Package 'hypergeo'

February 14, 2012

Title The hypergeometric function

Version 1.2-1

Author Robin K. S. Hankin

Depends R (>= 2.6.0), contfrac, elliptic

Description The hypergeometric function, hypergeo(), for complex numbers

Maintainer Robin K. S. Hankin <hankin.robin@googlemail.com>

Repository CRAN

Date/Publication 2009-09-08 18:14:38

R topics documented:

`hypergeo-package` *The hypergeometric function*

Description

The hypergeometric function for the whole complex plane

Details

Package:	hypergeo
Type:	Package
Version:	1.0
Date:	2008-04-16
License:	GPL

The front end function of the package is `hypergeo()`: depending on the value of the parameters, this executes one or more of many sub-functions.

Author(s)

Robin K. S. Hankin

Maintainer: <hankin.robin@gmail.com>

References

AnS

Examples

```
hypergeo(1.1,2.3,1.9 , 1+6i)

options(showHGcalls = TRUE)    # any non-null value counts as TRUE
hypergeo(4.1, 3.1, 5.1, 1+1i)  # shows trace back
options(showHGcalls = FALSE)   # reset
```

`f15.3.1` *Hypergeometric function using Euler's integral representation*

Description

Hypergeometric function using Euler's integral representation, evaluated using the numerical contour integrals

Usage

```
f15.3.1(A, B, C, z, h = 0)
```

Arguments

A,B,C	Parameters
z	Primary complex argument
h	specification for the path to be taken; see details

Details

Argument h specifies the path to be taken.

If h is real and of length 1, the path taken comprises two straight lines: one from 0 to $0.5 + hi$ and one from $0.5 + hi$ to 1 (if $h = 0$ the integration is performed over a single segment).

Otherwise, the integration is performed over length(h)+1 segments: 0 to h[1], then h[i] to h[i+1] for $1 \leq i \leq n - 1$, and finally h[n] to 1.

See examples and notes sections below.

Note

The Mellin-Barnes form is not yet coded up.

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

hypergeo

Examples

```
f <- function(h){f15.3.1(1,2,3, z=2, h=h)}


# Winding number [around 1/z] matters:
f(0.5)
f(c(1-1i, 1+1i, -2i))

# Accuracy isn't too bad; compare numerical to analytical result :
f(0.5) - (-1+1i*pi/2)
```

f15.3.10 *Transformations of the hypergeometric function*

Description

Transformations of the hypergeometric function detailed in AMS-55, page 559-560.

Usage

```
f15.3.10       (A, B,    z, tol = 0, maxiter = 2000, method = "a")
f15.3.10_a     (A, B,    z, tol = 0, maxiter = 2000              )
f15.3.10_b     (A, B,    z, tol = 0, maxiter = 2000              )
f15.3.11       (A, B, m, z, tol = 0, maxiter = 2000, method = "a")
f15.3.11_bit1  (A, B, m, z, tol = 0                              )
f15.3.11_bit2_a(A, B, m, z, tol = 0, maxiter = 2000              )
f15.3.11_bit2_b(A, B, m, z, tol = 0, maxiter = 2000              )
f15.3.12       (A, B, m, z, tol = 0, maxiter = 2000, method = "a")
f15.3.12_bit1  (A, B, m, z, tol = 0                              )
f15.3.12_bit2_a(A, B, m, z, tol = 0, maxiter = 2000              )
f15.3.12_bit2_b(A, B, m, z, tol = 0, maxiter = 2000              )
f15.3.13       (A, C,    z, tol = 0, maxiter = 2000, method = "a")
f15.3.13_a     (A, C,    z, tol = 0, maxiter = 2000              )
f15.3.13_b     (A, C,    z, tol = 0, maxiter = 2000              )
f15.3.14       (A, C, m, z, tol = 0, maxiter = 2000, method = "a")
f15.3.14_bit1_a(A, C, m, z, tol = 0, maxiter = 2000              )
f15.3.14_bit1_b(A, C, m, z, tol = 0, maxiter = 2000              )
f15.3.14_bit2  (A, C, m, z, tol = 0                              )
f15.3.13_14    (A, C, m, z, tol = 0, maxiter = 2000, method = "a")
f15.3.10_11_12 (A, B, m, z, tol = 0, maxiter = 2000, method = "a")
f15.1.1        (A, B, C, z, tol = 0, maxiter = 2000              )
```

