamma_i^+ \mathrm{e}^{(\beta_i^+/\alpha_i)^2}}{2\alpha_i\sqrt{2s}} \left(1 + \Phi(\frac{\beta_i^+}{\alpha_i})\right). \tag{38}$$

The $b \le 0$ case is trickier since we need to split the integration domain according to the sign of the argument in $\widehat{\Phi}_4(\cdot)$. On $x \in [-\frac{b}{a}, \infty)$, $ax + b \ge 0$, and we use $\widehat{\Phi}_4(ax + b)$ as before, whereas on $x \in [0, -\frac{b}{a}]$, $ax + b \le 0$, and we must use $-\widehat{\Phi}_4(-ax - b)$. This logic leads to the following expression for $\widehat{\mathbf{I}}^-(a, b, m, s)$:

$$\widehat{\mathbf{I}}^-(a, b, m, s) = \int_{-\frac{b}{a}}^\infty \widehat{\Phi}_4(ax + b)\mathrm{N}(x; m, s)\, dx - \int_0^{-\frac{b}{a}} \widehat{\Phi}_4(-ax - b)\mathrm{N}(x; m, s)\, dx.$$

To proceed further we need the definite integrals below (with $q \ge 0;\ s,\ \alpha > 0$):

$$\int_0^q \mathrm{N}(x; m, s)\, dx = \tfrac{1}{2}\left(\Phi(\frac{q - m}{\sqrt{2s}}) + \Phi(\frac{m}{\sqrt{2s}})\right),$$

Table II
$\mathbf{I}(a,b,m,s)$ Approximator Accuracy with Truth Known

Case	a	b	m	s	$\mathbf{I}$	$\widehat{\mathbf{I}}$	$\lvert\widehat{\mathbf{I}}-\mathbf{I}\rvert$
1	1	0	0	0.5	0.25000	0.25003	2.691E−5
2	$\sqrt{2}$	0	1	0.25	0.84888	0.84895	6.437E−5
3	$\sqrt{2}$	0	1.5	0.25	0.96639	0.96644	4.867E−5
4	$1/\sqrt{2}$	0	1	1	0.57798	0.57803	4.664E−5
5	$1/\sqrt{2}$	0	1.5	1	0.73201	0.73207	5.756E−5

Table III
$\lvert\widehat{\mathbf{I}}-\mathbf{I}_{MC}\rvert$ with Indicated MC Samples (100 Run Average), Truth Unknown

Case	a	b	m	s	$\widehat{\mathbf{I}}$	ΔMC (10^5)	ΔMC (10^6)
6	0.7	−0.5	1.3	0.8	0.38773	2.70E−4	6.51E−5
7	0.7	0.5	1.3	0.8	0.84719	4.39E−4	2.21E−4
8	1	−1	1	1	0.14965	1.14E−4	1.68E−5
9	0.5	−2	1	0.25	−0.93195	3.45E−4	1.05E−4
10	1.5	−1	−2	1.5	−0.012773	1.23E−5	2.45E−6

and

$$\int_0^q \mathrm{e}^{-\alpha^2x^2+2\beta x}\,dx = \frac{\sqrt{\pi}}{2\alpha}\mathrm{e}^{(\beta/\alpha)^2}\left(\Phi(q\alpha-\frac{\beta}{\alpha})+\Phi(\frac{\beta}{\alpha})\right),$$

along with further definitions for β^- and γ^- variables (the α variables are the same as before):

$$\beta_i^- = -ab_i - aba_i + \frac{m}{2s},$$
$$\gamma_i^- = c_i\exp\left(-2bb_i - a_ib^2 - \frac{m^2}{2s}\right).$$

