Spline approximations of the Lambert $W$ function and application to simulate generalized Gaussian noise with exponent $\alpha = 1/2$

Abdelilah Monir $^a$,*, Hamid Mraoui $^b$

$^a$ Faculty of Science (Lamda), University Mohammed First, Oujda, Morocco
$^b$ Faculty of Science, University Mohammed First, Oujda, Morocco

**Abstract**

In this paper we focus on the problem of synthesizing a generalized Gaussian noise with exponent $1/2$ using the inverse transform of the cumulative distribution function (c.d.f.). It has been previously shown that this inverse can be written in terms of the Lambert $W$ function. We have developed a new procedure to evaluate this special function using spline functions. Based on this method, we present techniques to generate generalized Gaussian noise with exponent $1/2$. It is observed that the method developed in this paper is better than the approach of Chapeau-Blondeau and Monir.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

The probability density of random signals derived from complex systems are often non-Gaussian, in the majority of cases with a heavy-tail. This density characterizes signals that may take, with high probability, values far from mean value. In the field of science and engineering, many modern processes, including signals, images and communication systems, frequently have to operate in complex environments dominated by non-Gaussian noises [13,1]. Efficient design, control, and performance evaluation in these contexts depend on the capability of modeling and synthesizing such non-Gaussian noises. A random variable $X$ described by the cumulative distribution function $F(x)$ can be synthesized by a non-linear memoryless transformation applied to uniform noise $U$ as $X = F^{-1}(U)$.

In this work, we used a class of heavy-tailed distribution called the Generalized Gaussian Distribution (GGD) with a shape parameter $\alpha$. For $\alpha = 1/2$, it was proved that the corresponding inverse $F^{-1}(x)$ is expressible by means of a special function called Lambert $W$ function. Chapeau-Blondeau and Monir [7] developed an algorithm to evaluate numerically $W_{-1}(x)$. They solved the defining equation, $W(x)e^{W(x)} = x$, by Halley’s third-order method. Their initial value was computed by either (1) the series expansion around the branch point, $x = -1/e$, (2) the asymptotic approximation when $x \to 0^+$, or (3) two sets of Padé approximant of order $[4/3]$. The relative error of the initial guess was less than $10^{-4}$. Our aim, by using the spline function, is to develop an accurate numerical method that obtains the double-precision solution.

Many applications of splines make use of some approximation method to produce a spline function from discrete data. The advantages of B-splines for signal representation are well known [21]. A spline approximation method consists of two main steps: First the degree and knot vector are determined, and then the B-spline coefficients of the approximation are computed from given data according to some formula. Popular methods include interpolation and least squares approximation. However, both of these methods require solution of a linear system of equations with as many unknowns as the dimension of the spline space, and are therefore not suitable for real-time processing of large streams of data. For this purpose local methods, which determine spline coefficients by using only local information, are more suitable. In other cases, like Hermite interpolation and Schoenberg’s Variation Diminishing spline approximation, the formula for the coefficients is given directly in terms of given values of the function to be interpolated. In practice most of the data derived from experiments are typically noisy, it is therefore better to approximate them by using quasi-interpolants (abbr. QIs).

QIs have been proposed as tools for fitting data obtained from scientific and engineering applications. A QI for a given function $f$ is usually obtained as a linear combination of the elements of a suitable set of functions which are required to be positive, to ensure stability and to have small local supports in order to achieve local control. The coefficients of the linear combinations are the values of linear functionals depending on $f$ and its derivatives or integrals. To ensure good approximation properties it is important...
that the quasi-interpolation methods reproduce polynomials and preferably the functions in the given spline space.

In this paper we develop an accurate approximation to compute the real-valued Lambert function \( W_{-1}(x) \) using a quasi-interpolation method. The method is a composite of (1) the asymptotic approximation when \( x \to 0^- \), and (2) the numerical solution of the equation \( xe^x = x \) by Halley’s third-order method. The initial guess is determined as the first zero of the control polygon of a QI which approximate \( f(z) = xe^z \), or one shot approximation based on quasi-interpolation.

The paper is organized as follows. In Section 2, we recall some facts about splines and QIs. Section 3 presents some properties of the generalized Gaussian density. In Section 4 simulation techniques for a generalized Gaussian random variable are analyzed using spline quasi-interpolation.

2. Background spline material

2.1. Univariate B-splines representation

In this subsection, we review some basic properties of B-splines, for more details we refer to [5,17].

For \( I := [a,b] \), we denote by \( \mathcal{S}_{d}(I, \mathbf{t}) \) the space of splines of degree \( d \) (order \( d+1 \)) and of class \( C^{d-1} \) on the partition \( \mathbf{t} := \{ a = t_1 < \ldots < t_{d+1} < \ldots < t_n < t_{n+1} = \ldots = t_{n+d-1} = b \} \).

