Giftet Inc.
Journal of Geolocation, Geo-information, and Geo-intelligence
Volume 2022, article ID 2022071604, 14 pages
https://doi.org/10.18610/JG3.2022.071604

Research Article

Landmark Computation of the Generalized Bessel Function Distributions of the First Kind: Part 2

Redeeming the time, because the days are evil.

—Ephesians 5:16

Ilir F. Progri

Giftet Inc., 5 Euclid Ave. #3, Worcester, MA 01610, USA

ORCID: 0000-0001-5197-1278

Correspondence should be addressed to Ilir Progri; iprogri@giftet.com

Received May 13, 2022; Revised May 14, 2022-June 17, 2022, Accepted July 16, 2022; Published November 1, 2022.

Scientific Editor-in-Chief/ Senior Editor: Ilir F. Progri

The landmark computation of the generalized Bessel function distribution of the first kind (GBFD1K) is its most efficient computation for values of the variable greater than zero and less than infinity and for non-integer values of another parameter. GBFD1K is split into three segments from zero to one, from one to twenty-three to thirty-seven and from that parameter between twenty-three to thirty-seven to less than infinity. The efficient computation of GBFD1K for the first segment two segments is performed using any of the formulae already developed earlier in Progri 2016 and 2018. The landmark computation of the third segment takes advantage of a cleaver recursive expansion that leads to the computation of the Progri L function for the non-integer values of another parameter. In order to compare and contrast the landmark computation with the other efficient computation of GBFD1K a significant improvement of the latter was employed making use of the properties of the exponential, logarithmic functions, and the landmark computation of the regularized incomplete gamma function (RIGF). According to the numerical results derived for each case to validate the theoretical models presented in the paper, the landmark computation via the Progri L function is faster and just as accurate than the efficient computation of GBFD1K for non-integer values of another parameter via MATLAB *gammainc* built in function (BIF) or Giftet *pgammainc* BIF. This paper the pinnacle of the wealth of computation wisdom and knowledge that was generated from the assessment of computation of the GBIFD1K cdd via the Kampé de Fériet function during the last few years.

Index Terms—Bessel functions, modified Bessel functions, cumulative distribution function, Kampé de Fériet function, Progri L function, incomplete gamma functions, hypergeometric series, landmark computation.

1 Introduction

The discovery of the landmark computation of the generalized Bessel function cumulative distribution functions (cdf) of the first kind (GBFD1K) for integer and non-integer values of the parameter p did neither happen by accident nor did it occur immediately. From Progri (2016, [1]) until Progri (2021, [12]) there is a total of twelve journal papers that I developed in the past six years in which I have accumulated a remarkable wealth of computational wisdom, knowledge, and understanding.

What does this computational wisdom, knowledge, and understanding consist of?

In addition to what is already explained in Progri (2022, [1]), it became clear to me that the computation of the exponential functions for negative exponents and the computation of the regularized (or normalized) incomplete gamma function (RIGF) needed to be fully investigated as in Progri (2022, [14]) and Progri (2022, [15]).

Why is this so significant? It is significant because in Progri (2022, [15]) the computation of the RIGF revealed that the MATLAB *integral* built in function (BIF) is more accurate than the MATLAB *gammainc* BIF. Therefore, I decided to keep the computation of the GBFD1K via the MATLAB *integral* BIF as the truth and replace the linear approximation function with the computation of the GBFD1K via the MATLAB *gammainc* BIF or via the Giftet *pgammainc* function.

Second, the landmark computation of the GBFD1K via the Kampé de Fériet functions has a profoundly, entirely new understanding in Giftet *kamdefer* BIF. The first closed form expression (CFE) of the Kampé de Fériet functions comes from the application of the power series expansion of the exponential functions for negative exponents Progri (2022, [14]) with the smallest region of conversion $0 \leq x < 1$. The second set of CFEs of the Kampé de Fériet functions comes from the application of the Kummer's first transformation in Progri (2022, [15]). In Progri (2022, [15]) Kummer's first transformation is unique and it significantly expands the region of conversion of the variable x. As an illustration, if the non-integer parameter $p = 1.5$ then the region of convergence for the accurate computation of the Kampé de Fériet functions is expanded to $0 \leq x < 24$. For values of $24 \leq x < \infty$ another expression of Nielsen (1906, [16]) was employed for the first time in this paper.

Third, a recursive algorithm for computing the GBFD1K for large values of x proved to be very computationally efficient and more accurate than any of the previous computation of the GBDF1K cdf. Therefore, the computation of the GBFD1K cdf via Kampé de Fériet functions for large values of x is no longer needed. Because the Kampé de Fériet functions occur in pairs this journal paper provides the reduction formulae for each case for the Kampé de Fériet functions that either were not clearly explained or were lacking.

Fourth, this journal paper provides landmark improvements in the numerical, theoretical results Sect. 5 that is published in this journal paper for the first time.

1. Significant improvements in the computation of the MATLAB *integral* BIF.
2. Inclusion for the first time of the computation of the GBFD1K via MATLAB *gammainc* BIF and Giftet *pgammainc* BIF.
3. Landmark improvements of the computation of the GBFD1K via Giftet *kamdefer* (Kampé de Fériet functions) BIF.
4. The development of the Giftet *progril2* BIF designed to implement the best of the computational speed algorithms with the best of the performance accuracy algorithms.

According to the MATLAB simulation results in Sect. 4, the computation of the GBFD1K cdf via the Giftet progril2 BIF is the most efficient computation in both computational speed and accuracy.

This paper is organized as follows: in Sect. 2 the landmark computation of the GBFD1K via Kampé de Fériet functions is discussed. The landmark efficient computation of the GBFD1K for non-integer values of a parameter is presented in Sect. 3. Kampé de Fériet functions reduction formulae are derived in Sect. 4. Section 5 contains numerical, theoretical results; Conclusion is provided in Sect. 6 along with acknowledgement and a list of references. In Appendix A, a modified expression of the GBFD1K for integer values is provided that needed some clarification from Progri (2022, [13]).

2 Landmark Computation of the GBFD1K via Kampé de Fériet Functions

In this section the landmark computation of the GBFD1K is given via the Kampé de Fériet functions that relates the earlier research that I published in Progri (2016, [1]), Progri (2018, [5]) that needs further explanation, clarification, and implementation for large values of the variable x.

For the reason I produced two journal papers: one that deals with the computation of the exponential functions for negative exponents in Progri (2022, [14]) and the other that describe the computation of the RIGF in Progri (2022, [15]).

Unfortunately, in Progri (2016, [1]) and Progri (2018, [5]) I proposed the computation of the IGF not the RIGF which led to computational problems for large values of the variable x.

In comparison and contrast to Progri (2016, [1] (80)) the GBFD1K cdf that employed the IGF definition, here I have utilized the RIGF [15] instead; hence, the GBFD1K cdf is given by:

$$F_{\text{GBessel1}}(x;a,d,p)=\frac{1}{C_1(d,p)}\sum_{k=0}^{\infty}\frac{\int_0^x t^{2p+2k}e^{-td}dt}{k!\Gamma(p_2+k)2^{p+2k}}$$

$$=\frac{1}{C_1(d,p)}\sum_{k=0}^{\infty}\frac{\frac{\gamma'(2p_1+2k,dx)}{d^{2p_1+2k}}}{k!\Gamma(p_2+k)2^{p+2k}}$$

$$=\rho(d,p)\sum_{k=0}^{\infty}\frac{\gamma'(2p_1+2k,dx)}{\Gamma(2p_1+2k)}\frac{(p_1)_k\frac{1}{d^{2k}}}{k!}$$

$$=\rho(d,p)\sum_{k=0}^{\infty}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!}$$

$$\cong\rho(d,p)\sum_{k=0}^{K}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!} \quad (1)$$

Where the normalization coefficient $\rho(d,p)$ is given by

$$\rho(d,p)=\left(1-\frac{1}{d^2}\right)^{p_1} \quad (2)$$

In Progri (2016, [1] (80)) the GBFD1K cdf is given by:

$$F_{\text{GBessel1}}(x;a,d,p)=\rho(d,p)[F_{1e}(x,d,p)-F_{1o}(x,d,p)] \quad (3)$$

In order to produce (3), the region of convergence is from $0\le x<1$, from Progri (2018, [5]) we can compute the RIGF in (1) as follows

$$\gamma(2p_1+2k,dx)=\frac{\gamma'(2p_1+2k,dx)}{\Gamma(2p_1+2k)}$$

$$=\frac{\frac{(dx)^{2p_1+2k}}{2}}{\Gamma(2p_1+2k)}\left|\begin{matrix}\frac{1}{(p_1+k)}\sum_{m=0}^{\infty}\frac{(p_1+k)_m x_2^m}{(p_3+k)_m\left(\frac{1}{2}\right)_m m!}-\\ \frac{dx}{(p_2+k)}\sum_{m=0}^{\infty}\frac{(p_2+k)_m x_2^m}{(p_4+k)_m\left(\frac{3}{2}\right)_m m!}\end{matrix}\right.$$

$$=(dx)^{2p_1}\left|\begin{matrix}\frac{(dx)^{2k}\sum_{m=0}^{\infty}\frac{(p_1+k)_m x_2^m}{(p_3+k)_m\left(\frac{1}{2}\right)_m m!}}{\Gamma(2p_2+2k)}-\\ \frac{dx(dx)^{2k}\sum_{m=0}^{\infty}\frac{(p_2+k)_m x_2^m}{(p_4+k)_m\left(\frac{3}{2}\right)_m m!}}{(2p_1+2k)^{-1}\Gamma(2p_3+2k)}\end{matrix}\right.$$

$$=\sqrt{\pi}\left(\frac{dx}{2}\right)^{2p_1}\left|\begin{matrix}\frac{\left(\frac{dx}{2}\right)^{2k}\sum_{m=0}^{\infty}\frac{(p_1+k)_m x_2^m}{(p_3+k)_m\left(\frac{1}{2}\right)_m m!}}{\Gamma(p_2+k)\Gamma(p_3+k)}-\\ \frac{dx\left(\frac{dx}{2}\right)^{2k}\sum_{m=0}^{\infty}\frac{(p_2+k)_m x_2^m}{(p_4+k)_m\left(\frac{3}{2}\right)_m m!}}{(2p_1+2k)^{-1}2\Gamma(p_3+k)\Gamma(p_4+k)}\end{matrix}\right.$$

