

Numerical Evaluation of the Gauss Hypergeometric Function with the hypergeo Package

by Robin K. S. Hankin

Abstract This paper introduces the **hypergeo** package of R routines for numerical calculation of hypergeometric functions. The package is focussed on efficient and accurate evaluation of the Gauss hypergeometric function over the whole of the complex plane within the constraints of fixed-precision arithmetic. The hypergeometric series is convergent only within the unit circle, so analytic continuation must be used to define the function outside the unit circle. This short document outlines the numerical and conceptual methods used in the package; and justifies the package philosophy, which is to maintain transparent and verifiable links between the software and Abramowitz and Stegun (1965). Most of the package functionality is accessed via the single function `hypergeo()`, which dispatches to one of several methods depending on the value of its arguments. The package is demonstrated in the context of game theory.

Introduction

The *geometric* series $\sum_{k=0}^{\infty} t_k$ with $t_k = z^k$ may be characterized by its first term and the constant ratio of successive terms $t_{k+1}/t_k = z$, giving the familiar identity $\sum_{k=0}^{\infty} z^k = (1-z)^{-1}$. Observe that while the series has unit radius of convergence, the right hand side is defined over the whole complex plane except for $z = 1$ where it has a pole. Series of this type may be generalized to a *hypergeometric* series in which the ratio of successive terms is a rational function of k :

$$\frac{t_{k+1}}{t_k} = \frac{P(k)}{Q(k)}$$

where $P(k)$ and $Q(k)$ are polynomials. If both numerator and denominator have been completely factored we would write

$$\frac{t_{k+1}}{t_k} = \frac{(k+a_1)(k+a_2)\cdots(k+a_p)}{(k+b_1)(k+b_2)\cdots(k+b_q)(k+1)} z$$

where z is the ratio of the leading terms of $P(k)$ and $Q(k)$ (the final term in the denominator is due to historical reasons), and if we require $t_0 = 1$ then we write

$$\sum_{k=0}^{\infty} t_k z^k = {}_pF_q \left[\begin{matrix} a_1, a_2, \dots, a_p \\ b_1, b_2, \dots, b_q \end{matrix}; z \right] \quad (1)$$

where it is understood that $q \geq p - 1$. The series representation, namely

$$1 + \frac{\prod_{i=1}^p a_i}{\prod_{i=1}^q b_i} z + \frac{\prod_{i=1}^p a_i (a_i + 1)}{\prod_{i=1}^q b_i (b_i + 1) 2!} z^2 + \cdots + \frac{\prod_{i=1}^p a_i (a_i + 1) \cdots (a_i + k)}{\prod_{i=1}^q b_i (b_i + 1) \cdots (b_i + k) k!} z^k + \cdots \quad (2)$$

is implemented in the package as `genhypergeo_series()` and operates by repeatedly incrementing the upper and lower index vectors (a_1, \dots, a_p) and (b_1, \dots, b_q) , and taking an appropriate running product. Terms are calculated and summed successively until a new term does not change the sum.

In most cases of practical interest one finds that $p = 2, q = 1$ suffices (Seaborn, 1991). Writing a, b, c for the two upper and one lower argument respectively, the resulting function ${}_2F_1(a, b; c; z)$ is known as *the* hypergeometric function, or Gauss's hypergeometric function. Many functions of elementary analysis are of this form; examples would include logarithmic and trigonometric functions, Bessel functions, etc. For example, ${}_2F_1\left(\frac{1}{2}, 1; \frac{3}{2}; -z^2\right) = z^{-1} \arctan z$.

Michel and Stoitsov (2008) state that physical applications are "plethora"; examples would include atomic collisions (Alder et al., 1956), cosmology (de la Cruz-Dombriz and Dobado, 2006), and analysis of Feynman diagrams (Davydychev and Kalmykov, 2004). In addition, naturally-occurring combinatorial series frequently have a sum expressible in terms of hypergeometric functions (Petkovšek et al., 1997). One meets higher-order hypergeometric functions occasionally; the hypergeometric distribution, for example, has a cumulative distribution function involving the ${}_3F_2$ generalized hypergeometric

function. An example from the author's work in the field of game theory is given below.