After some further calculations, we obtain $\widehat{\mathbf{I}}^-(a,b,m,s)$ as

$$\begin{aligned}&\tfrac{1}{2}\left(1-\Phi(\frac{m}{\sqrt{2s}})\right)+\Phi(\frac{b/a+m}{\sqrt{2s}})\\ &+\sum_{i=1}^4\frac{\gamma_i^-\,\mathrm{e}^{(\beta_i^-/\alpha_i)^2}}{2\alpha_i\sqrt{2s}}\left(\Phi(\frac{\beta_i^-}{\alpha_i})-\Phi(\frac{\alpha_ib}{a}+\frac{\beta_i^-}{\alpha_i})\right)\\ &-\sum_{i=1}^4\frac{\gamma_i^+\,\mathrm{e}^{(\beta_i^+/\alpha_i)^2}}{2\alpha_i\sqrt{2s}}\left(1+\Phi(\frac{\alpha_ib}{a}+\frac{\beta_i^+}{\alpha_i})\right). \qquad (39)\end{aligned}$$

It is easily verified that the expressions for $\widehat{\mathbf{I}}^-(a,b,m,s)$ (39) and $\widehat{\mathbf{I}}^+(a,b,m,s)$ (38) are identical when $b=0$.

5.2 Numerical Accuracy of $\widehat{\mathbf{I}}(a,b,m,s)$

To verify the accuracy of the approximations in (38) and (39), valid for $n=1$, it is preferable to use cases for which the analytical result is known so as to have a reference value. Unfortunately, after examining the literature on error function integrals [11, 31–33], it seems clear that there are very few cases for which (4) can be exactly evaluated. Two exceptions are presented below; both assume $b=0$. The second case, which generalises the first, can be derived from (2.18.1) in [33].

$$\mathbf{I}(1,0,0,\tfrac{1}{2}) = \frac{1}{\sqrt{\pi}}\int_0^\infty \Phi(x)\,\mathrm{e}^{-x^2}dx = \tfrac{1}{4}\Phi^2(\infty) = \tfrac{1}{4},$$

$$\begin{aligned}\mathbf{I}(a,0,m,\frac{1}{2a^2}) &= \frac{a}{\sqrt{\pi}}\int_0^\infty \Phi(ax)\,\mathrm{e}^{-a^2(x-m)^2}dx\\ &= \tfrac{1}{4}\left(1+\Phi(\frac{ma}{\sqrt{2}})\right)^2.\end{aligned}$$

These integrals are used to produce the results in Table II for 5 different cases where we compare the approximate value from (38) with its known true value. The absolute error in all 5 cases is between 2E−5 and 6.5E−5, which corresponds to 4 significant figure accuracy.

A more challenging set of cases for $b \neq 0$ is presented in Table III. In these cases, there is no known truth since (4) has no closed form. We therefore use Monte Carlo integration with 10^5 and also 10^6 random samples. The truncation point for the uniform distribution is adjusted so that all values of the integrand to the right of it are less than 1E−10. The cases are chosen to cover a range of possible values of $\mathbf{I}(a,b,m,s)$, which is bounded between −1 and +1. The integrand varies from a single positive peak (case 7) or negative peak (case 9), to cases with one positive and one negative peak (cases 6 & 8) and a very slight peak (case 10). The table gives the value of $\widehat{\mathbf{I}}$ from (38) or (39) as appropriate, and, in the last two columns, the absolute error (ΔMC) between $\widehat{\mathbf{I}}$ and the value obtained by averaging 100 Monte Carlo integrations with the indicated number of random samples N_s. For case 7, which is the case with the largest discrepancy of those tested, the (average) standard deviation on the Monte Carlo estimates for $N_s = 10^5$ is around 0.004 and 0.0012 for $N_s = 10^6$. The simulations were conducted in Matlab, using its built-in uniform random number generator and error function. The results confirm the agreement to at least 3 significant figures (discrepancy of the order 10^{-4}) with Monte Carlo integration. Note that MC integration is around $N_s/5$ times more computationally intensive than (38) and $N_s/14$ times more for (39), which is a highly significant saving for the same number of significant figures in accuracy. Moreover, to obtain an averaged MC estimate over N_{MC} runs, the computational complexity is N_{MC} times greater again.