$$=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\left|\begin{matrix}\frac{\left(\frac{dx}{2}\right)^{2k}\sum_{m=0}^{\infty}\frac{(p_1+k)_m x_2^m}{(p_3+k)_m\left(\frac{1}{2}\right)_m m!}}{(p_3)_k(p_2)_k}-\\ \frac{dxp_1\left(\frac{dx}{2}\right)^{2k}\sum_{m=0}^{\infty}\frac{(p_2+k)_m x_2^m}{(p_4+k)_m\left(\frac{3}{2}\right)_m m!}}{p_2(p_4)_k(p_1)_k}\end{matrix}\right. \quad (4)$$

where the functions, $F_{1e}(x,d,p)$ and $F_{1o}(x,d,p)$ can be computed from

$$F_{1e}(x,d,p)=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\sum_{k,m=0}^{\infty}\frac{(p_1)_{k+m}}{(p_3)_{k+m}(p_2)_k\left(\frac{1}{2}\right)_m}\frac{x_1^k}{k!}\frac{x_2^m}{m!}$$

$$=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}F_{1:1;1}^{1:0;0}\begin{bmatrix}p_1:-;-;\\ p_3:p_2;\frac{1}{2}x_1,x_2\end{bmatrix}$$

$$=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}K_{1e}(x,d,p) \quad (5)$$

$$F_{1o}(x,d,p)=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\frac{dxp_1}{p_2}\sum_{k,m=0}^{\infty}\frac{(p_2)_{k+m}}{(p_4)_{k+m}(p_2)_k\left(\frac{3}{2}\right)_m}\frac{x_1^k}{k!}\frac{x_2^m}{m!}$$

$$=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\frac{dxp_1}{p_2}F_{1:1;1}^{1:0;0}\begin{bmatrix}p_2:-;-;\\ p_4:p_2;\frac{3}{2}x_1,x_2\end{bmatrix}$$

$$=\frac{\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\frac{dxp_1}{p_2}K_{1o}(x,d,p) \quad (6)$$

the even and odd Kampé de Fériet functions, $K_{1e}(x,d,p)$ and $K_{1o}(x,d,p)$, are discussed in great detail in Progri (2018, [5], Progri (2019, [6]).

Substituting (5) and (6) into (4) and factoring out the common terms we obtain for $0\le x<1$:

$$F_{\text{GBessel1}}(x;a,d,p)=\frac{\rho(d,p)\sqrt{\pi}x_2^{p_1}\left[K_{1e}(x,d,p)-\frac{dxp_1K_{1o}(x,d,p)}{p_2}\right]}{\Gamma(p_3)\Gamma(p_2)} \quad (7)$$

Let us compare and contrast (7) with (see Progri (2018, [5]) (49)

$$F_{\text{GBessel1}}(x;a,d,p)=\frac{x^{2p_1}\left[K_{1e}(x,d,p)-\frac{dxp_1K_{1o}(x,d,p)}{p_2}\right]}{C_{11}^{-1}(d,p)} \quad (8)$$

Where

$$C_{11}(d,p)=\frac{(d^2-1)^{p_1}}{\Gamma(2p_2)}\equiv\frac{\sqrt{\pi}(d^2-1)^{p_1}}{4^{p_1}\Gamma(p_2)\Gamma(p_3)} \quad (9)$$

For (7) and (8) to be identical if and only if

$$\Gamma(2p_2)\equiv\frac{4^{p_1}}{\sqrt{\pi}}\Gamma(p_2)\Gamma(p_3) \quad (10)$$

$$\frac{\rho(d,p)\sqrt{\pi}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)} \equiv C_{11}(d,p)x^{2p_1} \quad (11)$$

Which shows that the derivation in Progri (2018, [5]) are correct.

Next, we derive the equations of the GBFD1K cdf for the values of $0 \le x \le 37$; hence, the GBFD1K cdf takes the form of

$$F_{\text{GBessel1}}(x;a,d,p) = \frac{F_{2e}(x,d,p)+F_{2o}(x,d,p)}{\rho^{-1}(d,p)} \quad (12)$$

The region of convergence of the RIGF in (1) to produce (12) is assumed to be $0 \le x \le 37$; hence, the RIGF in (1) can be written as

$$\gamma(2p_1+2k,dx) = \frac{\gamma'(2p_1+2k,dx)}{\Gamma(2p_1+2k)}$$

$$= \frac{(dx)^{2p_1+2k}e^{-dx}}{\Gamma(2p_2+2k)}\left|\begin{matrix}\sum_{m=0}^{\infty}\frac{x_2^m}{(p_3+k)_m(p_2+k)_m}+\\ \frac{\frac{dx}{2}\sum_{m=0}^{\infty}\frac{x_2^m}{(p_4+k)_m(p_3+k)_m}}{(p_2+k)}\end{matrix}\right.$$

$$= \frac{\sqrt{\pi}\left(\frac{dx}{2}\right)^{2p_1+2k}e^{-dx}}{\Gamma(p_2+k)\Gamma(p_3+k)}\left|\begin{matrix}\sum_{m=0}^{\infty}\frac{x_2^m}{(p_3+k)_m(p_2+k)_m}+\\ \frac{\frac{dx}{2}\sum_{m=0}^{\infty}\frac{x_2^m}{(p_4+k)_m(p_3+k)_m}}{(p_2+k)}\end{matrix}\right.$$

$$= \frac{\sqrt{\pi}e^{-dx}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\frac{\left(\frac{dx}{2}\right)^{2k}\left|\begin{matrix}\sum_{m=0}^{\infty}\frac{x_2^m}{(p_3+k)_m(p_2+k)_m}+\\ \frac{\frac{dx}{2}\sum_{m=0}^{\infty}\frac{x_2^m}{(p_4+k)_m(p_3+k)_m}}{(p_2+k)}\end{matrix}\right.}{(p_3)_k(p_2)_k} \quad (13)$$

where the functions, $F_{2e}(x,d,p)$and $F_{2o}(x,d,p)$ are given by

$$F_{2e}(x,d,p) = \frac{\sqrt{\pi}e^{-dx}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}\sum_{k,m=0}^{\infty}\frac{(p_1)_k(1)_m}{(p_3)_{k+m}(p_2)_{k+m}}\frac{x_1^k}{k!}\frac{x_2^m}{m!}$$

$$= \frac{\sqrt{\pi}e^{-dx}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}F_{1:1;0}^{0:1;1}\left[\begin{matrix}-:p_1;1;\\ p_3:p_2;-\end{matrix}x_1,x_2\right]$$

$$= \frac{\sqrt{\pi}e^{-dx}x_2^{p_1}}{\Gamma(p_3)\Gamma(p_2)}K_{2e}(x,d,p) \quad (14)$$

$$F_{2o}(x,d,p) = \frac{\sqrt{\pi}x_2^{p_1}e^{-dx}}{\Gamma(p_4)\Gamma(p_3)}\frac{dx}{2}\sum_{k,m=0}^{\infty}\frac{(p_1)_k(1)_m}{(p_4)_{k+m}(p_3)_{k+m}}\frac{x_1^k}{k!}\frac{x_2^m}{m!}$$

$$= \frac{\sqrt{\pi}e^{-dx}x_2^{p_1}}{\Gamma(p_4)\Gamma(p_3)}\sqrt{x_2}F_{1:1;0}^{0:1;1}\left[\begin{matrix}-:p_1;1;\\ p_4:p_3;-;\end{matrix}x_1,x_2\right]$$

$$= \frac{\sqrt{\pi}e^{-dx}x_2^{p_1}}{\Gamma(p_4)\Gamma(p_3)}\sqrt{x_2}K_{2o}(x,d,p) \quad (15)$$

the Kampé de Fériet functions, $K_{2e}(x,d,p)$ and $K_{2o}(x,d,p)$ are discussed in great detail in Progri (2018, [5]), Progri (2019, [6]).

Substituting (14) and (15) into (12) and factoring out the common terms we obtain for $0 \le x \le 37$:

$$F_{\text{GBessel1}}(x;a,d,p) = \frac{\rho(d,p)x_2^{p_1}\sqrt{\pi}\left[K_{2e}(x,d,p)+\frac{\sqrt{x_2}K_{2o}(x,d,p)}{p_2}\right]}{e^{dx}\Gamma(p_2)\Gamma(p_3)} \quad (16)$$

Let us compare and contrast (16) with (see Progri (2018, [5]) (21)

$$F_{\text{GBessel1}}(x;a,d,p) = \frac{x^{2p_1}e^{-dx}\left[K_{2e}(x,d,p)+\frac{\sqrt{x_2}K_{2o}(x,d,p)}{p_2}\right]}{C_{11}^{-1}(d,p)} \quad (17)$$

Again (17) and (16) are identical if and only if (9)-(11) are identical, which are true anyways; therefore, (16) and (17) are identical; which proves again that the derivations in Progri (2018, [5]) are correct.

The issues in Progri (2018, [5]) were not the derivations; it had to do with understanding the regions of convergence for the Kampé de Fériet functions, $K_{2e}(x,d,p)$and $K_{2o}(x,d,p)$.

What was incorrect in Progri (2018, [5]) is that I incorrectly tried to apply (8) for values of $x > 1$ and (17) for values of $x > 37$; i.e., I failed to put these CFEs in the correct context and I failed to produce the correct CFEs for values of $x \ge 37$.