Numerical implementations

There are two other numerical implementations for the hypergeometric function for R: the `gsl` package (Hankin, 2006b), a wrapper for the Gnu Scientific Library, although this does not cover complex values (Galassi et al., 2013); and the `appell` package (Bove et al., 2013) which implements the Gauss hypergeometric function as `hyp2f1()`.

Outside the R world, there are several proprietary implementations but the evaluation methodology is not available for inspection. Open-source implementations include that of Sage (Stein et al., 2015) and Maxima (2014). The `hypergeo` package is offered as an R-centric suite of functionality with an emphasis on multiple evaluation methodologies, and transparent coding with nomenclature and structure following that of Abramowitz and Stegun (1965). An example is given below in which the positions of the cut lines may be modified.

Equivalent forms

The hypergeometric function's series representation, namely

$${}_2F_1(a, b; c; z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k k!} z^k, \quad (a)_k = \Gamma(a+k)/\Gamma(a) \quad (15.1.1)$$

has unit radius of convergence by the ratio test [NB: equations with three-part numbers, as 15.1.1 above, are named for their reference in Abramowitz and Stegun (1965)]. However, the integral form

$${}_2F_1(a, b; c; z) = \frac{\Gamma(c)}{\Gamma(b)\Gamma(c-b)} \int_{t=0}^1 t^{b-1} (1-t)^{c-b-1} (1-tz)^{-a} dt, \quad (15.3.1)$$

due to Gauss, furnishes analytic continuation; it is usual to follow Riemann and define a cut along the positive real axis from 1 to ∞ and specify continuity from below (but see below). This is implemented as `f15.3.1()` in the package and exhibits surprisingly accurate evaluation.

Gauss also provided a continued fraction form for the hypergeometric function (implemented as `hypergeo_contfrac()` in the package) which has superior convergence rates for parts of the complex plane at the expense of more complicated convergence properties (Cuyt et al., 2008).

The hypergeo package

The `hypergeo` package provides some functionality for the hypergeometric function. The emphasis is on fast vectorized R-centric code, complex z and moderate real values for the auxiliary parameters a, b, c . Extension to complex auxiliary parameters might be possible but Michel and Stoitsov (2008) caution that this is not straightforward. The package is released under GPL-2.

The majority of the package functionality is accessed via the `hypergeo()` function whose behaviour is discussed below.

Observing the slow convergence of the series representation 15.1.1, the complex behaviour of the continued fraction representation, and the heavy computational expense of the integral representation 15.3.1, it is clear that non-trivial numerical techniques are required for a production package.

The package implements a generalization of the method of Forrey (1997) to the complex case. It utilizes the observation that the ratio of successive terms approaches z , and thus the strategy adopted is to seek a transformation which reduces the modulus of z to a minimum. Abramowitz and Stegun give the following transformations:

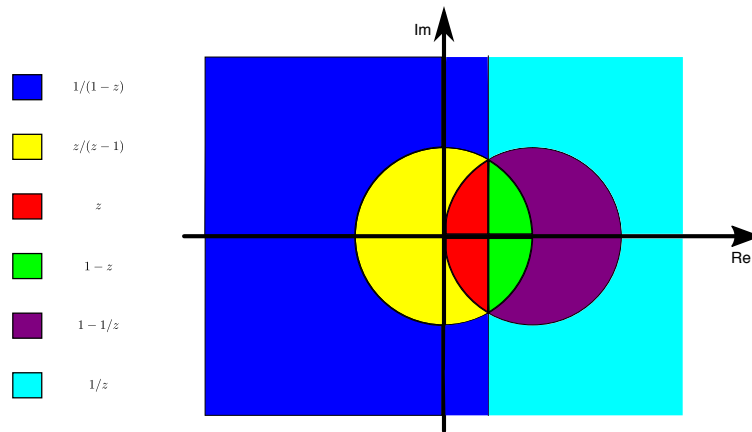


Figure 1: View of the complex plane showing which of equations 15.3.4 to 15.3.9 transforms to the value of smallest modulus. The yellow, green, and red region is the unit circle.