Arguments

`A,B,C`	Parameters of the hypergeometric function
`m`	Integer linking `A`, `B`, `C` as set out in AMS-55, page 559,560
`z`	primary complex argument
`tol,maxiter`	numerical parameters
`method`	Length 1 character vector specifying the method. See details

Details

Naming scheme (functions and arguments) follows AMS-55, pages 559-560.

The `method` argument to (eg) `f15.3.14()` specifies whether to use `psigamma()` directly (method "a"), or the recurrence 6.3.5 (method "b"). Press et al recommend method "b", presumably on the grounds of execution speed. I'm not so sure (method "a" seems to be more accurate in the sense that it returns values closer to those of Maple).

Method "c" means to use a totally dull, slow, direct (but clearly correct) summation, for the purposes of debugging. This is only used for the functions documented under `wolfram.Rd`

Functions `f15.3.13_14()` and `f15.3.10_11_12()` are convenience wrappers. For example, function `f15.3.13_14()` dispatches to either `f15.3.13()` or `f15.3.14()` depending on the value of `m`.

Note

These functions are not really designed to be called by the user: use `hypergeo()` instead, or `hypergeo_cover[123]()` for specific cases.

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

`hypergeo`,`wolfram`,`hypergeo_cover1`

Examples

```
f15.3.10_11_12(A=1.1, B=pi, m= +3, z=.1+.1i)
f15.3.10_11_12(A=1.1, B=pi, m= -3, z=.1+.1i)
```

`f15.3.3` *Various transformation formulae for the hypergeometric function*

Description

Transformations of the hypergeometric function: equations 15.3.3 to 15.3.9

Usage

```
f15.3.3(A, B, C, z, tol = 0, maxiter = 2000)
f15.3.4(A, B, C, z, tol = 0, maxiter = 2000)
f15.3.5(A, B, C, z, tol = 0, maxiter = 2000)
f15.3.6(A, B, C, z, tol = 0, maxiter = 2000)
f15.3.7(A, B, C, z, tol = 0, maxiter = 2000)
f15.3.8(A, B, C, z, tol = 0, maxiter = 2000)
f15.3.9(A, B, C, z, tol = 0, maxiter = 2000)
```

Arguments

`A,B,C`	Parameters of the hypergeometric function
`z`	Primary complex argument
`tol,maxiter`	parameters passed to `genhypergeo()`

Details

The naming scheme follows that of Abramowitz and Stegun

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. "Handbook of mathematical functions". New York: Dover

See Also

`hypergeo`

Examples

```
f15.3.4(1.1,2.2,3.4,-1+0.1i)
```

---

genhypergeo — *The generalized hypergeometric function*

Description

The generalized hypergeometric function, using either the series expansion or the continued fraction expansion.

Usage

```
genhypergeo(U, L, z, tol = 0, maxiter = 2000, check_mod = TRUE, polynomial = FALSE, debug = FALSE, series
genhypergeo_series(U, L, z, tol = 0, maxiter = 2000, check_mod = TRUE,
polynomial = FALSE, debug = FALSE)
genhypergeo_contfrac(U, L, z, tol = 0, maxiter = 2000)
```

Arguments

U,L	Upper and lower arguments respectively (currently real)
z	Primary complex argument (see notes)
tol	tolerance with default zero meaning to iterate until additional terms to not change the partial sum
maxiter	Maximum number of iterations to perform
check_mod	Boolean, with default TRUE meaning to check that the modulus of z is less than 1
polynomial	Boolean, with default FALSE meaning to evaluate the series until converged, or return a warning; and TRUE meaning to return the sum of maxiter terms, whether or not converged. This is useful when either A orB is a nonpositive integer in which case the hypergeometric function is a polynomial
debug	Boolean, with TRUE meaning to return debugging information and default FALSE meaning to return just the evaluate
series	In function genhypergeo(), Boolean argument with default TRUE meaning to return the result of genhypergeo_series() and FALSE the result of genhypergeo_contfrac()

Details

Function genhypergeo() is a wrapper for functions genhypergeo_series() and genhypergeo_contfrac().