Lucky I was able to fix this problem here. Finally, for values of $x \ge 37$ the GBFD1K cdf is given by

$$F_{\text{GBessel1}}(x;a,d,p) = \frac{\sum_{k=0}^{\infty}\frac{1-\frac{(dx)^{2p+2k}\sum_{m=0}^{M}\frac{(-1)^m(2k-2p)_m}{(dx)^m}}{e^{dx}\Gamma(2p_1+2k)}(p_1)_k\frac{1}{d^{2k}}}{k!}}{\rho^{-1}(d,p)}$$

$$= \frac{F(d,p)-F_3(x,d,p)}{\rho^{-1}(d,p)} \quad (18)$$

where the hypergeometric function, $F(d,p)$, and the Kampé de Fériet functions, $F_3(x,d,p)$ (see Progri (2018, [5]), Progri (2019, [6])) can be computed from

$$F(d,p) = F[p_1,-;d^{-2}] = \rho^{-1}(d,p)^{\text{i}} \quad (19)$$

$$F_3(x,d,p) = \sum_{k=0}^{\infty}\frac{\frac{e^{-dx}(dx)^{2p+2k}\sum_{m=0}^{M}\frac{(-1)^m(2k-2p)_m}{(dx)^m}}{\Gamma(2p_1+2k)}(p_1)_k\frac{1}{d^{2k}}}{k!}$$

$$= \frac{\sqrt{\pi}e^{-dx}\left(\frac{dx}{2}\right)^{2p}}{\Gamma(p_1)\Gamma(p_2)}\sum_{k=0}^{\infty}\frac{\sum_{m=0}^{M}\frac{(-1)^m(2k-2p)_m}{(dx)^m}}{(p_2)_k}\frac{\frac{x^{2k}}{2^{2k}}}{k!} \quad (20)$$

In order to compute (29) the Pochhammer symbol, $(2k-2p)_m$, must be split into two: one for even m and one for odd m. From Progri (2018, [3] (11) we have

$$(2k-2p)_{m=2n} = \frac{\Gamma(2k-2p+2n)}{\Gamma(2k-2p)}$$

$$= \frac{\frac{4^{k+n}\Gamma(k-p+n)\Gamma\left(k-p+\frac{1}{2}+n\right)}{2^{-2p_1}\sqrt{\pi}}}{\frac{2^{2k-2p-1}\Gamma(k-p)\Gamma\left(k-p+\frac{1}{2}\right)}{\sqrt{\pi}}}$$

$$= 4^n(k-p)_n\left(k-p+\frac{1}{2}\right)_n \quad (21)$$

And for odd m

$$(2k-2p)_{m=2n+1}=\frac{\Gamma(2k-2p+2n+1)}{\Gamma(2k-2p)}$$

$$=\frac{\frac{4^{k+n+0.5}\Gamma\left(k-p+n+\frac{1}{2}\right)\Gamma(k-p+1+n)}{2^{-2p_1}\sqrt{\pi}}}{\frac{2^{2k-2p-1}\Gamma(k-p)\Gamma\left(k-p+\frac{1}{2}\right)}{\sqrt{\pi}}}$$

$$=\frac{\left(k-p+\frac{1}{2}\right)_n(k-p+1)_n}{2^{-1}\times4^{-n}(k-p)^{-1}} \quad (22)$$

Equation (20) will be split into two coefficients: even and odd

$$F_3(x,d,p)=F_{3e}(x,d,p)+F_{3o}(x,d,p) \quad (23)$$

$$F_{3e}(x,d,p)=\frac{\sqrt{\pi}\left(\frac{dx}{2}\right)^{2p}\sum_{k,n=0}^{\infty}\frac{(-p)_{k+n}\left(-p+\frac{1}{2}\right)_{k+n}(1)_n}{(-p)_k\left(-p+\frac{1}{2}\right)_k(p_2)_k}\frac{\frac{x^{2k}}{2^{2k}}}{k!}\frac{\frac{4^n}{(dx)^{2n}}}{n!}}{e^{dx}\Gamma(p_1)\Gamma(p_2)}$$

$$=\frac{\sqrt{\pi}x_2^pe^{-dx}\sum_{k,n=0}^{\infty}\frac{(-p)_{k+n}\left(-p+\frac{1}{2}\right)_{k+n}(1)_n}{(-p)_k\left(-p+\frac{1}{2}\right)_k(p_2)_k}\frac{x_1^k}{k!}\frac{x_2^{-n}}{n!}}{\Gamma(p_1)\Gamma(p_2)}$$

$$=\frac{\sqrt{\pi}x_2^pe^{-dx}F_{0:3;0}^{2:0;1}\left[\begin{matrix}-p,-p+\frac{1}{2}:-;1;\\-;-p,-p+\frac{1}{2},p_2;-;\end{matrix}x_1,x_2^{-1}\right]}{\Gamma(p_1)\Gamma(p_2)}$$

$$=\frac{\sqrt{\pi}x_2^pe^{-dx}}{\Gamma(p_1)\Gamma(p_2)}K_{3e}(x,d,p) \quad (24)$$

$$F_{3o}(x,d,p)=\frac{\sqrt{\pi}e^{-dx}\left(\frac{dx}{2}\right)^{2p}}{\Gamma(p_1)\Gamma(p_2)}\sum_{k,n=0}^{\infty}\frac{\frac{(-1)^{2n+1}(2k-2p)_{2n+1}}{(dx)^{2n+1}}}{(p_2)_k}\frac{\frac{x^{2k}}{2^{2k}}}{k!}$$

$$=\frac{p\sqrt{\pi}\left(\frac{dx}{2}\right)^{2p-1}\sum_{k,n=0}^{\infty}\frac{\left(-p+\frac{1}{2}\right)_{k+n}(-p+1)_{k+n}(1)_n}{(-p)_k\left(-p+\frac{1}{2}\right)_k(p_2)_k}\frac{x_1^k}{k!}\frac{x_2^{-n}}{n!}}{e^{dx}\Gamma(p_1)\Gamma(p_2)}$$

$$=\frac{p\sqrt{\pi}e^{-dx}x_2^{p-\frac{1}{2}}F_{0:3;0}^{2:0;1}\left[\begin{matrix}-p+\frac{1}{2},-p+1:-;1;\\-;-p,-p+\frac{1}{2},p_2;-;\end{matrix}x_1,x_2^{-1}\right]}{\Gamma(p_1)\Gamma(p_2)}$$

$$=\frac{p\sqrt{\pi}e^{-dx}x_2^{p-\frac{1}{2}}}{\Gamma(p_1)\Gamma(p_2)}K_{3o}(x,d,p) \quad (25)$$

where the Kampé de Fériet functions, $K_{3e}(x,d,p)$ and $K_{3o}(x,d,p)$ can be computed from (see Progri (2018, [5]), Progri (2019, [6])).

The computation of the Kampé de Fériet functions, $K_{3e}(x,d,p)$ and $K_{3o}(x,d,p)$, is simplified when

$$-p+\frac{1}{2}=-m\equiv-\{0,1,\cdots\}\Rightarrow p=[m\equiv\{0,1,\cdots\}]-\frac{1}{2} \quad (26)$$

Substituting (26) we obtain two simplified expressions of the Kampé de Fériet functions, $K_{3e}(x,d,p)$ and $K_{3o}(x,d,p)$, is simplified when

$$K_{3e}(x,d,p)=\sum_{k,n=0}^{\infty}\frac{(-p)_{k+n}\left(-p+\frac{1}{2}\right)_{k+n}(1)_n}{(-p)_k\left(-p+\frac{1}{2}\right)_k(p_2)_k}\frac{x_1^k}{k!}\frac{x_2^{-n}}{n!}$$

$$=\sum_{k,n=0}^{m-1,\infty}\frac{(-p+k)_n(-m+k)_n(1)_n}{(p_2)_k}\frac{x_1^k}{k!}\frac{x_2^{-n}}{n!}$$

$$=\sum_{k=0}^{m-1}\frac{F\left[\begin{matrix}k-m,k-p+1,1\\-,-,-\end{matrix};x_2^{-1}\right]}{(p_2)_k}\frac{x_1^k}{k!} \quad (27)$$

$$K_{3o}(x,d,p)=\sum_{k,n=0}^{\infty}\frac{\left(-p+\frac{1}{2}+k\right)_n(-p+1)_{k+n}(1)_n}{(-p)_k(p_2)_k}\frac{x_1^k}{k!}\frac{x_2^{-n}}{n!}$$

$$=\sum_{k,n=0}^{m-1,\infty}\frac{(-m+k)_n(-p+1+k)_n(-p+1)_k(1)_n}{(-p)_k(p_2)_k}\frac{x_1^k}{k!}\frac{x_2^{-n}}{n!}$$

$$=\sum_{k=0}^{m-1}\frac{(-p+1)_kF\left[\begin{matrix}k-m,k-p+1,1\\-,-,-\end{matrix};x_2^{-1}\right]}{(-p)_k(p_2)_k}\frac{x_1^k}{k!} \quad (28)$$

Substituting (24) and (25) into (23) and factoring out the common terms, yields:

$$F_3(x,d,p)=\frac{\sqrt{\pi}e^{-dx}x_2^p\left[p_1K_{3e}(x,d,p)+\frac{pp_1K_{3o}(x,d,p)}{\sqrt{x_2}}\right]}{\Gamma(p_2)\Gamma(p_3)} \quad (29)$$

Finally, substituting (29) into (18) produces the desired CFE of the GBFD1K cdf for large values of $37\le x<\infty$

$$F_{\text{GBessel1}}(x;a,d,p)=1-\frac{2p_1\left[K_{3e}(x,d,p)+\frac{pK_{3o}(x,d,p)}{\sqrt{x_2}}\right]}{C_{11}^{-1}(d,p)x^{-2p}de^{dx}} \quad (30)$$

Equation (30) is an original CFE that I should have been produced in Progri (2018, [5]) that provides the much needed CFE of the GBFD1K cdf by means of Kampé de Fériet functions (see Progri (2018, [5]), Progri (2019, [6])) for values of the variable $37\le x<\infty$.

Equations (8), (17), and (30) provide a very good approximation for the GBFD1K cdf by means of Kampé de Fériet functions (see Progri (2018, [5]), Progri (2019, [6])) for values of the variable $0\le x<\infty$.

The only limitation that these CFEs (8), (17), and (30) is that they are not as computationally efficient as they should be; however, they led to the discovery of the most computationally efficient algorithms that are produced in this landmark journal paper.

This concludes the discussion on the landmark computation of the GBFD1K cdf for non-integer values of a parameter by means of Kampé de Fériet functions (see Progri (2018, [5]), Progri (2019, [6])).

