- Grandparent: 0. Numerical Evaluation of Special Functions[1]
- Parent: 3. Packages, Libraries and Systems[2]
- Previous: 3.3. Comprehensive Libraries[3]
- Next: 4. Functions of One Variable[4]

Press here[5] to get the full document in PostScript format.
Press here[6] to get this subdocument in PostScript format.

NUMERICAL EVALUATION OF SPECIAL FUNCTIONS

D. W. LOZIER AND F. W. J. OLVER

ABSTRACT. This document is an excerpt from the current hypertext version of an article that appeared in Walter Gautschi (ed.), *Mathematics of Computation 1943–1993: A Half-Century of Computational Mathematics*, Proceedings of Symposia in Applied Mathematics 48, American Mathematical Society, Providence, RI 02940, 1994. The symposium was held at the University of British Columbia August 9–13, 1993, in honor of the fiftieth anniversary of the journal *Mathematics of Computation*.

The original abstract follows.

Higher transcendental functions continue to play varied and important roles in investigations by engineers, mathematicians, scientists and statisticians. The purpose of this paper is to assist in locating useful approximations and software for the numerical generation of these functions, and to offer some suggestions for future developments in this field.

3.4. Interactive Systems. The software packages and libraries considered in the preceding three sections are used in conjunction with standard programming languages. These languages are not fully interactive. A program needs to be written, compiled and linked to libraries before it can be executed, and after the results are examined the cycle may need to be repeated to correct errors or change parameters. An *interactive system* provides a powerful set of commands which the user can enter at the keyboard. The response to each command is displayed immediately. The burden of programming and the compile-link-execute cycle is reduced. Programming in an interactive system serves a new purpose: to extend or customize the command set.

A striking characteristic of interactive systems is their ability to integrate non-numerical tasks with numerical computation. Graphical and symbolic computing work best in an interactive environment, and one or both are combined powerfully with numerical computing in commercially available interactive systems. The trend toward increased integration of these computational components is being recognized

[1] /nesf/Sect0/Sect0.html
[2] /nesf/Sect3/Sect3.html
[3] /nesf/Sect3-3/Sect3-3.html
[4] /nesf/Sect4/Sect4.html
[5] ftp://gams.nist.gov/pub/lozier/nesf/Paper.ps.Z
[6] ftp://gams.nist.gov/pub/lozier/nesf/Sect3-4.ps.Z

1991 *Mathematics Subject Classification.* Primary 65D20; Secondary 33-00.
The research of the second author has been supported by NSF Grant CCR 89-14933.

by recent developments of the IMSL, NAG and Scientific Desk libraries (§§3.3.2, 3.3.3 and 3.3.9 above).

A particular type of interactive system with a special capability for the numerical evaluation of special functions is the *computer algebra system*, developed to provide symbolic processing of mathematical formulas and intended, primarily, to assist in mathematical developments. These systems contain basic mathematical information that enables them to manipulate algebraic expressions, make substitutions, differentiate and integrate functions, solve algebraic, transcendental and differential equations, manipulate power series, and the like. Some knowledge of mathematical properties of special functions is built in, and more can be added by programming extensions to the command set. Numerical approximations are to be avoided, in keeping with the primary purpose of supporting exact mathematical developments, but floating-point computation is provided as a secondary capability. This often comes with a bonus when compared to the usual programming languages which simply use the hardware computer arithmetic: the precision can be set arbitrarily.

The rationale for arbitrary precision is not entirely clear. It is clear that exact rational arithmetic is essential in computer algebra applications. Perhaps multiple-precision floating-point, being relatively easy to implement, is considered a worthwhile additional capability. Also, evaluation of symbolic expressions may require high precision because of numerical sensitivity. Whatever the reason, for occasional usage of arbitrary-precision floating-point, computer algebra systems are well worth considering.

3.4.1. *HiQ* (*Apple Macintosh, Sun*). This system [**Bim93**] approaches the goal of reducing the need to write programs by making concentrated use of the graphical user interface (the image displays and controls associated with the computer screen). The system opens with a blank worksheet (in a window on the screen) and an array of icons. Each icon corresponds to a particular kind of task.

For example, one icon is called the "expression evaluator". When activated by the mouse, this icon presents a window with three areas. An algebraic expression is entered into the input area in a conventional programming-language syntax (like Fortran). This expression can contain numbers, symbols representing numbers, and symbols representing built-in HiQ functions. The symbols representing numbers are assigned numerical values in the options area as constants or finite arithmetical sequences. Output icons are generated in the output area when the "run button" (another icon) is "pushed" by clicking the mouse. The output icons, when activated, display tables and graphs of the computed data.

Other tasks that can be performed by similar sequences of manipulations with icons are numerical integration, optimization, data fitting, finding roots of polynomials and nonlinear functions of one variable, and solving nonlinear systems, integral equations, and initial-value and boundary-value problems in ordinary differential equations.

Special functions included in HiQ are Airy functions; beta, gamma, log gamma, psi, incomplete beta, incomplete gamma, and complementary incomplete gamma functions; Kelvin functions; Bessel functions of integer and half-integer order; Struve and Weber parabolic cylinder functions; hypergeometric function and series; confluent hypergeometric function and series; Riemann zeta function. Although HiQ performs complex arithmetic, most if not all the special functions are evaluated for real arguments only.

3.4.2. *Macsyma.* Macsyma [**Sym92**] is a computer algebra system that supports symbolic, graphical and numerical computing on personal computers, scientific workstations and mainframes. Its built-in capabilities can be extended by programming in either Lisp or an Algol-like procedural language.