$${}_2F_1(a, b; c; z) = (1 - z)^{-a} {}_2F_1\left(a, c - b; c; \frac{z}{z - 1}\right) \tag{15.3.4}$$

$$= (1 - z)^{-b} {}_2F_1\left(a, c - a; c; \frac{z}{z - 1}\right) \tag{15.3.5}$$

$$= \frac{\Gamma(c) \Gamma(c - a - b)}{\Gamma(c - a) \Gamma(c - b)} {}_2F_1(a, b; a + b - c + 1; 1 - z) + (1 - z)^{c - a - b} \frac{\Gamma(c) \Gamma(a + b - c)}{\Gamma(a) \Gamma(b)} {}_2F_1(c - a, c - b; c - a - b + 1; 1 - z) \tag{15.3.6}$$

$$= \frac{\Gamma(c) \Gamma(b - a)}{\Gamma(b) \Gamma(c - a)} (-z)^{-a} {}_2F_1\left(a, 1 - c + a; 1 - b + a; \frac{1}{z}\right) + \frac{\Gamma(c) \Gamma(a - b)}{\Gamma(a) \Gamma(c - b)} (-z)^{-b} {}_2F_1\left(b, 1 - c + b; 1 - a + b; \frac{1}{z}\right) \tag{15.3.7}$$

$$= (1 - z)^{-a} \frac{\Gamma(c) \Gamma(b - a)}{\Gamma(b) \Gamma(c - a)} {}_2F_1\left(a, c - b; a - b + 1; \frac{1}{1 - z}\right) + (1 - z)^{-b} \frac{\Gamma(c) \Gamma(a - b)}{\Gamma(a) \Gamma(c - b)} {}_2F_1\left(b, c - a; b - a + 1; \frac{1}{1 - z}\right) \tag{15.3.8}$$

$$= \frac{\Gamma(c) \Gamma(c - a - b)}{\Gamma(c - a) \Gamma(c - b)} z^{-a} {}_2F_1\left(a, a - c + 1; a + b - c + 1; 1 - \frac{1}{z}\right) + \frac{\Gamma(c) \Gamma(a + b - c)}{\Gamma(a) \Gamma(b)} (1 - z)^{c - a - b} z^{a - c} {}_2F_1\left(c - a, 1 - a; c - a - b + 1; 1 - \frac{1}{z}\right). \tag{15.3.9}$$

The primary argument in equations 15.3.4–15.3.9 is a member of the set

$$M = \left\{ z, \frac{z}{z - 1}, 1 - z, \frac{1}{z}, \frac{1}{1 - z}, 1 - \frac{1}{z} \right\};$$

and, observing that M is closed under functional composition, we may apply each of the transformations to the primary argument z and choose the one of smallest absolute value to evaluate using `genhypergeo_series()`; see Figure 1 for a diagram showing which parts of the complex plane use which transformation.

Given the appropriate transformation, the right hand side is evaluated using direct summation. If $|z| < 1$, the series is convergent by the ratio test, but may require a large number of terms to achieve acceptable numerical precision. Summation is dispatched to `genhypergeo_series()` which evaluates the generalized hypergeometric function, Equation 1; the R implementation uses multiplication by repeatedly incremented upper and lower indices a_i, b_i .

Thus for example if $(1 - z)^{-1}$ is small in absolute value we would use function `f15.3.8()`:

