2018 HAWAII UNIVERSITY INTERNATIONAL CONFERENCES
STEAM - SCIENCE, TECHNOLOGY & ENGINEERING, ARTS, MATHEMATICS & EDUCATION
JUNE 6 - 8, 2018 PRINCE WAIKIKI, HONOLULU, HAWAII

PRACTICAL APPLICATIONS OF THE INFINITE SERIES FOR CALCULUS STUDENTS

GOECKEL, GREGORY
DEPARTMENT OF MATHEMATICS
PRESBYTERIAN COLLEGE
CLINTON, SOUTH CAROLINA

Practial Applications of the Infinite Series for Calculus Students

Dr. Gregory Goeckel

ABSTRACT: When we teach power series, students are taught how to find the power series' interval of convergence and some ways on how to estimate how many terms are needed to calculate a value, at certain degree of accuracy. What is almost always overlooked is that numerical calculations are not done with infinite precision. Algorithms will be developed for calculating the exponential, logarithmic, and Bessel (first kind, of integer order) functions. Infinite series representations, properties, and identities of each of the functions are incorporated in the algorithms, resulting in robust, versatile, and accurate calculations. The Bessel algorithm will use a unique application to the Taylor's Series, centered at a non-zero center, and is applicable for complex arguments.

KEYWORDS: exponential function, logarithmic function, Bessel function, Taylor's series

INTRODUCTION

When we teach power series, students are taught how to find the power series' interval of convergence and some ways on how to estimate how many terms are needed to calculate a value, at certain degree of accuracy. What is almost always overlooked is that numerical calculations are not done with infinite precision. Loss of precision is also not mentioned. Two ways this enters is when a large number of calculations are performed and when numbers of near equal values are subtracted. So, if the series is not always positive and the result has a magnitude smaller than terms with larger magnitude, then one may assume that there is a loss of significant digits.

Algorithms will be developed for calculating the exponential, logarithmic, and Bessel (first kind, of integer order) functions. Infinite series representations, properties, and identities of each of the functions are incorporated in the algorithms, resulting in robust, versatile, and accurate calculations. The Bessel algorithm will use a unique application to the Taylor's Series, centered at a non-zero center, and is applicable for complex arguments.

NATURAL LOGARITHM

For the natural logarithm, $\ln(x)$, we have the following power series for the natural logarithm:

$$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} \tag{1}$$

where $|x| < 1$.

There are two problems in using this power series:

1. The series is only valid for $-1 < x < 1$.
2. For values near -1 and 1, a great many of terms would be needed.

First we turn our attention to approximating $\ln(2)$. Since

$$\ln(2) = \ln\left(0.5^{-1}\right) = -\ln(0.5) = -\ln(1-0.5),$$

we may use (1), with $x = 0.5$. The series will have all positive terms, double precision convergence within 20 terms, and quadruple precision convergence within 45 terms.

Suppose that $x > 0$.

1. Write $x = b * 2^m$, where $1 \le b < 2$ and m is an integer.
2. If $1 \le b \le \frac{4}{3}$, then define $a = b$ and $n = m$.
   Else define $a = \frac{b}{2}$ and $n = m + 1$.
   Thus we have $x = a * 2^n$, where $\dfrac{2}{3} < a \le \dfrac{4}{3}$ and n is an integer.
3. Set $z = 1 - a$, then $a = 1 - z$, $-\dfrac{1}{3} \le z < \dfrac{1}{3}$ and
   $$\ln(x) = \ln(a) + n\ln(2) = \ln(1-z) + n\ln(2).$$
4. Use (1) to approximate $\ln(x)$.
   Since $|z| < \frac{1}{3}$, the series will quickly converge.

RESULTS: NATURAL LOGARITHM
Comparing the double precision calculations of the above algorithm and the natural logarithm of the C++ compiler, using x ranging through the integers 2 throught 2310000, the average number of agreed digits was 15.94 (0.38% relative error). Comparing the quadruple precision calculations of the above algorithm and the natural logarithm of the C++ compiler, with quadmath, using x ranging through the integers 2 throught 2310000, the average number of agreed digits was 33.67 (0.97% relative error).

EXPONENTIAL FUNCTION
For the exponential function, e^x, the exponential function's power series is

$$e^x \quad = \quad \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{2}$$

where x is any real number.

When $|x| < 1$, (2) will quickly converge. However, for larger magnitudes of x, convergence may take a great many terms, introducing the possibility of roundoff errors.

Recall that for real numbers a and b,

$$e^{ab} \quad = \quad (e^a)^b \tag{3}$$

Suppose that $x > 1$.