5.3 Communication System Error Probabilities

As touched on in the introduction, integrals of the form (1) arise in modelling the average symbol error probability of certain RF communication systems over fading channels. Changing variables and expanding $Q(x)$ in (2), yields

$$\mathbf{J}_n(a) = \tfrac{1}{2} + \tfrac{1}{2}\sum_{j=1}^{n}(-1)^j \binom{n}{j} \int_0^\infty \Phi^j(\frac{ax}{\sqrt{2}})\mathrm{p}_X(x)\,dx. \tag{40}$$

The evaluation of these integrals is called for when calculating the error probability for M-ary PAM ($n = 1$ case), M-ary (square) QAM and coherent QPSK ($n = 1$ and $n = 2$ cases), and differentially encoded QPSK ($n = 1, 2, 3, 4$) [9].

As an illustrative example, we take the simplest case where $\mathrm{p}_Z(z) = \frac{1}{2}\mathrm{e}^{-z/2}$, and $z \geq 0$ has a χ_2^2 distribution. The transformed variable $x = \sqrt{z}$ then has a Rayleigh PDF, *viz.*: $\mathrm{p}_X(x) = x\,\mathrm{e}^{-x^2/2}$. In this particular case, we can write

$$\mathbf{J}_n(a) = \tfrac{1}{2} + \tfrac{1}{2}\sum_{j=1}^{n}(-1)^j \binom{n}{j} \mathbf{I}_j(a), \tag{41}$$

where $\mathbf{I}_j(a)$ is given by

$$\mathbf{I}_j(a) = \int_0^\infty \Phi^j(\frac{ax}{\sqrt{2}})\,x\,\mathrm{e}^{-x^2/2}\,dx = 2\int_0^\infty \Phi^j(ay)\,y\,\mathrm{e}^{-y^2}\,dy,$$

where $y = x/\sqrt{2}$, and, in particular,

$$\begin{aligned}
\mathbf{J}_1(a) &= \tfrac{1}{2} - \tfrac{1}{2}\mathbf{I}_1(a),\\
\mathbf{J}_2(a) &= \tfrac{1}{2} - \mathbf{I}_1(a) + \tfrac{1}{2}\mathbf{I}_2(a),\\
\mathbf{J}_3(a) &= \tfrac{1}{2} - \tfrac{3}{2}\mathbf{I}_1(a) + \tfrac{3}{2}\mathbf{I}_2(a) - \tfrac{1}{2}\mathbf{I}_3(a),\\
\mathbf{J}_4(a) &= \tfrac{1}{2} - 2\mathbf{I}_1(a) + 3\mathbf{I}_2(a) - 2\mathbf{I}_3(a) + \tfrac{1}{2}\mathbf{I}_4(a).
\end{aligned}$$

We can calculate the terms $\mathbf{I}_j(a)$ by applying integration by parts. Without going into too many details, the result is

$$\begin{aligned}
\mathbf{I}_1(a) &= \frac{a}{\sqrt{1+a^2}},\\
\mathbf{I}_2(a) &= 4\frac{a}{\sqrt{1+a^2}}\,\mathbf{I}^{(1)}(a,0,0,\frac{(1+a^2)^{-1}}{2}),\\
\mathbf{I}_3(a) &= \frac{3a}{\sqrt{1+a^2}} - \frac{12a}{\pi\sqrt{1+a^2}}\tan^{-1}\sqrt{\frac{1+a^2}{1+3a^2}},\\
\mathbf{I}_4(a) &= \frac{8a}{\sqrt{1+a^2}}\,\mathbf{I}^{(3)}(a,0,0,\frac{(1+a^2)^{-1}}{2}),
\end{aligned}$$

where $\mathbf{I}^{(j)}(a,b,m,s)$ is the integral defined in (4). The $\mathbf{I}_3(a)$ case involves an integral of the product of $\exp(-a^2y^2)$ and two error functions of different arguments contained in [33]. In general, it is easy to show that for $n \geq 1$:

$$\mathbf{I}_n(a) = \frac{2na}{\sqrt{1+a^2}}\,\mathbf{I}^{(n-1)}(a,0,0,\frac{(1+a^2)^{-1}}{2}).$$

We showed in the previous section how to develop an analytical approximation to this integral using the EQA for $\mathrm{erf}(x)$ in (26) when $n = 1$. When $n > 1$ a multinomial expansion is required to compute the n-th power of the 4-term EQA for $\mathrm{erf}^n(x)$.