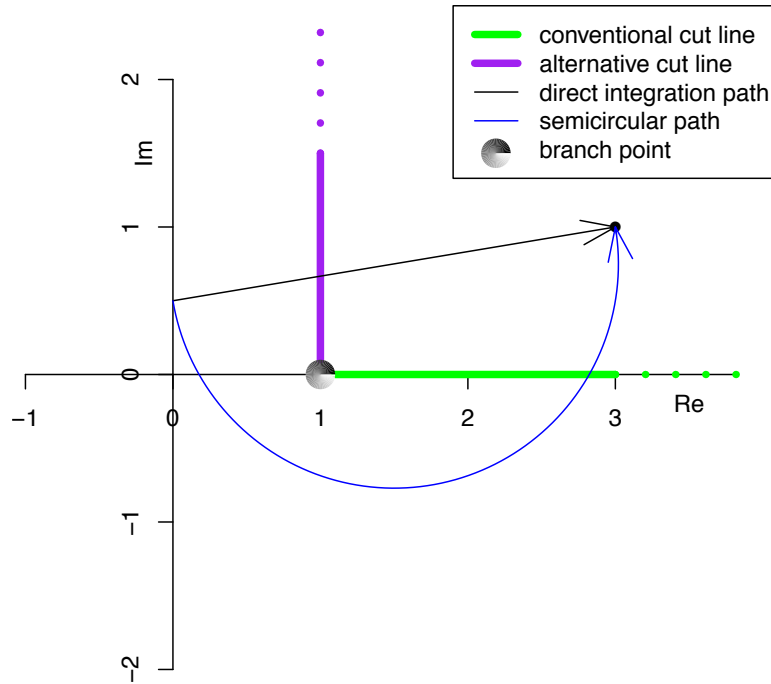


Figure 2: Different integration paths for evaluating ${}_2F_1(3+i)$ from a start point of $i/2$. The straight line path avoids the conventional cut line (green), unlike the semicircular path, which would be consistent with the alternative cut line (purple). The values at $z = 3 + i$ differ because of the residue at $z = 1$.

```
> require("hypergeo")
> f15.3.8

function(A, B, C, z, tol = 0, maxiter = 2000) {
  jj <- i15.3.8(A, B, C)
  jj[1] * (1-z)^(-A) * genhypergeo(U = c(A, C-B), L = A-B+1, z = 1/(1-z), tol = tol,
    maxiter = maxiter) + jj[2] * (1-z)^(-B) * genhypergeo(U = c(B, C-A), L = B-A+1,
    z = 1/(1-z), tol = tol, maxiter = maxiter)
}
```

(slightly edited in the interests of visual clarity). This is a typical internal function of the package and like all similar functions is named for its equation number in Abramowitz and Stegun (1965). Note the helper function `i15.3.9()`, which calculates the Gamma coefficients of the two hypergeometric terms in the identity. This structure allows transparent checking of the code.

Cut lines

The hypergeometric differential equation

$$z(1-z)F''(z) + [c - (a+b+1)z]F'(z) - abF(z) = 0, \tag{15.5.1}$$

together with a known value of $F(z)$ and $F'(z)$ may be used to define ${}_2F_1(z)$. Because $z = 1$ and $z = \infty$ are in general branch points, requiring $F(\cdot)$ to be single valued necessitates a cut line that connects these two points. It is usual to specify a cut line following the real axis from 1 to ∞ ; but sometimes this is inconvenient. Figure 2 shows an example of different integration paths being used to relocate the cut line.

The package includes functionality for solving equation 15.5.1 using `ode()` from the `deSolve` package (Soetaert et al., 2010):

```
> f15.5.1(
+   A = 1.1, B = 2.2, C = 3.5, z = 3+1i, startz = 0.5i,
+   u = function(u) straight(u, 0.5i, 3+1i),
+   udash = function(u) straightdash(u, 0.5i, 3+1i))

[1] -0.5354302+0.7081344i
```

```
> f15.5.1(
+   A = 1.1, B = 2.2, C = 3.5, z = 3+1i, startz = 0.5i,
+   u = function(u) semicircle(u, 0.5i, 3+1i, FALSE),
+   udash = function(u) semidash(u, 0.5i, 3+1i, FALSE))

[1] -1.395698-0.043599i

> hypergeo(1.1, 2.2, 3.5, 3+1i)

[1] -0.5354302+0.7081338i
```

See how the different integration paths give different results; the straight path value matches that of `hypergeo()`. The package also provides `hypergeo_press()`, which is somewhat more user-friendly but less flexible, and uses the method recommended by [Press et al. \(1992\)](#).