Function genhypergeo_series() is the workhorse for the whole package; every call to hypergeo() uses this function except for the (apparently rare—but see the examples section) cases where continued fractions are used.

The generalized hypergeometric function [here genhypergeo()] appears from time to time in the literature (eg Mathematica) as

$$F(U,L;z)=\sum_{n=0}^{\infty}\frac{(u_1)_n(u_2)_n\ldots(u_i)_n}{(l_1)_n(l_2)_n\ldots(l_j)_n}\cdot\frac{z^n}{n!}$$

where $U=(u_1,\ldots,u_i)$ and $L=(l_1,\ldots,l_i)$ are the "upper" and "lower" vectors respectively. The radius of convergence of this formula is 1.

For the Confluent Hypergeometric function, use genhypergeo() with length-1 vectors for arguments U and V.

For the ${}_0F_1$ function (ie no "upper" arguments), use genhypergeo(NULL,L,x).

See documentation for genhypergeo_contfrac() for details of the continued fraction representation.

Note

The radius of convergence for the series is 1 but under some circumstances, analytic continuation defines a function over the whole complex plane (possibly cut along $(0,\infty)$). Further work would be required to implement this.

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

hypergeo,genhypergeo_contfrac

Examples

```
genhypergeo(U=c(1.1,0.2,0.3), L=c(10.1,pi*4), check_mod=FALSE, z=1.12+0.2i)
genhypergeo(U=c(1.1,0.2,0.3), L=c(10.1,pi*4),z=4.12+0.2i,series=FALSE)
```

---

hypergeo *The hypergeometric function*

Description

The Hypergeometric and generalized hypergeometric functions as defined by Abramowitz and Stegun.

Usage

```
hypergeo(A, B, C, z, tol = 0, maxiter=2000)
```

Arguments

A,B,C	Parameters for hypergeo()
z	argument
tol	absolute tolerance; default value of zero means to continue iterating until the result does not change to machine precision; strictly positive values give less accuracy but faster evaluation
maxiter	Integer specifying maximum number of iterations

Details

The hypergeometric function as defined by Abramowitz and Stegun, equation 15.1.1, page 556 is

$$_1F_2(a,b;c;z) = \sum_{n=0}^{\infty} \frac{(a)_n(b)_n}{(c)_n} \cdot \frac{z^n}{n!}$$

where $(a)_n = a(a+1)\ldots(a+n-1) = \Gamma(a+n)/\Gamma(a)$ is the Pochammer symbol (6.1.22, page 256).

Function hypergeo() is the front-end for a rather unwieldy set of back-end functions which are called when the parameters A, B, C take certain values.

The general case (that is, when the parameters do not fall into a “special” category), is handled by `hypergeo_general()`. This applies whichever of the transformations given on page 559 gives the smallest modulus for the argument `z`.

Sometimes `hypergeo_general()` and all the transformations on page 559 fail to converge, in which case `hypergeo()` uses the continued fraction expansion `hypergeo_contfrac()`.

If this fails, the function uses integration via `f15.3.1()`.

Note

Abramowitz and Stegun state:

“The radius of convergence of the Gauss hypergeometric series ... is $|z| = 1$” (AMS-55, section 15.1, page 556).

This reference book gives the correct radius of convergence; use the ratio test to verify it. Thus if $|z| > 1$, the hypergeometric series will diverge and function `genhypergeo()` will fail to converge.

However, the hypergeometric function is meromorphic, so analytic continuation may be used to define a function over the whole complex plane (excepting any poles). Function `hypergeo()` should be used here.

Note that in using these transformations one sometimes draws a “full precision not achieved” warning from `gamma()`; and complex arguments are not allowed. I would suggest either ignoring the warning (the error of `gamma()` is unlikely to be large) or to use one of the bespoke functions such as `f15.3.4()` and tolerate the slower convergence, although this is not always possible.