Macsyma avoids numerical approximations unless floating-point numbers are introduced, either explicitly or as the result of special commands. Floating-point numbers are represented internally in machine single precision, machine double precision, or software arbitrary precision. When arbitrary precision is being used there is a precision specifier. Operand precisions are adjusted, if necessary, to the specified operational precision by truncating or extending with zero digits before arithmetic operations are performed. The precision specifier can be changed at any time.

Macsyma supports the numerical evaluation of elliptic, error, gamma, polygamma, polylogarithmic and zeta functions; Airy, Bessel and Legendre functions; complete elliptic integrals and the exponential integral; classical orthogonal polynomials. Only the gamma, polygamma and Riemann zeta functions are computable in arbitrary precision. Some of the other functions are restricted to single precision. Complex arguments are allowed for the error, gamma and Bessel functions.

3.4.3. ♣ *Maple.* Maple V Release 2 [**CGG+91**] is a computer algebra system containing symbolic, numerical and graphical capabilities. It is available for a wide range of computers, from personal computers and Unix workstations to vector supercomputers. The normal mode of operation is interactive. A Pascal-like programming language, called the Maple language, is provided also. Much of the Maple system is programmed in this language. This part, called the Maple Library, can be viewed on the screen or printed, and serves as useful supplementary documentation or as a guide for the preparation of additional library modules. The core of Maple, written in the C programming language, is not normally accessible to users.

Because of its emphasis on symbolic computing, Maple avoids any evaluation which would introduce an inexact result unless the user specifically requests it. Expressions are evaluated symbolically, with numbers rendered as rational fractions with arbitrarily long numerators and denominators or represented as symbols. The user can request floating-point evaluation to arbitrary precision. For mathematical functions, Maple detects certain special values and can make appropriate substitutions. Otherwise the functions are left as symbolic representations until the user explicitly requests their evaluation in floating-point format. These evaluations, if they can be done at all, are to the precision specified by the user.

3.4.4. *Mathcad (PC with Microsoft Windows).* This system [**Mat93a**] is oriented toward the engineering professions but is useful also in educational, mathematical, scientific and statistical applications. Mathcad can be regarded as an editor and calculator that can be used to create complete documents. These documents can include graphics, ordinary text, and mathematical text resulting from input commands and their associated numerical or symbolic output. Commands are selected from an extensive array of icons (similar to HiQ, §3.4.1 above) or they can be activated by appropriate keystrokes. Numerical commands support real and complex computations with scalars, vectors and matrices; numerical differentiation and integration; solution of algebraic equations; constrained and unconstrained minimization; Fourier transforms; statistical operations. Symbolic commands are supported

by a subset of Maple (§3.4.3 above). Coverage of special functions includes error, gamma and polylogarithmic functions; sine, cosine and Fresnel integrals; Bessel functions of integer order. Except for the gamma function, all arguments must be real.

3.4.5. ♣ *Mathematica.* Mathematica [**Wol91**] is a computer algebra system for symbolic, graphical and numerical computing on personal computers, scientific workstations, and larger computers. A highly developed user interface, available on some of this hardware, integrates Mathematica output with ordinary text for the preparation of complete documents entirely within the Mathematica system. A programming language, based on pattern matching, is included and can be used for extending the capabilities of the system.

As with other computer algebra systems, Mathematica uses floating-point numbers only when requested explicitly. If numbers are introduced with no more significant figures than the precision of the machine floating-point system, and if the machine underflow and overflow limits are not exceeded, then computations proceed in hardware floating-point arithmetic. On the other hand, numbers that are not machine-representable are stored in a software floating-point format. Each such number is tagged with its own precision, and computations are performed in software floating-point arithmetic. The precision of nonmachine numbers is arbitrary but the internal representation is set to the highest justifiable precision. This is determined by the number of significant figures in an input number and by the precision of the operands or arguments in arithmetic operations and function evaluations. If numbers in a hardware computation underflow or overflow, then the software arithmetic takes over automatically.

3.4.6. *Matlab.* This system [**Mat92**] uses matrix notation to provide a built-in set of commands for standard algorithms of numerical computation. A graphics capability is included also. Additional commands can be coded in concise procedures using Matlab notation. Symbolic computing is supported through a recently introduced option using Maple (§3.4.3 above). Matlab runs on a broad range of computers from personal computers and scientific workstations to vector supercomputers. One of its strengths is that it treats complex arithmetic as the natural extension of real arithmetic: variables do not have a fixed real or complex type as in Fortran. Nevertheless, Matlab's coverage of complex functions is limited. It supports Bessel functions of real order and complex argument z but warns in the online help system that the functions "may produce inaccurate results" for large order and $|z|$. Built-in special functions for real arguments and parameters include error and inverse error functions; gamma function; incomplete gamma and beta functions; Bessel functions I, J, K and Y; complete elliptic integrals; Jacobi's elliptic functions.

References

[Bim93] *HiQ reference manual, version 2.0*, Bimillennium Corporation, 16795 Lark Avenue, Suite 200, Los Gatos, California 95030, 1993.

[CGG+91] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monaghan, and S. M. Watt, *Maple V library reference manual*, Springer-Verlag, New York, 1991.

[Mat92] *MATLAB high performance numeric computation and visualization software reference guide*, The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Massachusetts 01760, August 1992, electronic mail address is info@mathworks.com.

[Mat93] *Mathcad 4.0 user's guide*, MathSoft Inc., 201 Broadway, Cambridge, Massachusetts 02139, March 1993.

[Sym92] *MACSYMA reference manual, version 13*, Symbolics, Inc., 20 Academy St., Arlington, Massachusetts 02174-6436, November 1992.

[Wol91] S. Wolfram, *Mathematica, A system for doing mathematics by computer*, second ed., Addison-Wesley, Redwood City, California, 1991.

Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, Md 20899

E-mail address: dlozier@nist.gov

Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742

E-mail address: olver@bessel.umd.edu