A B-spline basis of this space is \( \{ B_{i,d}(x) \}_{i=1}^{n} \). Each B-spline \( B_{i,d}(x) \) is finitely supported on \( [t_i, t_{i+d+1}] \) and is positive in the interior of its support. Furthermore, the \( \{ B_{i,d}(x) \}_{i=1}^{n} \) form a partition of unity, i.e., \( \sum_{i=1}^{n} B_{i,d}(x) = 1 \).

Now, we recall the following Marsden identity. For all polynomial \( p \) of degree \( d \), we have

\[
p(x) = \sum_{i=1}^{n} B[p](t_{i+1}, \ldots, t_{i+d}) B_{i,d}(x), \quad \forall x \in I,
\]

where \( B[p] \) is the blossom (polar form) of \( p \), i.e., \( B[p] \) is the unique symmetric multi-linear function such that \( B[p](x, \ldots, x) = p(x) \) for all \( x \in I \), see [20]. If we consider the polynomial \( p(x) = x^k \), then

\[
B[\mathcal{I}^k](x_1, \ldots, x_d) = \left( \sum_{1 \leq i_1 < \ldots < i_k \leq d} x_{i_1} x_{i_2} \ldots x_{i_k} \right) \left( \begin{array}{c} d \\ k \end{array} \right),
\]

where \( \mathcal{I}^k \) is the multinomial basis of \( x \).

In particular, the Grevelle points

\[
t_{r,d}^i := B[i](t_{i+1}, \ldots, t_{i+d}) = (t_{i+1} + \ldots + t_{i+d})/d,
\]

are the B-spline coefficients of \( x = \sum_{i=1}^{n} t_{r,d}^i B_{i,d}(x) \).

Let \( s \) be a spline in \( \mathcal{S}_{d}(I, \mathbf{t}) \) with coefficients \( (c_i) \), i.e.,

\[
s(x) = \sum_{i=1}^{n} c_i B_{i,d}(x), \quad \forall x \in I.
\]

The control polygon \( \Gamma_{d}(s) \) of \( s \) is defined as a piecewise linear function with vertices at \( (t_{i,d}, c_i) \). Furthermore, we know from classical spline theory [17,23] that the control polygon converges quadratically to the spline as the maximal distance between two neighboring knots tends to zero. More precisely, we have

\[
\| s - \Gamma_{d}(s) \|_{\infty} \leq K_{1} h^{2} \| s^{(2)} \|_{\infty}, \quad (2.1)
\]

where \( h = \max_{i}(t_{i+1} - t_i) \) and the constant \( K_1 \) only depends on \( d \).

The derivative of \( s \) is a spline in \( \mathcal{S}_{d-1}(I, \mathbf{t}) \) which can be written in the form

\[
s'(x) = \sum_{i=1}^{n+1} \Delta c_i B_{i,d-1}(x), \quad \forall x \in I,
\]

where

\[
\Delta c_i = \frac{c_{i+1} - c_{i}}{t_{i+d} - t_i} = \frac{c_{i-1} - c_{i}}{t_{i+1} - t_{i-d}},
\]

if we use the conventions that \( c_0 = c_{n+1} = 0 \) and \( \Delta c_i = 0 \) whenever \( t_{i,d} - t_{i-1,d} = 0 \).

De Boor’s algorithm [5] is a generalization of de Casteljau’s algorithm [6]. It provides a fast and numerically stable way for finding a point on a B-spline curve given a \( z \) in the domain. Let \( z \in [t_{d+1,mu+1}, t_{mu+1}] \), then from [5] we have

\[
\begin{align*}
\bullet & \quad c_i^d := c_i, \quad (\mu - d \leq i \leq \mu) \\
\bullet & \quad \text{For } r = 0, \ldots, d - 1 \\
\bullet & \quad \text{Let } t_{r,d}^i := t_{r,i+1}.
\end{align*}
\]

2.2. Univariate splines discrete quasi-interpolants

We denote by \( \mathcal{P}_d \) the space of polynomials of total degree at most \( d \). For a given function \( f \), univariate spline discrete QIs can be defined as operators of the form

\[
\tilde{Q}_d f = \sum_{i=1}^{n} \lambda_i(f) B_{i,d}(x), \quad (2.2)
\]

where \( \lambda_i(f) \) is a linear combination of discrete values of \( f \) at some points in the neighborhood of \( \text{supp}(B_{i,d}) \). More precisely, the coefficients functionals \( \lambda_i(f) \) are all in the form

\[
\lambda_i(f) = \sum_{j=0}^{d} w_{i,j} f(x_{i,j}),
\]

for suitable numbers \( (w_{i,j})_{j=0}^{d} \) and \( (x_{i,j})_{j=0}^{d} \).

In general we impose that \( \tilde{Q}_d \) is exact on the space \( \mathcal{P}_d \), i.e.,

\[
\tilde{Q}_d p = p, \quad \text{for all } p \in \mathcal{P}_d.
\]

Some authors impose furthermore that \( \tilde{Q}_d \) is a projector on the space of splines itself (see [16]), i.e.,

\[
\tilde{Q}_d s = s, \quad \text{for all } s \in \mathcal{S}_{d}(I, \mathbf{t}).
\]

As a consequence of this property, the approximation order is \( O(h^{d+1}) \) if \( f \) is a smooth function.