Author(s)

Robin K. S. Hankin

References

Abramowitz and Stegun 1955. *Handbook of mathematical functions with formulas, graphs and mathematical tables* (AMS-55). National Bureau of Standards

See Also

hypergeo_powerseries, hypergeo_contfrac, genhypergeo

Examples

```
#  equation 15.1.3, page 556:
f1 <- function(x){-log(1-x)/x}
f2 <- function(x){hypergeo(1,1,2,x)}
f3 <- function(x){hypergeo(1,1,2,x,tol=1e-10)}
x <- seq(from = -0.6,to=0.6,len=14)
f1(x)-f2(x)
f1(x)-f3(x)  # Note tighter tolerance

# equation 15.1.7, p556:
g1 <- function(x){log(x + sqrt(1+x^2))/x}
g2 <- function(x){hypergeo(1/2,1/2,3/2,-x^2)}
g1(x)-g2(x)  # should be small
```

```
abs(g1(x+0.1i) - g2(x+0.1i))  # should have small modulus.

# Just a random call, verified by Maple [ Hypergeom([],[1.22],0.9087) ]:
genhypergeo(NULL,1.22,0.9087)


# Little test of vectorization (warning: inefficient):
hypergeo(A=1.2+matrix(1:10,2,5)/10, B=1.4, C=1.665, z=1+2i)
```

`hypergeo_A_nonpos_int` *Hypergeometric functions for integer arguments*

Description

Hypergeometric functions for A and/or B being integers

Usage

```
hypergeo_A_nonpos_int(A, B, C, z, tol = 0)
hypergeo_AorB_nonpos_int(A, B, C, z, tol = 0)
```

Arguments

`A,B,C`	Parameters for the hypergeometric function
`z`	Primary complex argument
`tol`	tolerance

Details

The "point" of these functions is that if A and C (or B and C) are identical nonpositive integers, a warning needs to be given because the function is defined as the appropriate limit and one needs to be sure that both A and C approach that limit at the same speed.

Function `hypergeo_AorB_nonpos_int()` is a convenience wrapper for `hypergeo_A_nonpos_int()`.

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

hypergeo

Examples

```
jjR1 <- hypergeo(-4, pi, 2.2 , 1+6i)
jjR2 <- hypergeo(pi, -4, 2.2 , 1+6i)  # Former bug
jjM <- 3464.1890402837334002-353.94143580568566281i
```

---

hypergeo_contfrac *Continued fraction expansion of the hypergeometric function*

Description

Continued fraction expansion of the hypergeometric and generalized hypergeometric functions using continued fraction expansion.

Usage

```
hypergeo_contfrac(A, B, C, z, tol = 0, maxiter = 2000)
genhypergeo_contfrac_single(U, L, z, tol = 0, maxiter = 2000)
```

Arguments

A,B,C	Parameters (real)
U,L	In function genhypergeo_contfrac(), upper and lower arguments as in genhypergeo()
z	Complex argument
tol	tolerance (passed to GCF())
maxiter	maximum number of iterations

Details

These functions are included in the package in the interests of completeness, but it is not clear when it is advantageous to use continued fraction form rather than the series form.

Note

The function sometimes fails to converge to the correct value but no warning is given.

Function genhypergeo_contfrac() is documented under genhypergeo.Rd.

Author(s)

Robin K. S. Hankin

References

- M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover
- http://functions.wolfram.com/Hypergeometric2F1.pdf

See Also

genhypergeo

Examples

```
hypergeo_contfrac(0.3 , 0.6 , 3.3 , 0.1+0.3i)
# Compare Maple: 1.0042808294775511972+0.17044041575976110947e-1i

genhypergeo_contfrac_single(U=0.2 , L=c(9.9,2.7,8.7) , z=1+10i)
# (powerseries does not converge)
# Compare Maple: 1.0007289707983569879 + 0.86250714217251837317e-2i
```