Special cases

The series methods detailed above are not applicable for all values of the parameters a, b, c . If, for example, $c = a + b \pm m, m \in \mathbb{N}$ (a not uncommon case), then equation 15.3.6 is not useful because each term has a pole; and it is numerically difficult to approach the limit. In this case the package dispatches to `hypergeo_cover1()` which uses 15.3.4 through 15.3.9 but with 15.3.6 replaced with suitable limiting forms such as

$${}_2F_1(a, b; a + b; z) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \sum_{n=0}^{\infty} \frac{(a)_n(b)_n}{(n!)^2} [2\psi(n + 1) - \psi(a + n) - \psi(b + n) - \log(1 - z)] (1 - z)^n,$$

$$\pi < |\arg(1 - z)| < \pi, |1 - z| < 1 \tag{15.3.10}$$

This equation is comparable to 15.3.6 in terms of computational complexity but requires evaluation of the digamma function ψ . Equation 15.3.10 is evaluated in the package using an algorithm similar to that for `genhypergeo_series()` but includes a runtime option which specifies whether to evaluate $\psi(\cdot)$ *ab initio* each time it is needed, or to use the recurrence relation $\psi(z + 1) = \psi(z) + 1/z$ at each iteration after the first. These two options appear to be comparable in terms of both numerical accuracy and speed of execution, but further work would be needed to specify which is preferable in this context.

A similar methodology is used for the case $b = a \pm m, m = 0, 1, 2, \dots$ in which case the package dispatches to `hypergeo_cover2()`.

However, the case $c - a = 0, 1, 2, \dots$ is not covered by [Abramowitz and Stegun \(1965\)](#) and the package dispatches to `hypergeo_cover3()` which uses formulae taken from the Wolfram functions site ([Wolfram, 2014](#)). For example `w07.23.06.0026.01()` gives a straightforwardly implementable numerical expression for ${}_2F_1$ as a sum of two *finite* series and a generalized hypergeometric function ${}_3F_2$ with primary argument z^{-1} .

In all these cases, the limiting behaviour is problematic. For example, consider a case where $|1 - z| \ll 1$ and $a + b - c$ is close to, but not exactly equal to, zero. Then equation 15.3.10 is not applicable. The analytic value of the hypergeometric function in these circumstances is typically of moderate modulus, but both terms of equation 15.3.6 have large modulus and the numerics are susceptible to cancellation errors. However, in practice this issue seems to be rare as it arises only in contrived situations where one is deliberately testing the system. If a user really was interested in exploring this part of parameter space to high numerical precision then the package provides alternative methodologies such as the integral form `f15.3.1()` or the continued fraction form `genhypergeo_contfrac()`.

Critical points

All the above methods fail when $z = \frac{1}{2} \pm \frac{i\sqrt{3}}{2}$, because none of the transformations 15.3.6–15.3.9 change the modulus of z from 1. The function is convergent at these points but numerical evaluation is difficult. This issue does not arise in the real case considered by [Forrey \(1997\)](#).

These points were considered by [Buhring \(1987\)](#) who presented a computational method for these values; however, his method is not suitable for finite-precision arithmetic (a brief discussion is presented at `?buhring`) and the package employs either `hypergeo_gosper()` which uses iterative scheme due to Gosper ([Johansson et al., 2013](#)), or the residue theorem if z is close to either of these points.