Now we give some QIs based on points evaluators. Let

\[
t_i \leq x_0 < x_1 < \ldots < x_i \leq x_{i+d+1}
\]

lie in the support \( [t_i, t_{i+d+1}] \) of the B-spline \( B_{i,d}(x) \) for \( 1 \leq i \leq n \). Then there exists a Lagrange basis

\[
p_{i,j}(x) = \prod_{r=0, r \neq j}^{d} \frac{x - x_r}{x_i - x_r}, \quad j = 0, \ldots, d,
\]

so that the polynomial

\[
p_i(x) = \sum_{j=0}^{d} f(x_{i,j}) p_{i,j}(x), \quad x \in I,
\]
satisfies the interpolation conditions \( p_i(x_{i,j}) = f(x_{i,j}) \) for \( 0 \leq j \leq d \).

Consider the QI
\[
Q_d f = \sum_{i=1}^{n} B_i[p_i](t_{i+1}, \ldots, t_{i+d}) B_{i,d,t}.
\]

Then, \( Q_d \) reproduces all polynomials of degree \( d \). By this approach, we can compute the coefficient functionals \( \lambda_i(f) := B_i[p_i](t_1, \ldots, t_{i+d}) \). Indeed, we have
\[
\lambda_i(f) = B_i[p_i](t_{i+1}, \ldots, t_{i+d}) = \sum_{j=0}^{d} f(x_{i,j}) B_i[p_i,j](t_{i+1}, \ldots, t_{i+d}).
\]

We now give an example for \( d = 3 \) and sample points \( x_{i,j} \):
\[
\lambda_1(f) = f(a),
\]
\[
\lambda_2(f) = \frac{2h_2(2\gamma_2 + h_1(\gamma_2 + 2\gamma_1))}{3\gamma_1}\ f(t_4) + \frac{\gamma_1\sigma_1}{3h_2\gamma_2}\ f(t_5)
- \frac{h_1^2\sigma_1}{3h_2h_3\gamma_1} f(t_6) + \frac{h_2^2\gamma_1}{3h_2\gamma_2\gamma_1} f(t_7),
\]
\[
\lambda_i(f) = \xi_i f(t_{i+1}) + \eta_i f(t_{i+2}) + \theta_i f(t_{i+3}), \quad 3 \leq i \leq n - 2,
\]
\[
\lambda_{n-1}(f) = \frac{2h_{n-1}\gamma_{n-2} + h_{n}(h_{n-1} + \gamma_{n-2})}{3\gamma_{n-1}\sigma_{n-2}} f(t_{n-2})
+ \frac{\gamma_{n-1}\sigma_{n-2}}{3h_{n-1}\gamma_{n-2}} f(t_{n-1}) - \frac{h_2^2\sigma_{n-2}}{3h_{n-1}h_{n-2}\gamma_{n-1}} f(t_n)
+ \frac{h_1^2\gamma_{n-1}}{3h_{n-2}\gamma_{n-2}\sigma_{n-2}} f(t_{n+1}),
\]
\[
\lambda_n(f) = f(b),
\]
where
\[
\sigma_1 = h_1 + h_{i+1} + h_{i+3}, \quad \gamma_1 = h_i + h_{i+1},
\]
and
\[
\zeta_i = -\frac{h_2^2}{3h_{i+2}(h_{i+2} + h_{i-1})}, \quad \eta_i = \frac{(h_{i-2} + h_{i-1})^2}{3h_{i-2}h_{i-1}}.
\]
\[
\theta_i = 1 - \zeta_i - \eta_i.
\]

Then, it is easy to verify that the QI of the form
\[
Q_3 f = \sum_{i=1}^{n} \lambda_i(f) B_{i,3,t}
\]
is exact on \( P_3 \). Hence, we have
\[
\|f - Q_3 f\|_\infty = O(h^4).
\]

3. Generalized Gaussian noise

3.1. Lambert W function

The Lambert W function dates back to two mathematicians, Johann Lambert (1728–1777) and Leonhard Euler (1707–1783) [11]. This function has progressively been recognized in many various fields of pure and applied mathematics [10], geophysics [4], physics [9], and engineering [7,19].

The Lambert W function is defined to be the inverse of the function \( w = \exp(w) = x \) [11]. This function is multivalued displaying two real branches. If \( x \) is real in the interval \(-1/e \leq x < 0\), there are two real values for \( W(x) \). The branch that satisfies \(-1 \leq W(x) \) is defined as the principal branch of the W function and is denoted by \( W_0(x) \). The branch that satisfies \( W(x) \leq -1 \) is denoted by \( W_{-1}(x) \). If \( x \) is real and \( x \geq 0 \), there is a single real value for \( W(x) \), which also belongs to the principal branch \( W_0(x) \). It is the real branch \( W_{-1}(x) \) for \(-1/e \leq x < 0\) that will be useful to us presently. Both real branches \( W_{-1}(x) \) and \( W_0(x) \), for \( x \) real, are represented in Fig. 1.