---

hypergeo_cover1 *Hypergeometric functions for special values of the parameters*

Description

Hypergeometric functions for special values of the parameters

Usage

```
hypergeo_cover1(A, B, m, z, tol = 0, maxiter = 2000, method = "a", give = FALSE)
hypergeo_cover2(A, C, m, z, tol = 0, maxiter = 2000, method = "a", give = FALSE)
hypergeo_cover3(A, n, m, z, tol = 0, maxiter = 2000, method = "a", give = FALSE)
```

Arguments

A,B,C	parameters for the hypergeometric function
m,n	Integers (positive or negative)
z	Primary complex argument
tol,maxiter	Numerical arguments passed to genhypergeo()
method	Method, passed to f15.3.10() (qv)
give	Boolean with TRUE meaning to return the choice of function used and default FALSE meaning to return the function's evaluate

Details

These functions deal with the exceptional cases listed on page 559-560.

- Function `hypergeo_cover1()` deals with the case $C = A + B \pm m, m = 0, 1, 2, \ldots$
- Function `hypergeo_cover2()` deals with the case $B = A \pm m, m = 0, 1, 2, \ldots$
- Function `hypergeo_cover3()` deals with the case $C - A = 0, -1, -2, \ldots$ [elementary] and $C - A = 1, 2, \ldots$ [not covered by AMS-55]

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

hypergeo,f15.3.10,wolfram

Examples

```
# Test hypergeo_cover1():
jjR <- hypergeo(pi,pi/2,3*pi/2-4, z=0.1+0.2i)
jjM <- 0.53745229690249593045 + 1.8917456473240515664i


# Test hypergeo_cover2():
jjM <- -0.15888831928748121465e-5 + 0.40339599711492215912e-4i
jjR <- hypergeo(pi,pi+2, 1.1 , 1+10i)  # This is 15.3.13
stopifnot(Mod(jjR-jjM)<1e-10)


# Test hypergeo_cover3()
jjM <- -0.24397135980533720308e-1 + 0.28819643319432922231i
jjR <- hypergeo(pi, 1.4, pi+4, 1+6i)
stopifnot(Mod(jjR-jjM)<1e-10)
```

---

hypergeo_powerseries *The hypergeometric function as determined by power series*

Description

The hypergeometric function as determined by infinite (`hypergeo_powerseries()`) or finite (`hypergeo_taylor()`) power series

Usage

```
hypergeo_powerseries(A, B, C, z, tol = 0, maxiter = 2000)
```

Arguments

A,B,C	Parameters of the hypergeometric function
z	Primary complex argument
tol,maxiter	Numerical arguments

Details

Function `hypergeo_powerseries()` is the primary decision-making function of the package. It is this function that detects degenerate cases of the three parameters and dispatches accordingly. Non-degenerate cases are sent to function `hypergeo_general()`.

Function `hypergeo_taylor()` deals with cases where the hypergeometric function is a polynomial.

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

hypergeo,genhypergeo

Examples

```
jjR <- hypergeo(pi,-4,2.2,1+5i)
jjM <- 1670.8287595795885335 - 204.81995157365381258i
```

---

i15.3.6 *Helper functions*

Description

Helper functions for equations 15.3.6-15.3.9

Usage

```
i15.3.6(A, B, C)
i15.3.7(A, B, C)
i15.3.8(A, B, C)
i15.3.9(A, B, C)
j15.3.6(A, B, C)
j15.3.7(A, B, C)
j15.3.8(A, B, C)
j15.3.9(A, B, C)
```

Arguments

`A,B,C`	Parameters of the hypergeometric function

Details

Functions `i15.3.?()` return the factors at the beginning of equations 15.3.6-9. These functions return zero if the denominator is infinite (because it includes a gamma function of a nonpositive integer).

Functions `j15.3.?()` check for the appropriate arguments of the gamma function being nonpositive integers.