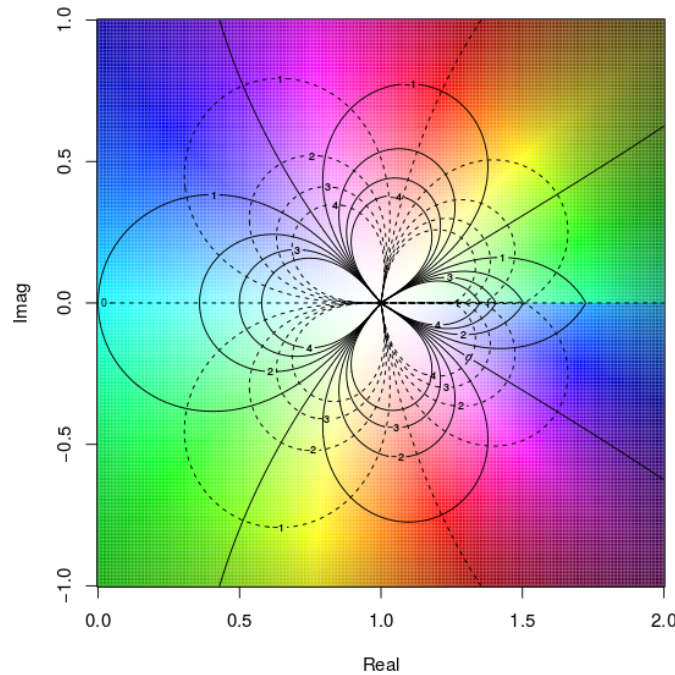


Figure 3: View of the function ${}_2F_1\left(2, \frac{1}{2}; \frac{2}{3}; z\right)$ evaluated over a part of the complex plane using the **hypergeo** package. Function visualization following [Thaller \(1998\)](#) and the **elliptic** package ([Hankin, 2006a](#)); hue corresponds to argument and saturation to modulus. Solid contour lines correspond to real function values and dotted to imaginary function values. Note the cut line along the real axis starting at $(1, 0)$, made visible by an abrupt change in hue.

Package testing suite

The package comes with an extensive test suite in the `tests/` directory. The tests fall into two main categories, firstly comparison with either Maple or Mathematica output following [Becken and Schmelcher \(2000\)](#); and secondly, verification of identities which appear in [Abramowitz and Stegun \(1965\)](#) as elementary special cases. Consider, for example,

$${}_2F_1\left(a, 1 - a; \frac{3}{2}; \sin^2(z)\right) = \frac{\sin[(2a - 1)z]}{(2a - 1)\sin z} \tag{15.1.15}$$

The left and right hand sides are given by `eqn15.1.15a()` and `eqn15.1.15b()` respectively which agree to numerical precision in the test suite; but care must be taken with regard to the placing of branch cuts. Further validation is provided by checking against known analytical results. For example, it is known that

$${}_2F_1\left(2, b; \frac{5 - b}{2}; -\frac{1}{2}\right) = 1 - \frac{b}{3} \tag{3}$$

so, for example,

```
> hypergeo(2, 1, 2, -1/2)
[1] 0.666666666666667+0i
```

The package in use

The **hypergeo** package offers direct numerical functionality to the R user on the command line. The package is designed for use with R and [Figure 3](#) shows the package being used to visualize ${}_2F_1\left(2, \frac{1}{2}; \frac{2}{3}; z\right)$ over a region of the complex plane.

A second example is given from the author’s current work in game theory. Consider a game in which a player is given n counters each of which she must allocate into one of two boxes, A or B .

At times $t = 1, 2, 3 \dots$ a box is identified at random and, if it is not empty, a counter removed from it; box A is chosen with probability p and box B with probability $1 - p$. The object of the game is to remove all counters as quickly as possible. If the player places a counters in box A and b in B , then the probability mass function (PMF) of removing the final counter at time $t = a + b + r$ is

$$p^a(1-p)^b \left[\binom{a+b+r-1}{a-1, b+r} (1-p)^r + \binom{a+b+r-1}{a+r, b-1} p^r \right], \quad r = 0, 1, 2, \dots \quad (4)$$

The two terms correspond to the final counter being removed from box A or B respectively. The PMF for r has expectation

$$p^a(1-p)^b \left[p \binom{a+b}{a+1, b-1} {}_2F_1(a+b+1, 2; a+2; p) + (1-p) \binom{a+b}{a-1, b+1} {}_2F_1(a+b+1, 2; b+2; 1-p) \right] \quad (5)$$

with R idiom:

```
> expected <- function(a, b, p) {
+   Re(
+     choose(a+b, b) * p^a * (1-p)^b *
+     (p * b/(1+a) * hypergeo(a+b+1, 2, a+2, p) +
+     (1-p) * a/(1+b) * hypergeo(a+b+1, 2, b+2, 1-p)))
+ }
```

Thus if $p = 0.8$ and given $n = 10$ counters we might wonder whether it is preferable to allocate them (8,2) or (9,1):

```
> c(expected(8, 2, 0.8), expected(9, 1, 0.8))
```

```
[1] 3.019899 1.921089
```

showing that the latter allocation is preferable in expectation.