6 Conclusion & Further Work

We set out to find an approximation for Q or error functions that also provides a good fit for small arguments. At the same time we sought functional forms of the approximator that lead to analytical closed forms for a set of key integrals $\mathbf{I}^{(n)}(a,b,m,s)$. Focussing on the error function, we were led to consider approximators based on exponentials of quadratic functions satisfying interpolation and smoothness constraints at the lower and upper limits of the approximation interval. We derived a 4-term exponential-quadratic approximator with 12 parameters satisfying two interpolation and two first derivative constraints. We showed how to transform this 12-parameter constrained optimisation problem into an unconstrained 8-parameter problem. The resulting least-squares cost function is complicated and requires symbolic algebra to obtain. We formulated a gradient descent optimisation algorithm to estimate the unknown parameters. This step also requires symbolic algebra to obtain the derivatives of the cost function.

Following gradient descent optimisation, we obtained a 4-term EQA for the error function having a maximum absolute error of $1.65E{-}4$ on the real line. Applying the 4-term EQA, we obtained an analytical approximation to the key integral in the $n = 1$ case. We compared the analytical approximation $\widehat{\mathbf{I}}(a,b,m,s)$ in various cases to Monte Carlo integration and obtained agreement to $2.2E{-}4$. Further examples were given for error probability calculations on fading communication channels that are reducible to sums of terms involving the set of key integrals. The use of a 4-term EQA results in a massive computational compared with MC integration, while offering 3 - 4 significant figure accuracy.

The 4-term EQA we obtained is not optimal in any specific sense: the progress of the gradient descent algorithm depends strongly on the choice of knots and the initial parameter values. A possible direction for further research is to vary the initialisation settings to see if better results can be obtained. There is also the question of convergence and stability of the gradient descent algorithm, since some optimisation settings lead to positive leading exponents ($a_i < 0$). An obvious way to obtain more accurate approximations is to move to an EQA with $M > 4$ terms. Manual optimisation may be inferior to well designed search strategies for this type of approximation problem [16, 27] and a number of avenues for extension are conceivable, including the application of "Bayesian optimisation," which applies Gaussian process regression to construct a model of the objective function's dependence on the hyperparameters.

Regardless of whether the 4-term EQA can be significantly improved, the proposed approximation can be applied to obtain efficient and accurate evaluations of definite integrals involving expectations of Q or error functions with respect to random variables that involve quadratic exponentials in their PDFs. The task of applying the 4-term EQA to powers of the error function involves a multinomial expansion in products of exponentials in quadratic arguments. Although the theory remains within the framework established in this paper, the investigation of this area is left for a future study.

Appendix

Matlab Symbolic Toolbox commands for generating the derivatives code for the 4-term, 4-constraint gradient descent algorithm.

```
syms a1 a2 a3 a4 b1 b2 b3 b4 c1 c2 c3 c4
syms d1 d2 d3 d4 e1 e2 e3 e4 q r s x
S=solve(c1+c2+c3+c4-1,...
   d1*c1+d2*c2+d3*c3+d4*c4-q,...
   b1*c1+b2*c2+b3*c3+b4*c4-s,...
   e1*c1+e2*c2+e3*c3+e4*c4-r,c1,c2,c3,c4);
p4=1-S.c1*exp(-a1*x^2+2*b1*x)...
  -S.c2*exp(-a2*x^2+2*b2*x)...
  -S.c3*exp(-a3*x^2+2*b3*x)...
  -S.c4*exp(-a4*x^2+2*b4*x)
phi4=subs(p4,{d1,d2,d3,d4,e1,e2,e3,e4},
  {exp(-a1+2*b1),exp(-a2+2*b2),...
   exp(-a3+2*b3),exp(-a4+2*b4),...
  (a1-b1)*exp(-a1+2*b1),...
  (a2-b2)*exp(-a2+2*b2),...
  (a3-b3)*exp(-a3+2*b3),...
  (a4-b4)*exp(-a4+2*b4)})
dphi4da1=diff(phi4,a1);
dphi4da2=diff(phi4,a2);
dphi4da3=diff(phi4,a3);
dphi4da4=diff(phi4,a4);
dphi4db1=diff(phi4,b1);
dphi4db2=diff(phi4,b2);
dphi4db3=diff(phi4,b3);
dphi4db4=diff(phi4,b4);
ccode(S.c1); ccode(S.c2); ccode(S.c3);
ccode(S.c4); ccode(phi4); ccode(dphi4da1);
ccode(dphi4da2); ccode(dphi4da3);
ccode(dphi4da4); ccode(dphi4db1);
ccode(dphi4db2); ccode(dphi4db3);
ccode(dphi4db4);
```