Author(s)

Robin K. S. Hankin

References

M. Abramowitz and I. A. Stegun 1965. *Handbook of mathematical functions*. New York: Dover

See Also

`hypergeo`

Examples

```
i15.3.6(1.1, 3.2, pi)
```

`is.nonpos` *Various utilities*

Description

Various utilities needing nonce functions

Usage

```
is.near_integer(i,  tol=getOption("tolerance"))
is.nonpos(i)
is.zero(i)
isgood(x, tol)
```

Arguments

i	Numerical vector of suspected integers
tol	Tolerance
x	Argument to isgood()

Details

- Function is.near_integer(i) returns TRUE if i is “near” [that is, within tol] an integer; if the option is unset then 1e-11 is used.
- Function is.nonpos() returns TRUE if i is near a nonpositive integer
- Function is.zero() returns TRUE if i is, er, near zero
- Function isgood() checks for all elements of x having absolute values less than tol

Note

Function isgood() uses zero as the default tolerance (argument tol passed in from hypergeo()); compare the different meaning of tol used in is.near_integer().

Here, “integer” means one of the sequence $0, 1, 2, \ldots$ [ie *not* the Gaussian integers].

Author(s)

Robin K. S. Hankin

Examples

```
is.zero(4)
```

---

wolfram — *Various functions taken from the Wolfram Functions Site*

Description

Various functions taken from the Wolfram Functions Site

Usage

```
w07.23.06.0026.01(A, n, m, z, tol = 0, maxiter = 2000, method = "a")
w07.23.06.0026.01_bit1(A, n, m, z, tol = 0)
w07.23.06.0026.01_bit2(A, n, m, z, tol = 0, maxiter = 2000)
w07.23.06.0026.01_bit3_a(A, n, m, z, tol = 0)
w07.23.06.0026.01_bit3_b(A, n, m, z, tol = 0)
w07.23.06.0026.01_bit3_c(A, n, m, z, tol = 0)
w07.23.06.0029.01(A, n, m, z, tol = 0, maxiter = 2000)
w07.23.06.0031.01(A, n, m, z, tol = 0, maxiter = 2000)
w07.23.06.0031.01_bit1(A, n, m, z, tol = 0, maxiter = 2000)
w07.23.06.0031.01_bit2(A, n, m, z, tol = 0, maxiter = 2000)
```

Arguments

`A`	Parameter of hypergeometric function
`m,n`	Integers
`z`	Primary complex argument
`tol,maxiter`	Numerical arguments as per `genhypergeo()`
`method`	Character, specifying method to be used

Details

The `method` argument is described at `f15.3.10`. All functions' names follow the conventions in `Hypergeometric2F1.pdf`.

- Function `w07.23.06.0026.01(A, n, m, z)` returns ${}_2F_1(A, A+n, A+m, z)$ where m and n are nonnegative integers with $m \geq n$.
- Function `w07.23.06.0029.01(A, n, m, z)` returns ${}_2F_1(A, A+n, A-m, z)$.
- Function `w07.23.06.0031.01(A, n, m, z)` returns ${}_2F_1(A, A+n, A+m, z)$ with $m \leq n$.

Author(s)

Robin K. S. Hankin

References

http://functions.wolfram.com/Hypergeometric2F1.pdf

See Also

`f15.3.10`,`hypergeo`

Examples

```
# Here we catch some answers from Maple (jjM) and compare it with R's:


jjM <- 0.95437201847068289095 + 0.80868687461954479439i # Maple's answer
jjR <- w07.23.06.0026.01(A=1.1 , n=1 , m=4 , z=1+1i)
# [In practice, one would type 'hypergeo(1.1, 2.1, 5.1, 1+1i)']

stopifnot(Mod(jjM - jjR) < 1e-10)



jjM <- -0.25955090546083991160e-3 - 0.59642767921444716242e-3i
jjR <- w07.23.06.0029.01(A=4.1 , n=1 , m=1 , z=1+4i)
# [In practice, one would type 'hypergeo(4.1, 3.1, 5.1, 1+1i)']

stopifnot(Mod(jjM - jjR) < 1e-15)



jjM <- 0.33186808222278923715e-1 - 0.40188208572232037363e-1i
jjR <- w07.23.06.0031.01(6.7,2,1,2+1i)
# [In practice, one would type 'hypergeo(6.7, 8.7, 7.7, 2+1i)']
stopifnot(Mod(jjM - jjR) < 1e-10)
```

Index