Conclusions

Evaluation of the hypergeometric function is hard, as evidenced by the extensive literature concerning its numerical evaluation (Becken and Schmelcher, 2000; Michel and Stoitsov, 2008; Forrey, 1997; Buhring, 1987). The **hypergeo** package is presented as a modular, R-centric implementation with multiple evaluation methodologies, providing reasonably accurate results over the complex plane and covering moderate real values of the auxiliary parameters a, b, c .

Bibliography

- M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1965. [p81, 82, 84, 85, 86]
- K. Alder, A. S. Bohr, T. Huus, B. Mottelson, and A. Winther. Study of nuclear structure by electromagnetic excitation with accelerated ions. *Reviews of Modern Physics*, 28:432, 1956. [p81]
- W. Becken and P. Schmelcher. The analytic continuation of the Gaussian hypergeometric function ${}_2F_1(a, b; c; z)$ for arbitrary parameters. *Journal of Computational and Applied Mathematics*, 126:449–478, 2000. [p86, 87]
- D. S. Bove et al. *appell: Compute Appell's F1 Hypergeometric Function*, 2013. URL <https://CRAN.R-project.org/package=appell>. R package version 0.0-4. [p82]
- W. Buhring. An analytic continuation of the hypergeometric series. *SIAM Journal on Mathematical Analysis*, 18(3):884–889, 1987. [p85, 87]
- A. Cuyt et al. *Handbook of Continued Fractions for Special Functions*. Springer-Verlag, 2008. [p82]
- A. I. Davydychev and M. Y. Kalmykov. Massive Feynman diagrams and inverse binomial sums. *Nuclear Physics B*, 699:3–64, 2004. [p81]

- Á. de la Cruz-Dombriz and A. Dobado. $f(r)$ gravity without a cosmological constant. *Physical Review D*, 74(8–15), Oct. 2006. [p81]
- R. C. Forrey. Computing the hypergeometric function. *Journal of Computational Physics*, 137:79–100, 1997. [p82, 85, 87]
- M. Galassi et al. *GNU Scientific Library*, 2013. URL <http://www.gnu.org/software/gsl/>. Reference Manual edition 1.16. [p82]
- R. K. S. Hankin. Introducing elliptic, an R package for elliptic and modular functions. *Journal of Statistical Software*, 15, Feb. 2006a. [p86]
- R. K. S. Hankin. Special functions in R: Introducing the gsl package. *R News*, 6, Oct. 2006b. [p82]
- F. Johansson et al. *mpmath: A Python Library for Arbitrary-Precision Floating-Point Arithmetic (Version 0.18)*, Oct. 2013. <http://code.google.com/p/mpmath/>. [p85]
- Maxima. *Maxima, a Computer Algebra System. Version 5.34.1*. <http://maxima.sourceforge.net/>, 2014. URL <http://maxima.sourceforge.net/>. [p82]
- N. Michel and M. V. Stoitsov. Fast computation of the Gauss hypergeometric function with all its parameters complex with application to the Pöschl-Teller-Ginocchio potential wave functions. *Computer Physics Communications*, 178:535–551, 2008. URL <http://arxiv.org/abs/0708.0116v1>. [p81, 82, 87]
- M. Petkovšek, H. S. Wilf, and D. Zeilberger. *A = B*. A. K. Peters, Ltd., 1997. [p81]
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: The art of scientific computing*, 1992. [p85]
- J. B. Seaborn. *Hypergeometric Functions and Their Applications*. Springer-Verlag, 1991. [p81]
- K. Soetaert, T. Petzoldt, and R. W. Setzer. Solving differential equations in R: Package desolve. *Journal of Statistical Software*, 33(9):1–25, 2010. URL <http://www.jstatsoft.org/v33/i09>. [p84]
- W. Stein et al. *Sage Mathematics Software (Version 5.8)*. The Sage Development Team, 2015. URL <http://www.sagemath.org>. [p82]
- B. Thaller. Visualization of complex functions. *The Mathematica Journal*, 7(2):163–180, 1998. [p86]
- S. Wolfram. The hypergeometric function, 2014. File *Hypergeometric2F1.pdf*, downloaded from <http://functions.wolfram.com/HypergeometricFunctions/>. [p85]

Robin K. S. Hankin
Auckland University of Technology
New Zealand
hankin.robin@gmail.com