3.2. Generalized Gaussian distribution

The probability density function of generalized Gaussian noise is defined as
\[
f(x) = \frac{b\alpha}{2\Gamma(1/\alpha)} \exp\left(-|x|^\alpha\right),
\]
with parameter \( b \) written in terms of the complete Gamma function: \( b = [\Gamma(3/\alpha)/\Gamma(1/\alpha)]^{1/3} \), and an exponent \( \alpha > 0 \) (shape parameter).

Eq. (3.1) characterizes a zero-mean unit-variance random quantity \( X \); the probability density of any variable \( Y \) with mean \( \mu \) and variance \( \sigma^2 \) can be obtained using the following transformation \( Y = \sigma X + \mu \).

For the exponent \( \alpha = 2 \) one recovers the standard Gaussian density. For \( \alpha > 2 \) one gets densities that go to zero more rapidly than the Gaussian density, the limit \( \alpha = +\infty \) yielding uniform density. For \( 0 < \alpha < 2 \) one gets densities that go to zero less rapidly than the Gaussian density. They belong to the class of leptokurtic, or heavy-tailed, densities, which are relevant to a broad variety of processes, including modern technologies like communications, transportation and finance [13,1]. In contrast to other heavy-tailed densities such as stable densities [22], leptokurtic generalized Gaussian densities keep the interesting property of having all their moments finite, making them suitable for physical modeling of many processes with heavy tails. Generalized Gaussian noises have recently attracted attention as offering good models for various processes of current interest such as speech, audio or video signals [24,8,2], images [3,18,25,14], and turbulence [12]. Efficient simulation of generalized Gaussian noise, to which the present study seeks to contribute, can thus benefit from a better understanding and control of such processes.

One common instance of a leptokurtic generalized Gaussian density is obtained when \( \alpha = 1 \) which yields the Laplace noise, with probability density
\[
f(x) = \frac{1}{\sqrt{2}} \exp(-\sqrt{2}|x|)
\]
and cumulative distribution function
\[
F(x) = \begin{cases} 
\frac{1}{2} \exp(-\sqrt{2}|x|) & \text{for } x \leq 0, \\
1 - \frac{1}{2} \exp(-\sqrt{2}|x|) & \text{for } x \geq 0.
\end{cases}
\]
The inverse function
\[
F^{-1}(x) = \begin{cases} \frac{1}{2}\sqrt{\frac{x}{30}} \ln(2x) & \text{for } 0 < x \leq 1/2, \\ -\frac{1}{2}\sqrt{\frac{x}{30}} \ln[2(1-x)] & \text{for } 1/2 \leq x < 1. \end{cases} \quad (3.4)
\]
then solves the problem of synthesizing a Laplace random noise \( X \) from a uniform noise \( U \) as exposed in the introduction.

We now consider the case \( \alpha = 1/2 \) associated to the probability density \([7]\)
\[
f(x) = \frac{\sqrt[3]{30}}{2} \exp(-|2\sqrt{30}x|^{1/2}) \quad (3.5)
\]
and cumulative distribution function
\[
F(x) = \begin{cases} \frac{1}{2}(1 + |2\sqrt{30}x|^{1/2}) \exp(-|2\sqrt{30}x|^{1/2}) & \text{for } x \leq 0, \\ 1 - \frac{1}{2}(1 + |2\sqrt{30}x|^{1/2}) \exp(-|2\sqrt{30}x|^{1/2}) & \text{for } x \geq 0. \end{cases} \quad (3.6)
\]
Numerical results highlight that the technique based on the inverse cumulative distribution function written in terms of the Lambert \( W \) function is the most efficient. It is well known that this inverse can be expressed in terms of the Lambert function \( W_{-1} \) (see \([7]\)), i.e.,
\[
F^{-1}(x) = \begin{cases} \frac{1}{2\sqrt{30}} \left[ 1 + W_{-1}(-2x/e) \right]^2 & \text{for } 0 < x \leq 1/2, \\ \frac{1}{2\sqrt{30}} \left[ 1 + W_{-1}(-2(1-x)/e) \right]^2 & \text{for } 1/2 \leq x < 1. \end{cases} \quad (3.7)
\]
Eq. (3.7) solves the problem of obtaining a large number of realizations of generalized Gaussian density with \( \alpha = 1/2 \), provided we are able to compute the function \( W_{-1} \). Based on such results, an accurate simulation procedure can be defined: generate a random number uniform on \([0,1]\), then approximate the function \( W_{-1} \), and finally calculate \( F^{-1} \).