References

[1] C. Hastings, Jr., *Approximations for Digital Computers*, Princeton, NJ: Princeton University Press, pp. 167–169, 1955.

[2] C. W. Clenshaw, *Chebyshev series for mathematical functions*, National Physical Laboratory Mathematical Tables, vol. 5, Her Majesty's Stationery Office, London, 1962.

[3] W. J. Cody, "Rational Chebyshev approximations for the error function", *Mathematics of Computation*, Vol. 23, No. 107, pp. 631–637, July 1969.

[4] J. L. Schonfelder, "Chebyshev expansions for the error and related functions", *Mathematics of Computation*. vol. 32, no. 144, pp. 1232–1240, 1978.

[5] S. Chevillard, "The functions erf and erfc computed with arbitrary precision and explicit error bounds," *Information and Computation*, vol. 216, pp. 72–95, 2012.

[6] M. K. Simon, "Some new twists to problems involving the Gaussian probability integral," *IEEE Transactions on Communications*, vol. 46, no. 2, pp. 200–210, 1998.

[7] M. K. Simon, "Single integral representations of certain integer powers of the Gaussian Q-function and their application", *IEEE Communications Letters*, vol. 6, no. 12, pp. 532–534, 2002.

[8] ——, "A simpler form of the Craig representation for the two-dimensional joint Gaussian Q-function", *IEEE Communications Letters*, vol. 6, no. 2, pp. 49–51, 2002.

[9] R. M. Radaydeh and M. M. Matalgah, "Results for integrals involving m-th power of the Gaussian Q-function over Rayleigh fading channels with applications," *Proceedings of the International Conference on Communications*, pp. 5910–5914, 2007.

[10] G. W. Pulford and K. Kondrashov, "Convergence and Optimality Analysis of Low-Dimensional Generative Adversarial Networks using Error Function Integrals", *IEEE Access*, vol. 9, pp. 165366–165384, Dec. 2021.

[11] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*, 7th ed. Academic Press, 2007.

[12] H. A. Fayed and A. F. Atiya, "An evaluation of the integral of the product of the error function and the normal probability density with application to the bivariate normal integral", *Mathematics of Computation*, vol. 83, no. 285, pp. 235–250, 2013.

[13] C.R. Selvakumar, "Approximations to complementary error function by method of least squares", *Proceedings of the IEEE*, vol. 70, no. 4, pp. , 1982.

[14] S. A. Dyer and J. S. Dyer, "Approximations to Error Functions", *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 6, pp. 45–48, 2007.

[15] S. Aggarwal, "A Survey-cum-Tutorial on Approximations to Gaussian Q Function for Symbol Error Probability Analysis Over Nakagami-m Fading Channels", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2195–2223, 2019.

[16] V. N. Q. Bao, L. P. Tuyen, and H. H. Tue, "A Survey on Approximations of One-Dimensional Gaussian Q-Function," *REV Journal on Electronics and Communications*, vol. 5, no. 1-2, pp. 1–14, 2015.