3 Landmark Efficient Computation of the GBFD1K for Non-Integer Values of a Parameter

In Sect. 2 I explained all the modifications that needed to be made in Progri (2016, [1]) (117) and Progri (2018, [5]) (23)-(25), (49)-(51) as it relates to the region of convergence of x.

The only drawback in Sect. 2 is that CFEs (8), (17), and (30) are not as computationally efficient as some other approaches that are explained in the numerical results Sect. It is, therefore, the purpose of this section to derive the most computationally efficient CFEs of the GBFD1K cdf for non-integer values of a parameter.

In Progri (2016, [1]) (107) to (114) I developed for the first time the landmark efficient computation of the GBFD1K for non-integer values of a parameter, p. However, I failed to provide the context when such an expression can be used and how it should be used.

First, I am rederiving Progri (2016, [1]) (113) in a slightly different way.

Employing an identity from (Nelson) (see Gradshteyn, Ryzhik, 2007 [17] pg. 901 ex. 8.356 5.) $\gamma(2p_1+2k,dx)$ as follows:

$$\gamma(2p_1+2k,dx)=\gamma(2p_1,dx)-e^{-dx}\sum_{n=0}^{2k-1}\frac{(dx)^{(2p_1+n)}}{\Gamma(2p_2+n)}$$

$$=\gamma(2p_1,dx)-\frac{e^{-dx}(dx)^{2p_1}}{\Gamma(2p_2)}\sum_{n=0}^{2k-1}\frac{(dx)^n}{(2p_2)_n} \quad (31)$$

$$\Gamma(2p_1+2k,dx)=\Gamma(2p_1,dx)+\frac{e^{-dx}(dx)^{2p_1}}{\Gamma(2p_2)}\sum_{n=0}^{2k-1}\frac{(dx)^n}{(2p_2)_n} \quad (32)$$

Next, substituting (31) into (1) produces:

$$F_{\mathrm{GBessel1}}(x;a,d,p)=\frac{\sum_{k=0}^{\infty}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!}}{\rho^{-1}(d,p)}$$

$$=\frac{\sum_{k=0}^{\infty}\frac{\left[\gamma(2p_1,dx)-\frac{e^{-dx}(dx)^{2p_1}\sum_{n=0}^{2k-1}\frac{(dx)^n}{(2p_2)_n}}{\Gamma(2p_2)}\right]\frac{(p_1)_k}{d^{2k}}}{k!}}{\rho^{-1}(d,p)}$$

$$=\frac{\left|\begin{array}{c}\gamma(2p_1,dx)\sum_{k=0}^{\infty}\frac{\frac{(p_1)_k}{d^{2k}}}{k!}-\\ \frac{(dx)^{2p_1}\sum_{k=0}^{\infty}\frac{\sum_{n=0}^{2k-1}\frac{(dx)^n}{(2p_2)_n}\frac{(p_1)_k}{d^{2k}}}{k!}}{e^{dx}\Gamma(2p_2)}\end{array}\right.}{\rho^{-1}(d,p)}$$

$$=L_1(d,p,x)-L_2(d,p,x) \quad (33)$$

where

$$L_1(d,p,x)=\frac{\sum_{k=0}^{\infty}\frac{\gamma(2p_1,dx)\frac{(p_1)_k}{d^{2k}}}{k!}}{\rho^{-1}(d,p)}=\frac{\gamma(2p_1,dx)\sum_{k=0}^{\infty}\frac{\frac{(p_1)_k}{d^{2k}}}{k!}}{\rho^{-1}(d,p)}$$

$$=\frac{\gamma(2p_1,dx)F[p_1,-;d^{-2}]}{\rho^{-1}(d,p)}\equiv\gamma(2p_1,dx) \quad (34)$$

$$L_2(d,p,x)=\frac{e^{-dx}(dx)^{2p_1}\sum_{k=0}^{\infty}\sum_{n=0}^{2k-1}\frac{(dx)^n}{(2p_2)_n}\frac{(p_1)_k}{k!d^{2k}}}{\Gamma(2p_2)\rho^{-1}(d,p)}$$

$$=\frac{C_{11}(d,p)x^{2p_1}}{e^{dx}}\sum_{k=0}^{\infty}\sum_{n=0}^{2k-1}\frac{(dx)^n}{(2p_2)_n}\frac{(p_1)_k}{k!d^{2k}} \quad (35)$$

In (34) we applied Progri (2016, [1]) (105) and we obtain the same answer as in Progri (2016, [1]) (111) and (113).

Equation (35) is identical to Progri (2016, [1]) (112).

In (34) we can apply the CFEs of the RIGF that are carefully examined in Progri (2022, [15]).

$$\gamma(2p_1,dx)=\frac{(dx)^{(2p_1)}\Phi\left[{2p_1 \atop 2p_1+1};-dx\right]}{(2p_1)\Gamma(2p_1)};\ 0\le dx<1$$

$$=\frac{(dx)^{(2p_1)}\Phi\left[{2p_1 \atop 2p_1+1};-dx\right]}{\Gamma(2p_2)};\ 0\le dx<1$$

$$=\frac{(dx)^{(2p_1)}e^{-dx}\Phi\left[{1 \atop 2p_1+1};dx\right]}{(2p_1)\Gamma(2p_1)};\ 0\le dx<20$$

$$=\frac{(dx)^{(2p_1)}e^{-dx}\Phi\left[{1 \atop 2p_1+1};dx\right]}{\Gamma(2p_2)};\ 0\le dx<20$$

$$=1-\frac{(dx)^{2p}\sum_{k=0}^{M}\frac{(-1)^k(1-2p_1)_k}{(dx)^k}}{e^{dx}\Gamma(2p_1)};\ 20\le dx<\infty \quad (36)$$

Substituting (36) into (34) produces the solution for $K_1(d,p,x)$

$$L_1(d,p,x)=\Phi\left[{2p_1 \atop 2p_1+1};-dx\right];\ 0\le dx<1$$

$$\cong\frac{(dx)^{(2p_1)}\sum_{n=0}^{N}\frac{(-1)^n(2p_1)_n(dx)^n}{(2p_2)_n n!}}{\Gamma(2p_2)} \quad (37)$$

$$L_1(d,p,x)=\frac{\Phi\left[{1 \atop 2p_1+1};dx\right]}{(dx)^{-(2p_1)}\Gamma(2p_2)e^{dx}};\ 0\le dx<20 \quad (38)$$

$$L_1(d,p,x)=1-\frac{\sum_{k=0}^{M}\frac{(1-2p_1)_k}{(-1)^{-k}(dx)^k}}{(dx)^{-2p}e^{dx}\Gamma(2p_1)};\ 20\le dx<\infty \quad (39)$$

Equations (32)-(34), (36)-(39) provide the CFEs of the GBFD1K for non-integer values of a parameter p.

It turns out that (36) or (38) and (39) are only useful for values of $24\le dx<\infty$. Therefore, in constructing the fastest (i.e., the most computationally efficient algorithm) and the most accurate algorithm I propose the Progri landmark efficient computation of the GBFD1K for non-integer values of a parameter, p as follows:

$$F_{\mathrm{GBessel1}}(x;a,d,p)=\left|\begin{array}{ll}(1) & 0\le x<24^{\mathrm{ii}}\\ (33) & 24\le x<\infty\end{array}\right.; \quad (40)$$

It appears that the Progri landmark efficient computation of the GBFD1K for non-integer values of a parameter, p was derived entirely in Progri (2016, [1]) (88) and (113). I just was never able to test all the possible options at that time. The main purpose of this publication is to show once more that all the derivations in Progri (2016, [1]) are correct. It is that the contest of these CFEs needed to be clarified.

4 Kampé de Fériet Reduction Formulae

Landmark computation of the GBFD1K via Kampé de Fériet functions and the landmark efficient computation of the GBFD1K for non-integer values of a parameter provide the opportunity to produce more reduction formulae of the Kampé de Fériet functions.

Equating (1) with (7) produces for values of $0 \leq x < 1$ as follows

$$K_{1e}(x,d,p) - \frac{dxp_1K_{1o}(x,d,p)}{p_2} = \frac{\sum_{k=0}^{\infty}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!}}{\sqrt{\pi}x_2^{p_1}\Gamma^{-1}(p_3)\Gamma^{-1}(p_2)} \quad (41)$$

Similarly, equating (1) with (16) yields for values of $0 \leq x < 37$ as given below

$$K_{2e}(x,d,p) + \frac{\sqrt{x_2}K_{2o}(x,d,p)}{p_2} = \frac{\sum_{k=0}^{\infty}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!}}{e^{-dx}\sqrt{\pi}x_2^{p_1}\Gamma^{-1}(p_3)\Gamma^{-1}(p_2)} \quad (42)$$

Next, equating (16) with (30) produces for values of $24 \leq x < 37$ as follows

$$K_{2e}(x,d,p) + \frac{\sqrt{x_2}K_{2o}(x,d,p)}{p_2} = \frac{e^{dx}[\gamma(2p_1,dx) - L_2(d,p,x)]}{\sqrt{\pi}x_2^{p_1}\Gamma^{-1}(p_3)\Gamma^{-1}(p_2)\rho^{-1}(d,p)} \quad (43)$$

Finally, equating (1) with (27) we obtain for values of $37 \leq x < \infty$ as given below

$$\frac{K_{3e}(x,d,p)+\frac{pK_{3o}(x,d,p)}{\sqrt{x_2}}}{d} = \frac{1-\frac{\sum_{k=0}^{\infty}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!}}{\rho^{-1}(d,p)}}{2p_1C_{11}(d,p)x^{2p}e^{-dx}} \quad (44)$$

We cannot do any better than any of these formulae for the computation of the Kampé de Fériet functions.

5 Numerical, Theoretical Results

I made major improvements in the last six years since the first publication of Progri (2016, [1]) as it relates to the computational wisdom, knowledge, and understanding.

I attempt to describe most of computational wisdom, knowledge, and understanding as it relates to the computation of the GBFD1K cdf either vial MATLAB BIF or Giftet BIF.

In the current journal paper the truth (GBFD1K cdf) can be computed via two different ways:

1. Via MATLAB *integral* BIF
2. Via MATLAB *gammainc* BIF

The linear approximation is eliminated in the landmark computation. The linear approximation was necessary in the beginning when I needed to understand the development of the computational models, their properties, singularities, etc. Once it served its purpose, the linear approximation is replaced with the computation via MATLAB *gammainc* BIF.