4. Numerical evaluation of \( W_{-1} \)

The Lambert function \( W_{-1} \) is defined on \([-1/e, 0]\). As \( x \to 0^+ \), \( W_{-1} \) diverges logarithmically. Hence, we are going to divide this interval in two sub-intervals \([-1/e, \epsilon]\) and \([\epsilon, 0]\), where \( \epsilon \) is a real negative close to zero.

4.1. Asymptotic expansions for \( W_{-1}(z) \) as \( z \to 0^- \)

For \( z \) approaching \( 0^- \), the Lagrange inversion theorem provides an asymptotic series expansion, with \( L_1 = \ln(-z) \) and \( L_2 = \ln(-\ln(-z)) \), which reads \([11]\)
\[
W_{-1}(z) = L_1 - L_2 + \sum_{\ell=0}^{+\infty} \sum_{m=1}^{+\infty} C_{\ell m} L_1^{-\ell} L_2^m. \quad (4.1)
\]
The coefficients \( C_{\ell m} \) are expressible as \( C_{\ell m} = (-1)^{\ell} S(\ell + m, \ell + 1)/m! \), where \( S(\ell + m, \ell + 1) \) is a nonnegative Stirling number of the first kind \([15]\), computable via the generating function
\[
x(x-1) \ldots (x-n+1) = \sum_{m=0}^{n} (-1)^{n-m} S(n, m) x^m, \quad n \geq 1, \quad (4.2)
\]
and \( S(n, m) = 0 \) for \( m > n \). It follows from Eq. (4.2) that \( \forall n \geq 1 \) one has \( S(n, 0) = 0 \) and \( S(n, n) = 1 \). Also available is the recursion
\[
S(n, m) = S(n-1, m-1) + (n-1) S(n-1, m), \quad n > 1. \quad (4.3)
\]

Table 1

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-10^{-4})</td>
<td>(6.2343 \times 10^{-10})</td>
</tr>
<tr>
<td>(-10^{-5})</td>
<td>(2.3626 \times 10^{-11})</td>
</tr>
<tr>
<td>(-10^{-6})</td>
<td>(9.4910 \times 10^{-13})</td>
</tr>
<tr>
<td>(-10^{-7})</td>
<td>(4.0139 \times 10^{-14})</td>
</tr>
<tr>
<td>(-10^{-8})</td>
<td>(1.7485 \times 10^{-15})</td>
</tr>
<tr>
<td>(-10^{-9})</td>
<td>(7.6060 \times 10^{-17})</td>
</tr>
</tbody>
</table>

The first terms of the series of Eq. (4.1) are
\[
\tilde{W}_{-1}(z) = L_1 - L_2 + \frac{(-2 + L_2) L_2}{2 L_1^2} + \frac{(6 - 9 L_2 + 2 L_2^2) L_2}{6 L_1^3} + \frac{(-12 + 36 L_2 - 22 L_2^2 + 31 L_2^3) L_2}{12 L_1^4} + \frac{(60 - 300 L_2 + 350 L_2^2 - 125 L_2^3 + 12 L_2^4) L_2}{60 L_1^5}, \quad (4.4)
\]
where \( \tilde{W}_{-1} \) denotes approximation of \( W_{-1} \), and the asymptotic error is assessed to \( \| \tilde{W}_{-1} - W_{-1} \|_\infty = O((L_2/L_1)^2) \) \([7,22]\).

In Table 1, we give the maximum absolute errors computed at various points of interval \([\epsilon, 0]\), the numerical result showed that when \( \epsilon \) approach \( 0^- \) then we have a good approximation of \( W_{-1} \).

4.2. Approximation for \( W_{-1} \) on \([-1/e, \epsilon]\)

4.2.1. Halley’s method

Let
\[
f(z, x) = f_x(z) = x \exp(x) - z, \quad z \in [-1/e, \epsilon]
\]
by means of iterative root-finding methods for a given \( z \), we can find \( W_{-1}(z) \) as the root of \( f_x(z) = 0 \).

One possibility to evaluate numerically \( f_x(x) = 0 \) is to use Halley’s method:
\[
x_{j+1} = x_j - \frac{x_j e^{x_j} - z}{(x_j + 1) e^{x_j}} - \frac{x_j e^{x_j} - z}{2(x_j + 1)(x_j e^{x_j} - z)} \quad (4.5)
\]
However, this method has some practical difficulties. If the initial value \( x_0 \) is not good enough, the method might diverge. This leads to the question of how to find a good starting value, which is tricky in itself. In order to locate the zeros of \( f_x \), we approximate \( f_x \) by its QI (see \(2.2\))
\[
Q_d f_x = \sum_{i=1}^{n} \lambda_i(f_x) B_{i,d,t}.
\]
Put \( s = Q_d f_x \). If \( Q_d \) is exact on \( \mathbb{P}_d \), then from (2.1) there exists two constants \( K_2 \) and \( C_2 \) that depend on \( z \) and \( d \), but not on \( t \), such that
\[
\| f_x - I_{d,t}(s) \|_\infty \leq K_2 h^2 \| s(1) \|_\infty + C_2 h^{d+1} \| f_x^{(d+1)} \|_\infty.
\]
Hence, a good starting point for the iterative procedure (4.5) is the first zero of the control polynomial \( I_{d,t}(s) \). The first zero of the control polynomial \( I_{d,t}(s) \) is the zero of the linear segment connecting the two points \((t_{k-1}^d, \lambda_{k-1}(f_x)) \) and \((t_k^d, \lambda_k(f_x)) \), where \( k \) is the smallest zero index, i.e., \( k \) is the smallest integer such that \( \lambda_{k-1}(f_x) \lambda_k(f_x) \leq 0 \) and \( \lambda_{k-1}(f_x) \neq 0 \). The zero is characterized by the equation
\[
(1 - \mu) \lambda_{k-1}(f_x) + \mu \lambda_k(f_x) = 0,
\]
which has the solution
Table 2
The maximum absolute errors for Halley's method when \( d = 1 \).