1. Write $x = a * 2^n$, where $0.5 \leq a < 1$ and n is a whole number.
2. To approximate e^x, first use (2) to approximate e^a.
3. Repeatedly square this result n times.

For $x < -1$, use the above scheme to approximate e^{-x}, then take it's reciprocal.

RESULTS: EXPONENTIAL
Comparing the double precision calculations of the above algorithm and the exponential function of the C++ compiler, using x ranging from 1 to 709, step size of 0.001, the average number of agreed digits was 12.46 (22.13% relative error). Comparing the quadruple precision calculations of the above algorithm and the the exponential function of the C++ compiler, with quadmath, using x ranging from 1 to 11356, step size of 0.02, the average number of agreed digits was 29.46 (13.35% relative error).

BESSEL FUNCTION
Bessel functions of the first kind, $J_n(x)$, when using the standard math or complex headers in C++, are only defined for real numbers. The need to evaluate $J_n(x)$, in C++, at complex values, has driven me to look for, or develop, a method that would accurately, and efficiently, make these calculations.

The power series for $J_n(z)$, for all complex z, is

$$J_n(z) \quad = \quad \left(\frac{z}{2}\right)^n \sum_{k=0}^{\infty} \left(\frac{z}{2}\right)^{2k} \frac{(-1)^k}{k!(n+k)!} \tag{4}$$

For $|z| \leq 2$, the power series will rapidly converge for all n.

For $|z| > 2$, consider the following ratio:

$$\frac{\left|\frac{(-1)^{k+1}\left(\frac{z}{2}\right)^{2(k+1)}}{(k+1)!((k+1)+n)!}\right|}{\left|\frac{(-1)^k\left(\frac{z}{2}\right)^{2k}}{k!(k+n)!}\right|} \quad = \quad \frac{|z|^2}{4(k+1)(k+1+n)} \tag{5}$$

So, for z, such that $|z| \leq 2\sqrt{n+1}$, the power series for the Bessel function has terms whose magnitude is always decreasing. Thus the convergence of the power series should happen in a small number of terms. Note that this region of values is relatively small; in fact does not include even the first non-zero zero of $J_n(z)$. Thus the power series was only partially helpful.

From here we now turn to the Taylor's series for answers. Since $J_n(z)$ is a smooth function, we may consider its Taylor's series, expanded around a number z_0:

$$J_n(z) = \sum_{k=0}^{\infty} \frac{J_n^{(k)}(z_0)(z-z_0)^k}{k!} \tag{6}$$

Now there is the big issue of finding the general derivative of $J_n(z)$. For this, we have the following identities:

$$J_n'(z) = J_{n-1}(z) - \frac{n}{z} J_n(z) \tag{7}$$

$$J_{n-1}'(z) = -J_n(z) + \frac{n-1}{z} J_{n-1}(z) \tag{8}$$

Using (7) and (8), we may recursively write all the derivatives of $J_n(z)$ in terms of $J_n(z)$ and $J_{n-1}(z)$.

Notice that, in general,

$$\begin{aligned} \frac{d}{dz}\left(\left(a_0 + \sum_{k=1}^{q} a_k z^{-k}\right) J_{n-1}(z) + \left(b_0 + \sum_{k=1}^{q} b_k z^{-k}\right) J_n(z)\right) &= \\ \left(b_0 + \sum_{k=1}^{q} \left((n-k)a_{k-1} + b_k\right) z^{-k} + (n-q-1)a_q z^{-q-1}\right) J_{n-1}(z) &- \\ \left(a_0 + \sum_{k=1}^{q} \left(a_k + (k+n-1)b_{k-1}\right) z^{-k} + (q+n)b_q z^{-q-1}\right) J_n(z) & \end{aligned} \tag{9}$$

Thus we can recursively write the m^{th} derivative of $J_n(z)$ as:

$$\left(a_{m,0} + \sum_{k=1}^{m} a_{m,k} z^{-k}\right) J_{n-1}(z) + \left(b_{m,0} + \sum_{k=1}^{m} b_{m,k} z^{-k}\right) J_n(z) \tag{10}$$

where

$a_{0,0} = 0$, $b_{0,0} = 1$

$a_{m,0} = b_{m-1,0}$, $b_{m,0} = -a_{m-1,0}$

$a_{m,m} = (n-m)a_{m-1,m-1}$, $b_{m,m} = -(m+n-1)b_{m-1,m-1}$

For $1 \leq k < m$

$$a_{m,k} = (n-k)a_{m-1,k-1} + b_{m-1,k}$$
$$b_{m,k} = -a_{m-1,k} - (k+n-1)b_{m-1,k-1}$$

In the same manner, we can recursively write the m^{th} derivative of $J_{n-1}(z)$ as:

$$\left(a_{m,0} + \sum_{k=1}^{m} a_{m,k} z^{-k}\right) J_{n-1}(z) + \left(b_{m,0} + \sum_{k=1}^{m} b_{m,k} z^{-k}\right) J_n(z) \tag{11}$$

where $a_{0,0} = 1$, $b_{0,0} = 0$, and all other coefficients are defined as in (10).