[17] I. M. Tanash and T. Riihonen, "Global minimax approximations and bounds for the Gaussian Q-function by sums of exponentials", *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6514–6524, 2020.

[18] M. Abramowitz and I. A. Stegun, (eds.) *Handbook of Mathematical Functions*, National Bureau of Standards, Washington, D.C., 1964.

[19] P. Borjesson and C. E. Sundberg, "Simple Approximations of the Error Function Q(x) for Communications Applications", *IEEE Transactions on Communications*, vol. 27, no. 3, pp. 639–643, 1979.

[20] W. M. Jang, "A simple upper bound of the Gaussian Q-function with closed-form error bound", *IEEE Communications Letters*, vol. 14, no. 2, p. 157, Feb. 2011.

[21] G. K. Karagiannidis and A. S. Lioumpas, "An improved approximation for the Gaussian Q-function", *IEEE Communications Letters*, vol. 11, no. 8, pp.644–646, Aug. 2007.

[22] P. Loskot and N. C. Beaulieu, "Prony and polynomial approximations for evaluation of the average probability of error over slow-fading channels", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1269–1280, Mar. 2009.

[23] L. M. Benitez and F. Casadevall, "Versatile, accurate, and analytically tractable approximation for the Gaussian Q function", *IEEE Transactions on Communications*, vol. 59, no. 4, pp. 917–922, Apr. 2011.

[24] M. Chiani, D. Dardari, and M. K. Simon, ""New exponential bounds and approximations for the computation of error probability in fading channels"," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 840–845, 2003.

[25] P. C. Sofotasios and S. Freear, "Novel expressions for the Marcum and one dimensional Q-functions", *Proceedings*

of the International Symposium on Wireless Communication Systems, pp. 736–740, 2010.
[26] A. Annamalai, E. Adebola and O. Olabiyi, "Simple Closed-Form Approximations for the ASER of Digital Modulations over Fading Channels", *Proceedings of the International Conference on Wireless Networks*, pp. 1–7, 2012.
[27] P. Dao Ngoc, U. Nguyen Quang, H. Nguyen Xuan, and R. McKay, "Evolving approximations for the Gaussian Q function by genetic programming with semantic based crossover", *Congress on Evolutionary Computation*, pp. 1–6, 2012.
[28] P. Van Halen, "Accurate analytical approximations for error function and its integral", *Electronics Letters*, vol. 25, no. 9, pp. 561–563, 1989.
[29] R. M. Howard, "Arbitrarily Accurate Analytical Approximations for the Error Function", Technical Report, School of Elec. Eng. Comp. & Math. Sci., Curtin University, Perth, Australia, pp.1–44, Dec. 2020.
[30] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
[31] E. W. Ng and M. Geller, "A table of integrals of the error functions," *J. Research of the National Bureau of Standards*, vol. 73B, no. 1, pp. 1–20, 1968.
[32] A. P. Prudnikov, Y. A. Brychkov, and O. I. Marichev, *Integrals and Series, Volume 2 - Special Functions*. New York: Gordon & Breach Science Publishers, 1986.
[33] N. E. Korotkov and A. N. Korotkov, *Integrals Related to the Error Function*, 1st ed. Chapman and Hall (CRC Press), 2020.

Graham W. Pulford (IEEE Senior Member 2005) is a graduate of the University of NSW (BSc.(hons), B.E. 1988) and Australian National University (Ph.D. 1992, telecommunications). He worked in biological signal processing (1992); then on sensor fusion and radar tracking algorithm development at the University of Melbourne (1993-2000). He worked with Thales Underwater Systems (2001-2003 and 2010-2015) in Australia and France on passive and active sonar systems and QinetiQ UK (2004-2009) on INS-GNSS navigation systems. From 2015-2019 he worked on machine learning algorithm design, including for medical devices. Since the start of the Covid-19 pandemic in 2020, he has been a freelance research engineer. In 2022, he developed algorithms for multisensor passive RF tracking with LiveLink Aerospace (UK). His research interests are in design and performance analysis of target tracking and machine learning algorithms.