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>( n )</th>
<th>Error, <em>halley</em> (4.7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-10^{-4})</td>
<td>32</td>
<td>6.6613 \times 10^{-16}</td>
</tr>
<tr>
<td>(-10^{-5})</td>
<td>64</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-6})</td>
<td>64</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-7})</td>
<td>64</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-8})</td>
<td>64</td>
<td>1.6653 \times 10^{-16}</td>
</tr>
<tr>
<td>(-10^{-9})</td>
<td>128</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
</tbody>
</table>

Table 3
The maximum absolute errors for Halley's method using \( Q_{M}f_{z} \).

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>( n )</th>
<th>Error, <em>halley</em> (4.7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-10^{-4})</td>
<td>16</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-5})</td>
<td>16</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-6})</td>
<td>32</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-7})</td>
<td>32</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
<tr>
<td>(-10^{-8})</td>
<td>32</td>
<td>1.1102 \times 10^{-16}</td>
</tr>
<tr>
<td>(-10^{-9})</td>
<td>32</td>
<td>5.5511 \times 10^{-17}</td>
</tr>
</tbody>
</table>

\[ \mu = -\frac{\lambda_{k-1}(f_{z})}{\lambda_{k}(f_{z}) - \lambda_{k-1}(f_{z})}. \]

Therefore, the control polygon of \( s = Q_{M}f_{z} \) has a zero at

\[ x_{0} = t_{x} - \frac{\lambda_{k}(f_{z})}{\Delta \lambda_{k}(f_{z})}. \] (4.6)

After carrying out 3 iterations, we compute using a 100 uniform grid \( Z \) in the domain \([-1/e, \epsilon] \) the maximum absolute errors

\[ \text{Error, _halley_} := \max_{z \in Z} | f_{z}(x_{3}) |. \] (4.7)

Table 2 shows the maximum absolute errors of Halley’s method (4.5) after carrying out 3 iterations for different choices of \( n \).

To improve the rate of convergence, we start the interpolation as the zero of the control polygon of the quasi-interpolant \( Q_{M}f_{z} \). Using this starting point, the following Table 3 illustrates the maximum absolute errors of Halley’s method (4.5) after carrying out 3 iterations for different choices of \( n \).

Thus, the final approximation of \( W_{-1}(z) \) is given by

\[ W_{-1}(z) = \left\{ \begin{array}{ll} \text{Halley’s method (4.5),} & \epsilon \leq z < \epsilon; \\ \text{\[4.8\]} & \epsilon \leq z < 0. \end{array} \right. \]

**4.2.2. One-shot approximation using quasi-interpolation**

In order to get a one-shot approximation, we develop a new procedure to evaluate \( W_{-1} \) on \([-1/e, \epsilon]\) by using spline quasi-interpolation.

We start with a number \( n + 1 \) of data points \( (y_{j}, x_{j})^{n+1}_{j=1} \). We assume that the points \( y_{j} \) are given by

\[ y_{j} := y_{j} \exp(x_{j}), \quad j = 1, \ldots, n + 1, \]

where

\[ x_{n+2-j} := \tilde{W}_{-1}(\epsilon) + (j - 1) h, \quad j = 1, \ldots, n + 1, \]

with \( h = \frac{(-1 - \tilde{W}_{-1}(\epsilon))}{n} \). From \( (y_{j})^{n+1}_{j=1} \) we form the knot vector

\[ t := (t_{j})^{n+6}_{j=1} = (y_{1}, y_{1}, y_{1}, y_{2}, \ldots, y_{n}, y_{n+1}, y_{n+1}, y_{n+1}, y_{n+1}). \]