Since $J_n(-z) = (-1)^n J_n(z)$ and $J_n(\overline{z}) = \overline{J_n(z)}$, for all complex z, we need only find an algorithm to calculate $J_n(z)$, where z has non-negative real and imaginary parts.

1. If $|z| \leq 2\sqrt{n+1}$, then use (4) to calculate $J_n(z)$.
2. Else
   (a) If $\text{Re}(z) < 2\sqrt{n+1}$, set $z_0 = \text{Re}(z) + i\sqrt{4(n+1) - (\text{Re}(z))^2}$ and use (4) to calculate $J_{n-1}(z_0)$ and $J_n(z_0)$.
   (b) Else set $z_0 = \text{Re}(z)$ and use built-in real-valued function to calculate $J_{n-1}(z_0)$ and $J_n(z_0)$.
   (c) While $\text{Im}(z - z_0) > 1$
      i. Set $z_1 = z_0 + i$
      ii. Use (6) to calculate $J_{n-1}(z_1)$ and $J_n(z_1)$, where each derivative is calculated using $J_{n-1}(z_0)$ and $J_n(z_0)$ in (10) and (11).
      iii. Set $z_0 = z_1$
   (d) Use (6) to calculate $J_n(z_1)$, where each derivative is calculated using $J_{n-1}(z_0)$ and $J_n(z_0)$ in (10) and (11).

RESULTS: COMPLEX BESSEL CALCULATIONS

To check for accuracy of the above algorithm, I observed how accurate the algorithm preserved the identity:

$$J_{n+2}(z) = \frac{2(n+1)}{z} J_{n+1}(z) - J_n(z)$$

100 complex numbers were randomly chosen with the real part from $(0, 300)$ and the imaginary part from $(0, 10)$. The identity was checked for each $0 \leq n \leq 198$. Using double precision, the average number of agreed digits was 14.58 (8.88% relative error). Using quadruple precision, the average number of agreed digits was 32.44 (4.59% relative error).

100 complex numbers were randomly chosen with the real part from $(0, 300)$ and the imaginary part from $(10, 300)$. The identity was checked for each $0 \leq n \leq 198$. Using double precision, the average number of agreed digits was 14.19 (11.31% relative error). Using quadruple precision, the average number of agreed digits was 32.13 (5.5% relative error).

CONCLUSION

The natural logarithm algorithm perform much better thand the exponential algorithm, when compared to the established calculations. The Bessel function algorithm, especially with quadruple precision, did an acceptible job is preserving the given identity.

References

[1] Milton Abramowitz and Irene A. Stegun *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables* National Bureau of Standards Applied Mathematics Series 55 (1965).

[2] P. BarateUa, M. Garetto, G. Vinardi, Approximation of the Bessel function $J_\nu(x)$ by numerical integration, *Journal of Computational and Applied Mathematics*, volume 7, no. 2, (1981).

[3] J. Birrell, A Recursive Method for Computing Certain Bessel Function Integrals, arXiv:1509.06308 [math.CA].

[4] D. Borwein, J. M. Borwein, and O. Chan, The Evaluation of Bessel functions via exp-arc integrals, September 22, 2007.

[5] J. Harrison, Fast and Accurate Bessel Function Computation, *Proceedings of ARITH19, the 19th IEEE Conference on Computer Arithmetic*, IEEE Computer Society Press, pp. 104-113 (2009).

[6] U. D. Jentschura and E. Ltstedt, Numerical calculation of Bessel, Hankel and Airy functions, *Comput.Phys.Commun.*, pp. 506-519, 183 (2011).

[7] http://keisan.casio.com/calculator

[8] http://functions.wolfram.com/webMathematica/FunctionPlotting.jsp?name=BesselJ

Dr. Gregory Goeckel
Department of Mathematics
Presbyterian College
Greer, South Carolina.

Practical Applications of the Infinite Series for Calculus Students

Synopsis:

Algorithms will be developed for calculating the exponential, logarithmic, and Bessel functions. Properties and identities of each of the functions are incorporated in the algorithms, resulting in robust, versatile, and accurate calculations. The Bessel algorithm will use a unique application to the Taylor's Series, centered at a non-zero center, and is applicable for complex arguments.