The landmark computation of the RIGF produced another landmark of the GBFD1K is accomplished via three other options via Giftet:

1. *pgamainc* MATLAB function. This is new computation that was neither published in Progri (2016, [1]) nor in Progri (2018, [5]).
2. *kamdefer* MATLAB function. This option is only useful for values of $0 \leq x < 24$ which is also a function of the non-integer parameter p. Outside this interval the recursive implementation via the *progril2* MATLAB function is much faster and more accurate than *kamdefer* MATLAB function will even be.
3. *pgamainc* MATLAB function for $0 \leq x < 24$ and *progril2* MATLAB for $24 \leq x < \infty$.

Therefore, the landmark computation of the GBFD1K is performed via five different options. The MATLAB *integral* BIF is considered the truth. When Option one is equal to

1. one via MATLAB *gamainc* BIF, $0 \leq x < \infty$,
2. two via Giftet *pgamainc* BIF, $0 \leq x < \infty$,
3. three via Giftet *kamdefer* BIF, $0 \leq x < 24$ and Giftet *progril2* BIF for $24 \leq x < \infty$,
4. four via Giftet *pgamainc* BIF for $0 \leq x < 24$ and Giftet *progril2* BIF for $24 \leq x < \infty$.

These options are explained further in great detail in the following five subsections.

5.1 Computation of the GBFD1K cdf via MATLAB integral BIF

The very first computation of the GBFD1K cdf in Progri 2016 [1] was by means of the MATLAB *integral* BIF.

However, the implementation of the MATLAB *integral* BIF required improvements of the GBFD1K pdf that was done in Progri (2022, [13]) as follows:

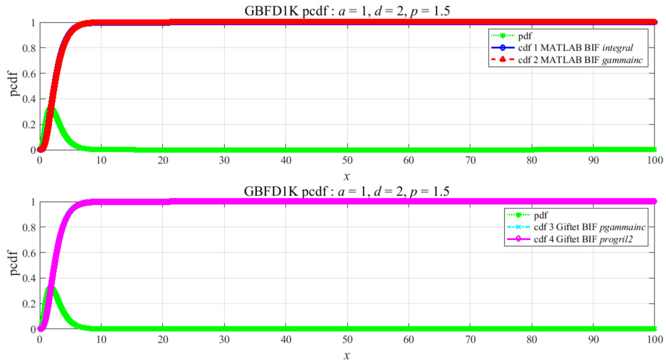
$$f_{\text{GBessel1}}(x;a,d,p) = \left|\begin{matrix} f(x) & 0 \leq x < 100 \\ \exp[\log\_f(x)] & 100 \leq x < \infty \end{matrix}\right. \quad (45)$$

Where

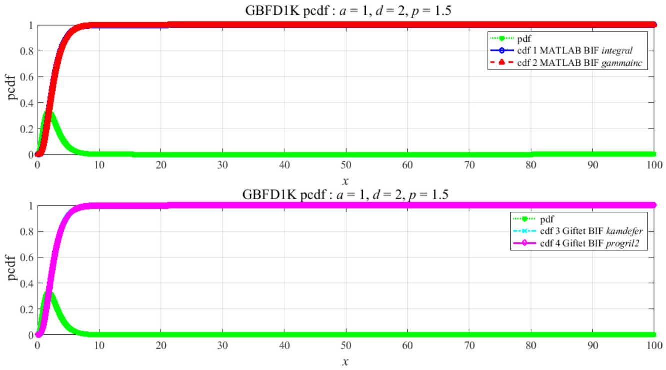
$$f(x) = \frac{x^p e^{-xd} I_p(x)}{C_1(p,d)}, 0 \leq x < 100 \quad (46)$$

$$\begin{aligned} \log\_f(x) &= \log[f(x)],\ 100 \leq x < \infty \\ &= p\log x - xd + \log I_p(x) - \log\_C_1 \quad (47) \end{aligned}$$

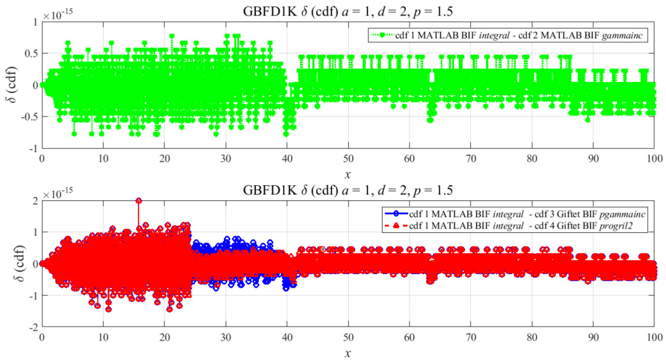
$$C_1(p,d) = \frac{2^p\Gamma(p_1)}{\sqrt{\pi}|1-d^2|^{p_1}} \quad (48)$$



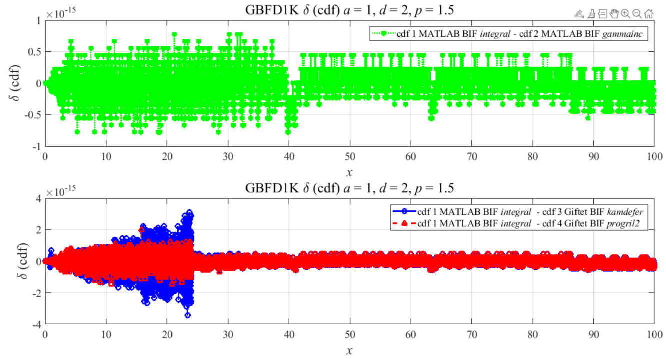
(a) (top) cdf 1: *integral*, cdf 2: *gammainc* op = [1 3 28 20], (bottom) cdf 3: *pgammainc* op = [2 3 110 28], cdf 4: *progril2* op = [4 3 120 28].



(c) (top) cdf 1: *integral*, cdf 2: *gammainc* op = [1 3 28 20], (bottom) cdf 3: *kamdefer* op = [3 4 28 80], cdf 4: *progril2* op = [4 3 120 28].



(b) (top) error between cdf 1: *integral* minus cdf 2: *gammainc* (a), (bottom) error between cdf 1: *integral* minus cdf 3: *pgammainc* (a); error between cdf 1: *integral* cdf 4: *progril2*.



(d) (top) error between cdf 1: integral minus cdf 2: *gammainc* (c), (bottom) error between cdf 1: *integral* minus cdf 3: *kamdefer* (c); error between cdf 1: *integral* minus cdf 4: *progril2*.

Figure 1: GBFD1K pdf and cdf and cdf error for MATLAB BIF, *integral*, *gammainc*, and Giftet *pgammainc*, *kamdefer*, and *progril2* for $a = 1$, $d = 2$, and $p = 1.5$.

$$\log\_C_1 = p\log 2 + \log[\Gamma(p_1)] - \frac{\log(\pi)}{2} - \frac{\log(|1-d^2|)}{p_1^{-1}} \qquad (49)$$

MATLAB has *logbesseli* BIF that is employed to compute $\log I_p(x)$ and *gammaln* BIF that is also used to compute $\log[\Gamma(p_1)]$.

These improvements on the computation of the pdf will eliminate the singularities when the MATLAB integral BIF is used for values of $100 \le x < \infty$.

Simulation results of the MATLAB *integral* BIF are shown in Figs. 1(a), (c), 2(a), (c), and 3(a) (top).

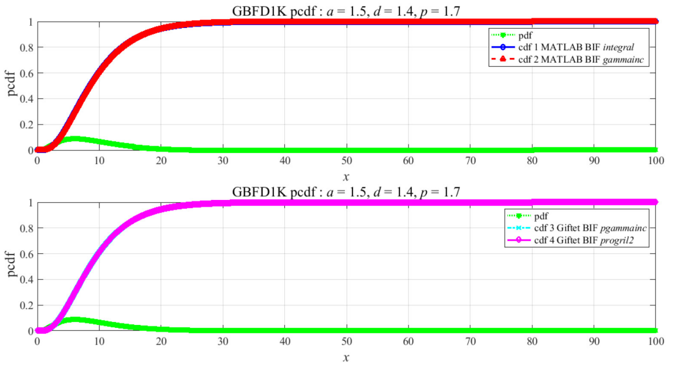
Because the MATLAB *integral* BIF is not used specifically for the computation of the GBFD1K; hence, it is not expected to be fast; however, it is expected to be accurate. In Progri (2022, [15]), I provided sufficient detail that the MATLAB *integral* BIF is a better approximation of the RIGF than the MATLAB *gammainc* BIF for values of the parameter greater than one.

5.2 Landmark Computation of the GBFD1K cdf via MATLAB gammainc BIF

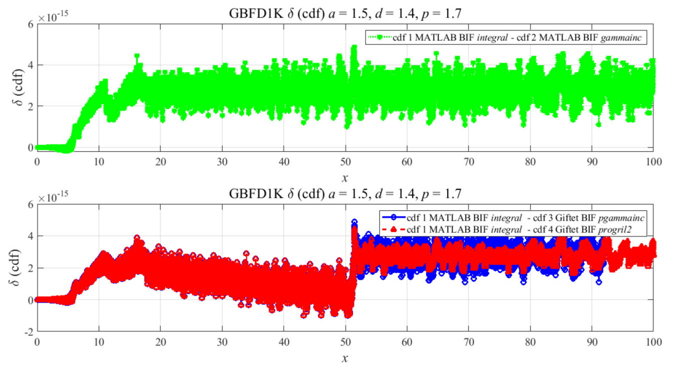
The decision to replace the linear approximation cdf with the MATLAB *gammainc* BIF was a very good decision.