Let \( s = \sum_{i=1}^{n+1} c_{i} B_{i,3}(t) \) be a spline in \( S_{3}(l, t) \), where the coefficients are respectively

\[ c_{1} = x_{1}, \]
\[ c_{2} = \frac{2h_{2}y_{2} + h_{1}(h_{2} + y_{2})}{3\gamma_{1}\gamma_{1}}x_{4} + \frac{y_{1}\sigma_{1}}{3h_{2}y_{2}}x_{5} - \frac{h_{1}^{2}\sigma_{1}}{3h_{2}h_{3}\gamma_{1}}x_{6} \]
\[ + \frac{h_{2}^{2}\gamma_{1}}{3h_{3}\gamma_{2}\sigma_{1}}x_{7}, \]
\[ c_{i} = \frac{2h_{i-1}y_{i-2} + h_{i-1}y_{i-2} + y_{i-2}}{3h_{i-1}\gamma_{i-2}}x_{n-2} + \frac{y_{i-1}\sigma_{i-2}}{3h_{i-2}h_{i-1}\gamma_{i-2}}x_{n-1} \]
\[ - \frac{h_{i}^{2}\sigma_{i-2}}{3h_{i-1}h_{i-2}\gamma_{i-2}}x_{n} + \frac{h_{i}^{2}\gamma_{i-1}}{3h_{i-1}-h_{i-2}\gamma_{i-2}}x_{n+1}, \]

where

\[ h_{i} = y_{i} - y_{i-1}, \quad \sigma_{i} = h_{i} + h_{i+1} + h_{i+3}, \quad \gamma_{i} = h_{i} + h_{i+1}, \]

and

\[ \zeta = \frac{h_{i}^{2}}{3h_{i-2}h_{i-1}(h_{i-2} + h_{i-1})}, \quad \eta_{i} = \frac{(h_{i-2} + h_{i-1})^{2}}{3h_{i-2}h_{i-1}}, \]
\[ \theta_{i} = 1 - \zeta - \eta_{i}. \]

Since,

\[ x_{j} = W_{-1}(y_{j}), \quad j = 1, \ldots, n + 1, \]

hence, we can consider \( s \) as the approximant to \( W_{-1} \) on \([-1/e, \epsilon]\). From (2.3), we have

\[ s(z) = Q_{3}W_{-1}(z), \quad \text{for all } z \in [-1/e, \epsilon]. \]

Hence, using (2.4) we get

\[ ||s - W_{-1}||_{\infty} = ||Q_{3}W_{-1} - W_{-1}||_{\infty} = O(h^{4}), \]

where \( h = \max_{i} h_{i} \).

Consequently, we can approximate \( W_{-1} \) over \([-1/e, 0]\) as accurate as desired by piecewise approximation

\[ W_{-1}(z) = \left\{ \begin{array}{ll} s(z), & -1/e \leq z < \epsilon; \\ W_{-1}(z), & \epsilon \leq z < 0. \end{array} \right. \] (4.9)

In Table 4, we compute the maximum absolute errors of one-shot approximation using (4.9) for different choices of \( n \), with \( \epsilon = -10^{-6} \).

**4.3. Simulating a generalized Gaussian noise with exponent 1/2**

We have applied the piecewise approximations given by (4.8) and (4.9), to the approximation of \( F^{-1}(x) \), and we have computed the mean error on a \( 10^{3} \) random grid \( G \) on \([0, 1] \), i.e.,

\[ E_{\text{mean}} = \sum_{x \in G} |F^{-1}(x) - x| \]
\[ 10^{3}. \] (4.10)
where

\[
\tilde{F}^{-1}(x) = \begin{cases} 
-\frac{1}{2\sqrt{3}} [1 + \tilde{W}_{-1}(-2x/e)]^2 & \text{for } 0 < x \leq 1/2, \\
\frac{1}{2\sqrt{3}} [1 + \tilde{W}_{-1}(-2(1-x)/e)]^2 & \text{for } 1/2 \leq x < 1,
\end{cases}
\]

(4.11)

with \(\tilde{W}_{-1}(x)\) is given by (4.8) or (4.9). In Tables 5 and 6 we have the values of the tabulated mean error (4.10) using the piecewise approximations (4.8) and (4.9), for different values of \(n\), where \(n\) indicates the number of quasi-interpolation points. We remark that when \(\epsilon \to 0^+\), both approaches are very accurate. All computations were performed in double precision.

As an application, we used the formula (4.11) to simulate a generalized Gaussian noise with exponent 1/2. For illustration, a typical evolution of this noise, over an interval of time with length \(10^3\), is shown in Fig. 2, where the approximation of Lambert \(W\) function given by (4.8) is used, with the following fixed parameters \(\epsilon = -10^{-6}\) and \(n = 2^8\). Similarly, Fig. 4 shows another example of realizations of noise, when the approximation provided by (4.9)

### Table 5

<table>
<thead>
<tr>
<th>(\epsilon = -0.033)</th>
<th>(\epsilon = -10^{-6})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>(E_{\text{mean}})</td>
</tr>
<tr>
<td>(2^4)</td>
<td>2.4455 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^5)</td>
<td>2.4498 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^6)</td>
<td>2.4524 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^7)</td>
<td>2.4553 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^8)</td>
<td>2.4489 \times 10^{-7}</td>
</tr>
</tbody>
</table>

