GNSTLIB: a new numerical library for the evaluation of mathematical functions

Javier Segura

Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria, Spain

In collaboration with A. Gil, G. Navas-Palencia and N. M. Temme.

Introducing GNSTLIB

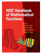
GNSTLIB is a numerical library written in C++11 for **fast** and **accurate** computation of special functions in double precision floating-point arithmetic. GNSTLIB can be used as stand-alone C++ library and provides wrappers for the major programming languages used in scientific computing, such as **Fortran**, **C**, **Python** and other software environments like **Excel**. Vectorized versions of all special functions will be implemented. These routines include an option to trigger OpenMP for parallelization, thus taking advantage of multi-core processors.

Math. team: A. Gil, G. Navas-Palencia, J. Segura, N. M. Temme

Software developer (and initiator of the project): G. Navas-Palencia

Mathematical functions = elementary functions + special functions

Some references for special functions:

1. The NIST  *Handbook of Mathematical Functions (2010)*
2. Books with software: Baker (1992), Moshier (1989), Thompson (1997), Zhang & Jin (1996), Numerical Recipes
3. Our book:  *GST, Numerical Methods for Special Functions, SIAM (2007) (with links to software)*
4. Journals: ACM TOMS, CPC, Applied Statistics
5. Software: CAS (Maple, Mathematica, SAGE), Matlab, NAG, free software...

Package	Axiom	CEPHES	FN	GSL	Kormanyos	MPFR	Maxima	Meta.Numerics	PARI-GP	SLATEC	Sage	mpmath	Baker	Lau	Num. Recipes	Thompson	Watanabe	Zhang	IMSL	Maple	Mathematica	Matlab	NAG	See Also
	Open Source												With Book						Commercial					
Language	Int.	C	Ftn	C	C++	C	Int.	C#	Int. C	Ftn	Int.	Py	C	C Java	C C++ Ftn	C Ftn Mma	Ftn	Ftn	C Ftn Java	Int.	Int.	Int.	C Ftn	
7.25(vii) $\mathscr{F}(z), G(z), z \in \mathbb{C}$																				✓	✓	✓		
8 Incomplete Gamma and Related Functions																								
8.28(ii) $\gamma(a,x), \Gamma(a,x), \gamma^*(a,x), P(a,x), Q(a,x), x, a \in \mathbb{R}$		✓	✓	✓	✓			✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
8.28(iii) $\gamma(a,x), \Gamma(a,x), \gamma^*(a,x), P(a,x), Q(a,x), x, a \in \mathbb{C}$		✓	✓	✓						✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	
8.28(iv) $\mathrm{B}(a,b), I_x(a,b), x, a, b \in \mathbb{R}$								✓																
8.28(v) $\mathrm{B}(a,b), I_z(a,b), z, a, b \in \mathbb{C}$											a	✓									✓			
8.28(vi) $E(x), x \in \mathbb{R}, p \in \mathbb{Z}$		✓		✓				✓		✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓		
8.28(vii) $E(z), z, p \in \mathbb{C}$											✓	✓								✓	✓	✓		

UC UNIVERSIDAD DE CANTABRIA

	Open Source												With Book						Commercial					
Package	Axiom	CEPHES	FN	GSL	Kormanyos	MPFR	Maxima	Meta.Numerics	PARI-GP	SLATEC	Sage	mpmath	Baker	Lau	Num. Recipes	Thompson	Watanabe	Zhang	IMSL	Maple	Mathematica	Matlab	NAG	See Also
Language	*Int.*	C	Ftn	C	C++	C	*Int.*	C#	*Int.* C	Ftn	*Int.*	Py	C	C Java	C C++ Ftn	C Ftn Mma	Ftn	Ftn	C Ftn Java	*Int.*	*Int.*	*Int.*	C Ftn	
14 Legendre and Related Functions																								
14.34(ii) $\mathsf{P}(x)$, $\mathsf{Q}_\nu(x)$, $P_\nu(x)$, $Q_\nu(x)$, $x, \nu \in \mathbb{R}$				✓	✓					✓	*a*	✓	✓		✓	✓		✓		✓	✓	✓	✓	
14.34(iii) $\mathsf{P}(z)$, $\mathsf{Q}_\nu(z)$, $P_\nu(z)$, $Q_\nu(z)$, $z, \nu \in \mathbb{C}$											*a*	✓	✓							✓	✓	*a*		
14.34(iv) $\mathsf{P}(x)$, $\mathsf{Q}_{-\frac{1}{2}+i\tau}(x)$, $\widehat{\mathsf{Q}}^{\mu}_{-\frac{1}{2}+i\tau}(x)$, $P_{-\frac{1}{2}+i\tau}(x)$, $Q_{-\frac{1}{2}+i\tau}(x)$				✓									✓			✓								

Information taken from http://dlmf.nist.gov/software/

What will we offer?

1. Quality control: in most cases we use our own methods and algorithms, which are extensively tested. Most of them have been published in peer-reviewed journals (verified software).
2. We expect to fill a number of gaps in the available libraries and to extend the availability to related topics (statistical distributions and inversion, Gauss quadrature,...)
3. Flexibility: it can be used by programmers in languages such as C/C++, Fortran, Python, or by users of software packages and programming environments like Microsoft Excel.
4. Parallelization will be available.