In the file Giftet *progril2 subfunction1* the implementation of the MATLAB *gammainc* BIF is accomplished via three options which correspond to option 2 taking on three values; hence, then option 2 is equation to:

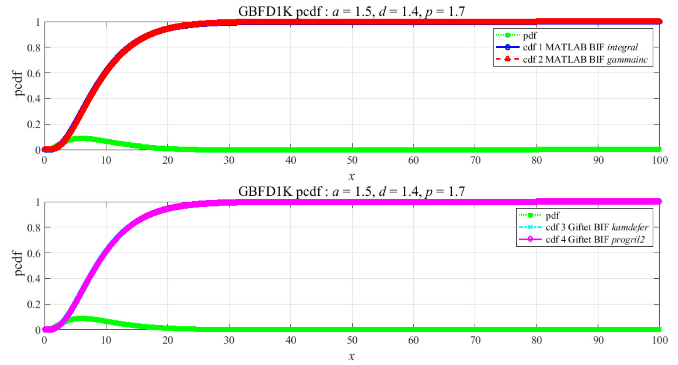
1. one, it means that the implementation is accomplished using MATLAB BIF functions such as factorial, or pocchammer, etc. This option is needed in the beginning to make sure that the implementation of (1) is accurate for small values of x and a small number of terms.
2. two, then the factorial, power, or pocchammer is implemented recursively. This option is supposed to be more accurate and faster than the first option because it eliminates the multiplication and division with very larger numbers.
3. Three, then the use of recursive algorithm makes use of *log* and *exp* MATLAB BIF because the logarithm turns the product into a summation and the division into a subtraction. Since, it is faster to add and subtract numbers than to multiply and or divide; hence, this option is supposed to be faster than the first two options.



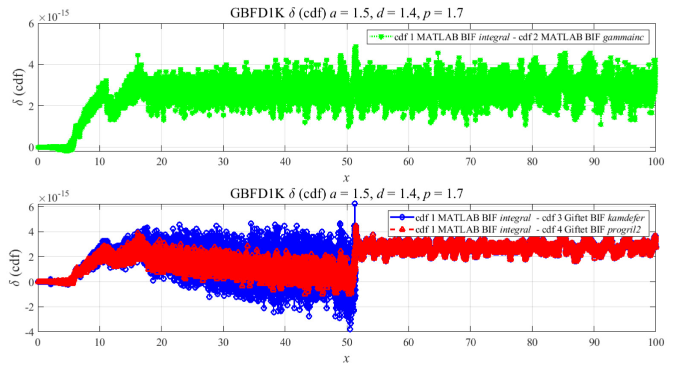
(a) (top) cdf 1: *integral*, cdf 2: *gammainc* op = [1 3 58 20], (bottom) cdf 3: *pgammainc* op = [2 3 200 78], cdf 4: *progril2* op = [4 3 150 68].



(b) (top) error between cdf 1: *integral* minus cdf 2: *gammainc* (a), (bottom) error between cdf 1: *integral* minus cdf 3: *pgammainc* (a); error between cdf 1: *integral* cdf 4: *progril2*.



(c) (top) cdf 1: integral, cdf 2: gammainc op = [1 3 58 20], (bottom) cdf 3: kamdefer op = [3 4 78 100], cdf 4: progril2 op = [4 3 150 68].



(d) (top) error between cdf 1: integral minus cdf 2: *gammainc* (c), (bottom) error between cdf 1: *integral* minus cdf 3: *kamdefer* (c); error between cdf 1: *integral* minus cdf4: *progril2*.

Figure 2: GBFD1K pdf and cdf and cdf error for MATLAB BIF, *integral*, *gammainc*, and Giftet *pgammainc*, *kamdefer*, and *progril2* for $a = 1.5$, $d = 1.4$, and $p = 1.7$.

What are the main benefits of this option? Because the MATLAB *gammainc* BIF is already optimized for accuracy and speed, the only unknown to compute (1) is the number of terms. The third element in the option vector is the integer number, K, that denotes the number of terms to approximate (1).

The option vector corresponding to the implementation of MATLAB *gammainc* BIF is as an example: the op vector looks as follows: op = [1 1 28 20]; [1 2 28 20]; [1 3 28 20].

The first integer number 1 corresponds to option 1 equal to 1 which means that MATLAB *gammainc* BIF is used; i.e., $op(1) = 1$.

The second integer number 1, 2, 3 correspond to option 2 equal to 1, 2, 3; $op(2) = \{1,2,3\}$.

The integer number 28 correspond to the number of terms K in (1); i.e., $op(3) = 28$.

The integer number 20 is not needed for option 1 equal to 1. It is needed for options 1 equal to 2, 3, and 4.

Simulation results of the MATLAB *gammainc* BIF are shown in Figs. 1(a), (c), 2(a), (c), and 3(a) (top).

This option is supposed to be the fastest and the most accurate of all the options. Nevertheless, its performance on both speed and accuracy will be assessed in this section.

5.3 *Landmark Computation of the GBFD1K cdf via Giftet pgammainc Function*

The landmark computation of the GBFD1K cdf via Giftet *pgammainc* function is an entirely new computation that resulted from Progri (2022, [15]).

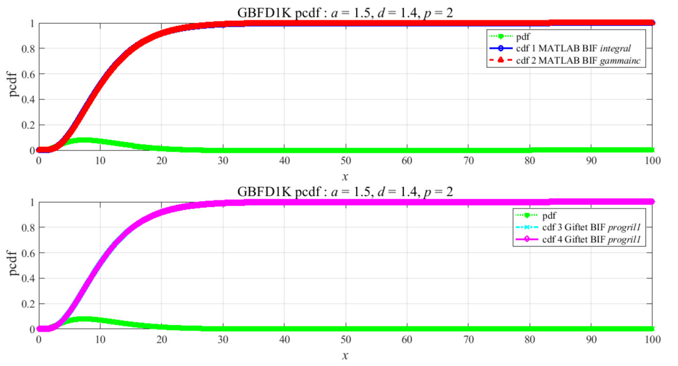
The option vector corresponding to the implementation of Giftet *pgammainc* BIF is as an example: the op vector looks as follows: op = [2 1 120 28]; [2 2 120 28]; [2 3 120 28].

The first integer number 2 corresponds to option 1 equal to 2 which means that Giftet *pgammainc* BIF is used; i.e., i.e., $op(1) = 2$.

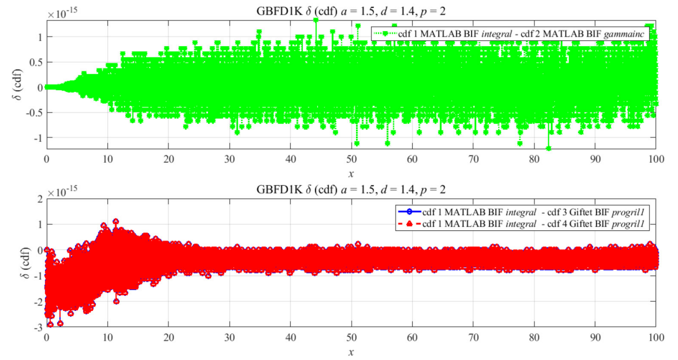
The second integer number 1, 2, 3 correspond to option 2 equal to 1, 2, 3 which are explained in Subsect. 5.3; i.e., $op(2) = \{1,2,3\}$.

The integer number 120 corresponds to the number of terms that are needed to approximate *pgammainc* (see Progri (2022, [15])); $op(3) = 120$.

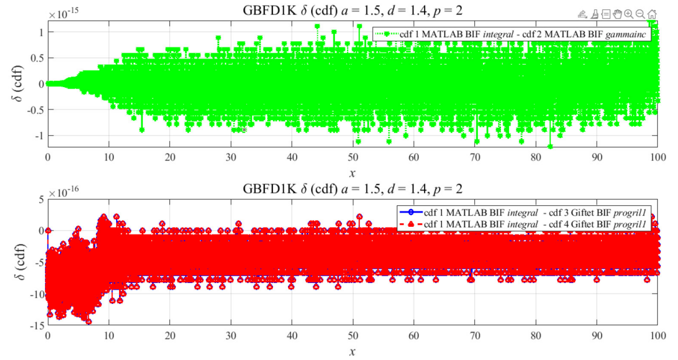
The integer number 28 correspond to the number of terms K



(a) (top) cdf 1: *integral*, cdf 2: *gammainc* $op = [13\ 60\ 20]$, (bottom) cdf 3: *progril1* $op = [2\ 3\ 60\ 29]$, cdf4: *progril1* $op = [4\ 4\ 60\ 58]$.



(b) (top) error between cdf 1: *integral* minus cdf 2: *gammainc* (a), (bottom) error between cdf 1: *integral* minus cdf 3: *progril1* (a); error between cdf 1: *integral* cdf4: *progril1*.



(c) (top) same as (b); (bottom) same as (b) but $op(2) = 4$.

Figure 3: GBFD1K pdf and cdf and cdf error for MATLAB BIF, *integral*, *gammainc*, and Giftet *progril1* for $a = 1.5$, $d = 1.4$, and $p = 2$.

in (1); $op(4) = 28$. Since, K was already optimized in Subsect. 5.1 then it only remains to optimize $op(3) = 120$.

Simulation results of the Giftet *pgammainc* BIF are shown in Figs. 1(a), 2(a) (bottom).

This option is supposed to compete with MATLAB *gammainc* BIF in both performance accuracy and speed.

5.4 Landmark Computation of the GBFD1K cdf via Giftet Kampé de Fériet Functions

The landmark computation of the GBFD1K cdf via Giftet *kamdefer* function is a significantly improved computation that originated in Progri (2016, [1]) to Progri (2022, [15]).

TABLE I: THE QUANTITATIVE COMPUTATIONAL PERFORMANCE

GBFD1K cdf:			
CFE 1 *int*; CDF2 *ginc*; CDF2 *pginc* vs. CFE 4 *pl2*			
$a = 1, d = 2, p = 1.5, 0 \le x \le 100$ $\sum(\cdot) = 28$ terms			
integral (s)	*gammain* (ms)	*pgammainc* (ms)	*progril2* (ms)
14.79	253.3	113.8	175.0 Op 3
14.66	259.9	120.1	139.7 Op 4
CFE 1 *int*; CDF2 *ginc*; CDF2 *pginc* vs. CFE 4 *pl2*			
$a = 1.5, d = 1.4, p = 1.7, 0 \le x \le 100$ $\sum(\cdot) = 78$ terms			
(s)	(ms)	(ms)	(ms)
12.86	787.3	600.2	593.6 Op 3
12.86	783.9	599.1	606.7 Op 4
CFE 1 *int*; CDF2 *ginc*; CDF2 *pll* vs. CFE 4 *pll*			
$a = 1.5, d = 1.4, p = 2.0, 0 \le x \le 100$ $\sum(\cdot) = 60$ terms			
(sec)	(ms)	(ms)	(ms)
11.40	793.8	169.6	141.2 Op 3
12.86	794.0	594.2	509.4 Op 4

The option vector corresponding to the implementation of Giftet *kamdefer* BIF is as an example: the op vector looks as follows:

$op = [3\ 1\ 80\ 80];\ [3\ 2\ 80\ 80];\ [3\ 3\ 28\ 80];\ [3\ 4\ 28\ 80]$.