### Table 6

<table>
<thead>
<tr>
<th>(\epsilon = -0.033)</th>
<th>(\epsilon = -10^{-6})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>(E_{\text{mean}})</td>
</tr>
<tr>
<td>(2^4)</td>
<td>3.7174 \times 10^{-4}</td>
</tr>
<tr>
<td>(2^5)</td>
<td>5.4601 \times 10^{-5}</td>
</tr>
<tr>
<td>(2^6)</td>
<td>6.6929 \times 10^{-6}</td>
</tr>
<tr>
<td>(2^7)</td>
<td>7.7933 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^8)</td>
<td>2.9184 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^9)</td>
<td>2.5027 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^{10})</td>
<td>2.4613 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^{11})</td>
<td>2.4013 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^{12})</td>
<td>2.4477 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^{13})</td>
<td>2.4501 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^{14})</td>
<td>2.4496 \times 10^{-7}</td>
</tr>
<tr>
<td>(2^{15})</td>
<td>2.4483 \times 10^{-7}</td>
</tr>
</tbody>
</table>

Furthermore, we have performed an estimation of its probability density based on \(10^7\) values drawn for the generalized Gaussian noise and collected into bins of width \(\Delta x = 0.1\). Fig. 3 and Fig. 5 show the estimate of the probability density function, superimposed with the theoretical density (3.5), for both noises synthesized previously (see Fig. 2 and Fig. 4). As can be seen in Fig. 3 and Fig. 5, the probability density function is well approximated by the empirical density using methods described above.

### 4.4. Discussion

The algorithm developed by Chapeau-Blondeau and Monir in [7] evaluates the approximation of \(W_{-1}(z)\) with a fixed relative error (less than \(10^{-6}\)). In order to obtain double-precision accuracy, we first proposed the algorithm (4.8) to evaluate \(W_{-1}(z)\). More precisely, we solved the equation \(f_z(x) = 0\) by Halley’s third method. The precision of the initial guess becomes higher using the zero of the control polygon of the quasi-interpolant \(Q^2 f_z\). Then, at most 3 iterations of Halley’s method are sufficient to achieve the double-precision solution in a reasonable time frame. In order to get a one shot approximation, we proposed the algorithm (4.9) to evaluate \(W_{-1}(z)\). This algorithm is composed of two stages. First, for a given \(z \in [\epsilon, 0]\), we used asymptotic approximation given by Eq. (4.4). Then, for a given \(z \in [-1/\epsilon, \epsilon]\), we have developed an accurate approximation using a QI method. Double precision accuracy is achieved where the adjustable parameters \(\epsilon\) and \(h\) are close to zero. The complexity of this method is closely related to the complexity of de Boor algorithm. For each evaluation, the computational cost of de Boor algorithm is determined by the polynomial order, \(d\), rather than the number of knots, and this order is equal to \(O(d^2)\).

In a practical application, people can select an appropriate scheme based on their different demands.

### 5. Conclusion

In this contribution, several methods for approximating the Lambert function \(W_{-1}(z)\) were described and discussed. Through numerical examinations, we confirm that the new methods based on spline functions provide an accurate procedure to approximate this special function, which plays an important role in generating generalized Gaussian noise with exponent 1/2.

Generalized Gaussian distributions have been proposed in many applications in science and engineering. Therefore in future work,
Fig. 3. Probability density function estimated for the generalized Gaussian noise with exponent $1/2$ synthesized by using Halley’s method (4.8), superimposed to the theoretical model of (3.5) (continuous solid line).

Fig. 4. Instances of GGD with parameter $\alpha = 1/2$ and $N = 10^3$ synthesized by one shot approximation using quasi-interpolation (4.9).

Fig. 5. Probability density function estimated for the generalized Gaussian noise with exponent $1/2$ synthesized by one shot approximation using quasi-interpolation (4.9), superimposed to the theoretical model of (3.5) (continuous solid line).
it will be interesting to apply methods based on spline functions to any applications which use the computation of the inverse function. For example, synthesizing generalized Gaussian noises with other values of $\alpha$.

Acknowledgments

The authors are extremely thankful to unknown referees for their useful comments and suggestions that have helped in improving the readability of this paper.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.dsp.2014.06.013.

References


Abdelilah Monir was born in Morocco in 1974. He received the master degree in statistics from University Paris 6, Paris, France, in 1999 and Ph.D. degree from University of Angers, Angers, France, in 2003. In 2005, he was a Research Associate with the Department of Pediatric, CHU Saint-Justine, Montreal University, Canada. Since 2005, he has been with Faculty of Science University Mohammed First, Morocco, where he is currently Professor of Computer Science. His research interests include Signal Processing, Image Processing and Bio-Statistics.

Hamid Mraoui received the Ph.D. degree from the Faculty of Science University Mohammed First, Morocco, in 2007. Now he is a Professor at the Department of Mathematics and Computer Science, Faculty of Science University Mohammed First. His current research interests include Approximation Theory (B-splines), CAGD, Wavelet and Signal Processing.