The plan is to provide software for:

1. Elementary Functions.
2. Gamma Functions.
3. Exponential, Logarithmic and Trigonometric Integrals.
4. Error Functions, Dawson's and Fresnel Integrals.
5. Incomplete Gamma and Generalized Exponential Integral.
6. Airy and Related Functions.
7. Bessel Functions.
8. Parabolic Cylinder Functions.
9. Confluent Hypergeometric Functions.
10. Beta and Incomplete Beta Functions.
11. Orthogonal Polynomials.
12. Gauss Hypergeometric Functions.
13. Zeros of special functions. Gauss quadrature.
14. ...Statistical distributions...

Our principles for computing mathematical functions:

1. The main objective are algorithms which produce reliable double precision values.
2. A given mathematical function is usually a special case of a more general function. Our approach is bottom-up: keep it simple.
3. We accept that it is necessary to combine several methods in order to compute a function accurately and efficiently for a wide range of its variables.
4. We accept that a theoretical error analysis is usually impossible for functions with several real or complex variables. We accept more empirical approaches.
5. The accuracy analysis is usually done by using functional relations, such as a Wronskian relations or by comparing with an alternative method of computation.
6. The selection of methods in different parameter domains is based on speed and accuracy, where the latter may prevail.
7. Possible scaling factors may be considered when available

So far, we have published algorithms and verified software (Fortran) for:

1. Gamma and related functions (2015).
2. Various types of Legendre functions with real parameters: Legendre and associated Legendre functions (1997, 1998), toroidal harmonics (2000), conical functions $P^m_{-1/2+i\tau}(x)$ (2009, 2012).
3. Homogeneous and inhomogeneous Airy functions (2002).
4. Solution of the Bessel equation $x^2y'' + xy + (x^2 + a^2)y = 0$ (2004). Numerically satisfactory pair $\{K_{ia}(x), L_{ia}(x)\}$.
5. Solution of the parabolic cylinder equation $y'' + (a \pm x^2/4)y = 0$ (2006 for $-$, 2011 and 2012 for $+$). Pairs of solutions $\{U(a,x), V(a,x)\}$ ($-$ case) and $\{W(a,x), W(a,-x)\}$
6. Incomplete gamma function ratios $P(a,x)$, $Q(a,x)$ (2012). These are the central gamma distribution functions.
7. Marcum functions $P_\mu(a,x)$, $Q_\mu(a,x)$ (2014). These are the non-central gamma distribution functions.

The current situation:

1. Elementary Functions.
2. Gamma Functions.
3. Exponential, Logarithmic and Trigonometric Integrals.
4. Error Functions, Dawson's and Fresnel Integrals.
5. Incomplete Gamma and Generalized Exponential Integral.
6. Airy and Related Functions.
7. Bessel Functions.
8. Parabolic Cylinder Functions.
9. Confluent Hypergeometric Functions.
10. Beta and Incomplete Beta Functions.
11. Orthogonal Polynomials.
12. Gauss Hypergeometric Functions.
13. Zeros of special functions. Gauss quadrature.
14. ...Statistical distributions...

Blue: already implemented.
Green: waiting for translation to C++.
Orange: Fortran codes available, but not complete yet.

Not only the functions listed before, but also a number of useful related functions are available.
For example, for the **gamma function** $\Gamma(x)$, apart from the computation of the function, we have variants such as:

1. The logarithm of the gamma function.
2. The function $\Gamma^*(x)$ (the regulated gamma function). This function is defined by

$$\Gamma^*(x) = \frac{\Gamma(x)}{\sqrt{2\pi/x}\, x^x e^{-x}}, \quad x > 0.$$

   When x is large, this function can be very important in algorithms where the function $\Gamma(x)$ is involved because $\Gamma^*(x) = 1 + \mathcal{O}(1/x)$.
3. The quotient of two gamma functions. The quotient of two gamma functions appears frequently in applications. From a computational point of view, problems can arise when trying to compute directly the ratio of functions in the case when the arguments of both functions are large.

GNSTLIB::gammastar

double `gammastar`(*const* double x, int &*err_id*)

Computes the scaled gamma function or regulated gamma function denoted as $\Gamma^*(x)$ for $x > 0$. $\Gamma^*(x)$ is defined by

$$\Gamma^*(x) = \frac{\Gamma(x)}{\sqrt{2\pi}x^{x-1/2}e^{-x}}.$$

When x is large, this function can be very important in algorithms where the function $\Gamma(x)$ is involved because $\Gamma^*(x) = 1 + O(1/x)$.

1. **Arguments:**
   - **x**: (double) - *argument*.
     Constraint: $\mathbf{x} > 0.0$.
   - **err_id**: (int) - *error identifier*.
2. **Errors and Warnings:**
   - `err_id` = 1:
     Argument **x** is too close to 0.0. Result is too large to be represented and ∞ is returned.
   - `err_id` = 4:
     Argument $\mathbf{x} \leq 0.0$, NaN is returned.
   - `err_id` = -1:
     Argument **x** is NaN or an unexpected error occurred. Contact the authors.

The GNSTLIB documentation is under construction ▸ GNSTLIB-doc

Release 0.1 is ready for download at Guillermo's website: ▸ GNSTLIB-0.1

Feedback would be greatly appreciated.

THANK YOU!