The first integer number 3 corresponds to option 1 equal to 3 which means that Giftet *kamdefer* BIF is used; i.e., $p(1) = 3$.

The second integer number 1, 2, 3, 4 correspond to option 2 equal to 1, 2, 3, 4 which are explained in Subsect. 5.3; i.e., $op(2) = \{1,2,3,4\}$. Further details of the computation of the *kamdefer* the reader may find in Progri (2018, [5]), Appendix A.

The integer number 80 or 28 corresponds to the number of terms that are needed in the outer summation of *kamdefer* (see Progri (2022, [15])), of terms K; $op(3) = \{80,28\}$.

The integer number 80 corresponds to the number of terms in the inner summation in (1); $op(4) = 80$.

In the current implementation of Giftet *kamdefer* function, when $op(2) = \{1,2\}$ then $op(3)$ is used for $op(4)$;; it does not matter what $op(4)$ is; however, then $op(2) = \{3,4\}$ then $op(3) = op(4)$ or $op(3) \neq op(4)$. This allowed for $op(3) = 28$; hence, reducing the computation time while maintaining the same performance as for $op(3) = 80$.

The Giftet *kamdefer* function is only used for $0 \le x \le 24$, for values of $24 \le x < \infty$ the option 1 equal to 4 is used instead. This option is discussed extensively next. Although the computation via the Giftet *kamdefer* function is not expected to be as fast and as accurate as via either the MATLAB *gammainc* BIF or Giftet *pgammainc* BIF it is an option that is a unique computation and as such it should merit our attention of detail.

Simulation results of the Giftet *kamdefer* BIF are shown in Figs. 1(c) and 2(c) (bottom).

This option is reduced significantly when (26)-(28) occur.

The performance in both computation time and accuracy will be assessed and it should prove a good option as a result of the significantly new improvements that were made in this journal paper.

5.5 Landmark Computation of the GBFD1K cdf via Progri (40) CFE

The landmark computation of the GBFD1K cdf via Progri (40) CFE is meant to be the best performance in computation time and nearly as good a performance in accuracy as either MATLAB *gammainc* BIF or Giftet *pgammainc* BIF as discussed in Progri (2022, [15]).

The option vector corresponding to the implementation of Giftet *progril2* BIF is as an example: the op vector looks as follows:

$$op = \begin{vmatrix} [4\ 1\ 80\ 28];\ [4\ 2\ 80\ 28]; \\ [4\ 3\ 120\ 28];\ [4\ 4\ 120\ 28] \end{vmatrix}.$$

The first integer number 4 corresponds to option 1 equal to 4 which means that Giftet (40) BIF is used; i.e., $op(1) = 4$.

The second integer number 1, 2, 3, 4 correspond to option 2 equal to 1, 2, 3, 4; i.e., $op(2) = \{1,2,3,4\}$; When $op(2) = \{1,2\}$ then Giftet *kamdefer* function is used (see Subsect. 5.4), and then $op(2) = \{3,4\}$ then Giftet *pgammainc* function is used (see Subsect. 5.3).

The third integer is the number of terms that is used either for Giftet *kamdefer* function; i.e., $op(3) = 80$ or Giftet *pgammainc* function; i.e., $op(3) = 120$.

The fourth integer is the number of the number of terms K in (1); i.e., $op(3) = 28$.

As an example, when the simulation option vector is set to $op = [4\ 3\ 120\ 28]$ or $op = [4\ 4\ 120\ 28]$ it is supposed to be the fastest and most accurate computation of the GBFD1K cdf via MATLAB.

Simulation results of the Giftet *progril2* BIF are shown in Figs. 1(a), (c) and 2(a), (c) (bottom).

In the simulation results I will refer to the option vector and the reader should be able to understand why it was utilized and what it means.

In Fig. 1 the computation of the GBFD1K pdf and cdf and cdf error for MATLAB BIF, *integral*, *gammainc*, and Giftet *pgammainc*, *kamdefer*, and *progril2* for $a = 1$, $d = 2$, and $p = 1.5$ is depicted.

(a) (top) cdf 1: *integral*, cdf 2: *gammainc* $op = [1\ 3\ 28\ 20]$, (bottom) cdf 3: *pgammainc* $op = [2\ 3\ 110\ 28]$, cdf 4: *progril2* $op = [4\ 3\ 120\ 28]$.

(b) (top) error between cdf 1: *integral* minus cdf 2: *gammainc* (a), (bottom) error between cdf 1: *integral* minus cdf 3: *pgammainc* (a); error between cdf 1: *integral* cdf 4: *progril2*.

(c) (top) cdf 1: *integral*, cdf 2: *gammainc* $op = [1\ 3\ 28\ 20]$, (bottom) cdf 3: *kamdefer* $op = [3\ 4\ 28\ 80]$, cdf 4: *progril2* $op = [4\ 3\ 120\ 28]$.

(d) (top) error between cdf 1: integral minus cdf 2: *gammainc* (c), (bottom) error between cdf 1: *integral* minus cdf 3: *kamdefer* (c); error between cdf 1: *integral* minus cdf 4: *progril2*.

Figure 2 displays the computation of the GBFD1K pdf and cdf and cdf error for MATLAB BIF, *integral*, *gammainc*, and Giftet *pgammainc*, *kamdefer*, and *progril2* for $a = 1.5$, $d = 1.4$, and $p = 1.7$.

(a) (top) cdf 1: *integral*, cdf 2: *gammainc* $op = [1\ 3\ 58\ 20]$, (bottom) cdf 3: *pgammainc* $op = [2\ 3\ 200\ 78]$, cdf 4: *progril2* $op = [4\ 3\ 150\ 68]$.

(b) (top) error between cdf 1: *integral* minus cdf 2: *gammainc* (a), (bottom) error between cdf 1: *integral* minus cdf 3: *pgammainc* (a); error between cdf 1: *integral* cdf 4: *progril2*.

(c) (top) cdf 1: integral, cdf 2: gammainc $op = [1\ 3\ 58\ 20]$, (bottom) cdf 3: kamdefer $op = [3\ 4\ 78\ 100]$, cdf 4: progril2 $op = [4\ 3\ 150\ 68]$.

(d) (top) error between cdf 1: integral minus cdf 2: *gammainc* (c), (bottom) error between cdf 1: *integral* minus cdf 3: *kamdefer* (c); error between cdf 1: *integral* minus cdf4: *progril2*.

When p is a non-integer, the computation of the GBFD1K cdf via MATLAB *gammainc* BIF is the most accurate followed by Giftet *progril2* BIF, followed by Giftet *pgammainc* BIF, and then Giftet *kamdefer* BIF.

Figure 3 illustrates the computation of GBFD1K pdf and cdf and cdf error for MATLAB BIF, *integral*, *gammainc*, and Giftet *progril1* for $a = 1.5$, $d = 1.4$, and $p = 2$.

(a) (top) cdf 1: *integral*, cdf 2: *gammainc* $op = [13\ 60\ 20]$, (bottom) cdf 3: *progril1* $op = [2\ 3\ 60\ 29]$, cdf4: *progril1* $op = [4\ 4\ 60\ 58]$.

(b) (top) error between cdf 1: *integral* minus cdf 2: *gammainc* (a), (bottom) error between cdf 1: *integral* minus cdf 3: *progril1* (a); error between cdf 1: *integral* cdf4: *progril1*.

(c) (top) same as (b); (bottom) same as (b) but $op(2) = 4$.

When p is an integer, the computation of the GBFD1K cdf via Giftet *progril1* BIF, is the most accurate followed by followed by MATLAB *gammainc* BIF.

In Tab. I, the quantitative computational performance of the simulation results of Figs. 1-3 are summarized. As illustrated from the results of Tab. 1, the MATLAB *integral* BIF is the slowest, followed by the MATLAB *gammainc* BIF. For the results generated for Fig. 1, Giftet *pgammainc* BIF is the fastest, and corresponding to the results of Fig. 2, Giftet *progril2* BIF is the fastest. When the parameter p is an integer, the computation of the GBFD1K via Giftet *progril1* BIF is faster than MATLAB *gammainc* BIF.

6 Conclusions

In this landmark journal paper, I have provided some of the most amazing derivations for computing the GBFD1K cdf via five options.

In the first option the computation is done via MATLAB *integral* BIF. For this option to work for large values of the variable x, the computation of the GBFD1K pdf needed modifications. Now even though the MATLAB *integral* BIF is slower than the all-other options it is an option that does not require a lot of other information such as number of terms or the option for other subfunctions within the option. The MATLAB *integral* BIF is only sensitive to the length of the vector and the total number of points of the vector, x.

The second option is the computation of the GBFD1K cdf via MATLAB *gammainc* BIF in (1). The only modification this option needs is the number of terms K. As shown from the simulation results and the computation time, the MATLAB *gammainc* BIF is just as good as the MATLAB *integral* BIF but fourteen times faster.

The third option is the option via the GBFD1K cdf via Giftet *pgammainc* BIF in (1). This was an option that was developed as a result to investigate some of the issues related to the convergence of Progri (2018, [5]) for large values of the variable x. Details of this option are provided in Progri (2022, [15]). As indicated by both the simulation results and the computation speed, the Giftet *pgammainc* BIF is just as accurate and faster sometimes even twice as fast as the MATLAB *gammainc* BIF because it allows for optimization of the number of terms inside the *pgammainc* BIF. Nevertheless, this may be a drawback because it is currently perfomed manually; however, future work will aim towards optimization of this option.

The fourth option is the computation of the GBFD1K cdf via the Giftet Kampé de Fériet Functions. All the issues related to the earlier convergence of this option were fully investigated and eliminated. This option was significantly improved as a result of adding simplifications for large values of x. This option has added tremendous values to the body of wisdom, knowledge, and understanding that did not exist prior to Progri (2016, [1]). Although this option is much faster than the MATLAB *integral* BIF, it is not nearly as fast as the other options that were added that did not exist prior to Progri (2016, [1]).

Finally, the last option is the computation of the GBFD1K cdf via the Giftet *progril2* function. This option is supposed to be the best of the best because it is meant to provide a combination of the very best options from Giftet *pgammainc* BIF and Giftet Kampé de Fériet Functions.

These options will continue to be investigated towards finding a better optimization algorithm that will be part of the *Giftet Indoor Geolocation Systems—Theory and Simulation* toolbox.

7 Acknowledgement

This work was supported by Giftet Inc. executive office.

I want to profoundly thank the MathWorks at Natick, Massachusetts for providing a sponsored MATLAB licence [20] to Giftet Inc. as part of the Indoor Geolocation Systems MATLAB Library development that will enable the results of this work to be published in Dr. Progri pioneer publication *Indoor Geolocation Systems—Theory and Applications*. *Vol. I* (Not yet available in print) [20].

This journal paper is dedicated to four special men in my life: my grandfather, Xhevdet Progri, my dear father, Fiqiri Progri, my father's first cousin Dr. Peter Demir, and Qazim Demir, the brother of my grandfather, Xhevdet Progri.

I would profoundly like to thank my grandmothers Behije Progri on my father's side and Hane Alicka from my mother's side. As a young boy I felt very lucky because I felt I was blessed to have both my grandparents on my father's side and on my mother's side. But, more importantly I felt I was very fortunate and they felt I was very fortunate. It took me a few decades to understand why I was very fortunate.

As I came to know the Lord and as I came to read and learn from the Bible, I understood why I was so fortunate. It has to do with the day of my birthday. Since I was born on the sixteenth day, I came to realize what an important number the number sixteen is in the Bible. Genesis 1:16, Mathew 2:11 (my

mother's birthday), Matthew 7:16, (John 1:16, 3:16, 4:16, 8:16),

This journal paper is also dedicated to the Golden Bear, Jack Nicklaus, the greatest golfer of all time. Needless to say, I have fallen in love with his masterpiece book, *Golf My Way*. Moreover, Jack Nicklaus [18] reminds me of my grandfather who I loved him very much.

Finally, I am also dedicating this journal paper to our most beloved President Ronald Reagan, who reminds me of my grandfather, [19] who spoke from the heart, who spoke the truth, who was a staunch supporter of personal freedom, the greatest leader of the free world, and perhaps the greatest critic of the wasteful government control and spending[iii].

"*Government is not the answer, government is the problem*."—Ronald Reagan

8 References

[1] I.F. Progri, "Generalized Bessel function distributions," *J. Geol. Geoinfo. Geointel.,* vol. 2016, article ID 2016071602, 15 pg., Nov. 2016. DOI: https://doi.org/10.18610/JG3.2016.071602

[2] I.F. Progri, "Exponential generalized Beta distribution," *J. Geol. Geoinfo. Geointel.,* vol. 2016, article ID 2016071603, 18 pg., Nov. 2016. DOI: https://doi.org/10.18610/JG3.2016.071603

[3] I.F. Progri, "Hypergeometric function partial derivatives," vol. 2016, article ID 2016071604, 21 pg., Nov. 2016. DOI: https://doi.org/10.18610/JG3.2016.071604

[4] I.F. Progri, "Generalized parabolic cylinder function distribution," vol. 2016, article ID 2016071605, 14 pg., Nov. 2016. DOI: https://doi.org/10.18610/JG3.2016.071605

[5] I.F. Progri, "Efficient computation of generalized Bessel function distributions," *J. Geol. Geoinfo. Geointel.*, vol. 2018, article ID 2018071604, 12 pg., Nov. 2018. DOI: https://doi.org/10.18610/JG3.2018.071605

[6] I.F. Progri, "Efficient computation of the special cases of the generalized Bessel function distributions," *J. Geol. Geoinfo. Geointel.,* vol. 2019, article ID 2019071601, 31 pg., Nov. 2019. DOI: https://doi.org/10.18610/JG3.2019.071601

[7] I.F. Progri, "The computation of a 2F2 hypergeometric function," *J. Geol. Geoinfo. Geointel.,* vol. 2020, article ID 2020071601, 12 pg., Nov. 2020. DOI: https://doi.org/10.18610/JG3.2020.071601

[8] I.F. Progri, "Confluent hypergeometric function irregular singularities," *J. Geol. Geoinfo. Geointel.,* vol. 2021, article ID 2021071601,12 pg., Nov. 2021. DOI: https://doi.org/10.18610/JG3.2021.071601

[9] I.F. Progri, "Efficient computation of the generalized parabolic cylinder function distribution," *J. Geol. Geoinfo. Geointel.,* vol. 2021, article ID 2021071602, 17 pg., Nov. 2021. DOI: https://doi.org/10.18610/JG3.2021.071602

[10] I.F. Progri, "Landmark computation of the generalized parabolic cylinder function distribution," *J. Geol. Geoinfo. Geointel.,* vol. 2021, article ID 2021071603, 15 pg., Nov. 2021. DOI: https://doi.org/10.18610/JG3.2021.071603

[11] I.F. Progri, "Special cases of the generalized parabolic cylinder function distribution," *J. Geol. Geoinfo. Geointel.,* vol. 2021, article ID 2021071604, 12 pg., Nov. 2021. DOI: https://doi.org/10.18610/JG3.2021.071604

[12] I.F. Progri, "The significance of Kampé de Fériet functions in the computation of certain generalized distributions," *J. Geol. Geoinfo. Geointel.,* vol. 2021, article ID 2021071605, 7 pg., Nov. 2021. DOI: https://doi.org/10.18610/JG3.2021.071605

[13] I.F. Progri, "Landmark computation of generalized Bessel function distributions of the first kind: part 1," *J. Geol. Geoinfo. Geointel.*, vol. 2022, article ID 2022071601, 13 pg., Nov. 2022. DOI: https://doi.org/10.18610/JG3.2022.071601

[14] I.F. Progri, "A study of the computation of the exponential function," *J. Geol. Geoinfo. Geointel.*, vol. 2022, article ID 2022071602, 10 pg., Nov. 2022. DOI: https://doi.org/10.18610/JG3.2022.071602

[15] I.F. Progri, "Landmark computation of the incomplete gamma function," *J. Geol. Geoinfo. Geointel.*, vol. 2022, article ID 2022071603, 16 pg., Nov. 2022. DOI: https://doi.org/10.18610/JG3.2022.071603

[16] N. Nielsen, *Handbuch der Theorie der Gammafunktion*. Leipzig: Teubner, 326 pg., 1906.

[17] I.S. Gradshteyn, I.M. Ryzhik, (A. Jeffrey, D. Zwillinger, editors) *Table of Integrals, Series, and Products*, 7th ed., Burlington, MA: Academic Press, 1171 pp., 2007.

[18] Anon., "Jack Nicklaus," *Wikipedia, the free encyclopedia*, June 2022. https://en.wikipedia.org/wiki/Jack_Nicklaus

[19] Anon., "Ronald Reagan," *Wikipedia, the free encyclopedia*, June 2022. https://en.wikipedia.org/wiki/Ronald_Reagan

[20] I. Progri, *Indoor Geolocation Systems—Theory and Applications*. *Vol. I*, 1st ed., Worcester, MA: Giftet Inc., ~800 pp., ~ 2022. (not yet available in print)

[21] Anon, “MATLAB 2022b,” *The MathWorks, Inc.*, Natick, MA, Copyright © 1994-2022, The MathWorks, Inc., http://mathworks.com/products/new_products/release2022b.html?requestedDomain=mathworks.com

values of the parameter α.

9 Appendix A: GBFD1K for Integer Values

A modified expression of the GBFD1K for integer values of the parameter can be derived from (1).

From Progri (2022, [15]) the RIGF can be written as in case of integer parameter p:

$$\frac{\gamma(2p_1+2k,dx)}{(2p+2k)!} = 1 - \frac{e^{-dx}(dx)^{2p+2k}\sum_{n=0}^{2p+2k}\binom{2p+2k}{n}n!(dx)^{-n}}{(2p+2k)!} \tag{50}$$

Substituting (51) into (1) yields

$$F_{\text{GBessel1}}(x;a,d,p) = \rho(d,p)\sum_{k=0}^{K}\frac{\gamma(2p_1+2k,dx)(p_1)_k\frac{1}{d^{2k}}}{k!}$$

$$= 1 - \frac{\rho(d,p)\sqrt{\pi}dx\sum_{k=0}^{K}\frac{\frac{x_1^k}{(p_2)_k}}{k!}\sum_{n=0}^{2p+2k}\binom{2p+2k}{n}n!\frac{x^{-n}}{d^n}}{\Gamma(p_1)\Gamma(p_2)e^{dx}}$$

$$= 1 - \frac{(d^2-1)^{p_1}x_1^p\sum_{k=0}^{K}\frac{\frac{x_1^k}{(p_2)_k}}{k!}\sum_{n=0}^{2p+2k}\binom{2p+2k}{n}n!\frac{x^{-n}}{d^n}}{d\left(\frac{1}{2}\right)_p p!e^{dx}} \tag{51}$$

Equation (51) is implemented in Giftet *progril1* MATLAB BIF.

---

[i] See Progri (2016, [1]) (104).

[ii] This bound is a function of the non-integer parameter p. The larger the values of p the larger the bound will be.

[iii] There is a false impression that only the research that is funded by the government bureaucrats is worthy of financial support, awards, & recognition. This is not to say that President Ronald Reagan wanted to demonize the role of the government as an enterprise, or resource, or originator of many inventions. But, when the government spending is abused for wasteful spending of the taxpayers trillions of $ dollars, and moreover, when it controls and suppresses innovation and ignores or completely denies funding to innovators and incubators based on political beliefs, then said President Ronald Reagan this type of government is the problem.