# Package 'appell'

February 19, 2015

**Maintainer** Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

**License** GPL (>= 3)

**Title** Compute Appell's F1 hypergeometric function

**LazyLoad** yes

**Author** Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch> with contributions by F. D. Colavecchia, R. C. Forrey, G. Gasaneo, N. L. J. Michel, L. F. Shampine, M. V. Stoitsov and H. A. Watts.

**Description** This package wraps Fortran code by F. D. Colavecchia and G. Gasaneo for computing the Appell's F1 hypergeometric function. Their program uses Fortran code by L. F. Shampine and H. A. Watts. Moreover, the hypergeometric function with complex arguments is computed with Fortran code by N. L. J. Michel and M. V. Stoitsov or with Fortran code by R. C. Forrey. See the function documentations for the references and please cite them accordingly.

**Version** 0.0-4

**Date** 2013-04-16

**Depends** R (>= 2.14.0)

**Collate** 'appellf1.R' 'appell-package.R' 'hyp2f1.R'

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2013-04-16 18:14:09

## R topics documented:

appell-package *Compute Appell's F1 hypergeometric function*

## Description

Compute Appell's F1 hypergeometric function

## Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

appellf1 *Compute Appell's F1 hypergeometric function*

## Description

This function is a wrapper for Fortran code written by F. D. Colavecchia and G. Gasaneo, which is available at http://cpc.cs.qub.ac.uk/summaries/ADSJ. The corresponding background reference is F. D. Colavecchia, G. Gasaneo and J. E. Miraglia (2001): Numerical evaluation of Appell's F1 hypergeometric function, Computer Physics Communications 138:29-43.

## Usage

```
appellf1(a, b1, b2, c, x, y, debug = FALSE,
  userflag = -1L,
  hyp2f1 = c("michel.stoitsov", "forrey"))
```

## Arguments

| | |
|---|---|
| a | complex parameter of Appell's F1 |
| b1 | complex parameter of Appell's F1 |
| b2 | complex parameter of Appell's F1 |
| c | complex parameter of Appell's F1 |
| x | numeric variable |
| y | numeric variable |
| debug | debug mode? (default is FALSE) |
| userflag | user flag variable (not used by default, expert option) |
| hyp2f1 | which algorithm should be used for computing the Gaussian hypergeometric function? See hyp2f1 for details. |

## Details

External code in "rkf45.f90" by L. F. Shampine and H. A. Watts is used which is available from netlib.org at http://www.netlib.org/ode/rkf45.f. It is published in G. E. Forsythe, M. A. Malcolm and C. B. Moler (1977): Computer Methods for Mathematical Computations, Prentice-Hall. Its performance is illustrated in F. Shampine, H. A. Watts and S. Davenport (1976): Solving non-stiff ordinary differential equations - the state of the art, SIAM Review 18:376-411.

The expert user can specify the actual computation with the parameter `userflag`. Here the values 1 and 2 correspond to ODE integration and series summation, respectively, while 0 decides between these two methods based on the parameter values. Other possible values are 15-17 and 21-30, each referring to an equation in F. D. Colavecchia et al. (2001). The default value of `userflag` is -1 and leaves the algorithm decision to the Fortran program, the result of which is returned in the list element `algoflag`. Here the additional values 5 and 6 correspond to simple and polynomial transformations, respectively.

## Value

A list with the algorithm and user flags as well as the complex value of Appell's F1 hypergeometric function.

## Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

## Examples

```
## library(appell)
## The following code compares results with those published in
## Colavecchia et al. (2001), tables 2-5.
## It also illustrates the very minor differences between results
## obtained with the hyp2f1 algorithm by R. Forrey and or that by
## N. Michel and M. Stoitsov, with the exception of no convergence
## problems with the latter. Therefore it is used as default algorithm.

## --------------------
## read the original table 2
table2orig <- read.table(file=
                         system.file(file.path("extdata", "table2.dat"),
                                     package="appell"),
                         col.names=c("x", "y", "absf1", "exact", "relError"))
table2orig

## compute the values here
table2orig <- cbind(table2orig,
                    absf1.forrey=0,
                    absf1.michel.stoitsov=0)

for(case in seq_len(nrow(table2orig)))
{
    table2orig[case, "absf1.forrey"] <-
        abs(appellf1(a=1,
```

```
                    b1=2+1i,
                    b2=1.5-0.5i,
                    c=1,
                    x=table2orig[case, "x"],
                    y=table2orig[case, "y"],
                    debug=TRUE,        # test debugging info as well
                    userflag=1L,
                    hyp2f1="forrey")$val)
    table2orig[case, "absf1.michel.stoitsov"] <-
        abs(appellf1(a=1,
                    b1=2+1i,
                    b2=1.5-0.5i,
                    c=1,
                    x=table2orig[case, "x"],
                    y=table2orig[case, "y"],
                    userflag=1L,
                    hyp2f1="michel.stoitsov")$val)
}
table2orig

## look at the (small) differences:
table2orig$absf1 - table2orig$absf1.forrey
table2orig$absf1 - table2orig$absf1.michel.stoitsov

## here no difference between the hyp2f1 choices:
identical(table2orig$absf1.forrey,
          table2orig$absf1.michel.stoitsov)

## --------------------
## read the original table 3
table3orig <- read.table(file=
                         system.file(file.path("extdata", "table3.dat"),
                                     package="appell"),
                         col.names=
                         c("x", "y", "f1ser", "f1int", "f1exact", "relErrorser",
                           "relErrorint"))

## compute the values here
table3orig <- cbind(table3orig,
                    f1ser.forrey=0,
                    f1int.forrey=0,
                    f1ser.michel.stoitsov=0,
                    f1int.michel.stoitsov=0)

for(case in seq_len(nrow(table3orig)))
{
    ## first everything with Forrey's algorithm for hyp2f1
    f1ser.forrey <- try(appellf1(a=1,
                                 b1=3+1i,
                                 b2=2-0.5i,
                                 c=5+0.5i,
                                 x=table3orig[case, "x"],
                                 y=table3orig[case, "y"],
```

```
                                userflag=2L,
                                debug=TRUE,
                                hyp2f1="forrey"))

    table3orig[case, "f1ser.forrey"] <-
        if(inherits(f1ser.forrey, "try-error"))
            NA
        else
            abs(f1ser.forrey$val)

    table3orig[case, "f1int.forrey"] <- abs(appellf1(a=1,
                                                     b1=3+1i,
                                                     b2=2-0.5i,
                                                     c=5+0.5i,
                                                     x=table3orig[case, "x"],
                                                     y=table3orig[case, "y"],
                                                     userflag=1L,
                                                     debug=TRUE,
                                                     hyp2f1="forrey")$val)

    ## then everything with the algorithm by Michel and Stoitsov for hyp2f1
    f1ser.michel.stoitsov <- try(appellf1(a=1,
                                b1=3+1i,
                                b2=2-0.5i,
                                c=5+0.5i,
                                x=table3orig[case, "x"],
                                y=table3orig[case, "y"],
                                userflag=2L,
                                debug=TRUE,
                                hyp2f1="michel.stoitsov"))

    table3orig[case, "f1ser.michel.stoitsov"] <-
        if(inherits(f1ser.michel.stoitsov, "try-error"))
            NA
        else
            abs(f1ser.michel.stoitsov$val)

    table3orig[case, "f1int.michel.stoitsov"] <- abs(appellf1(a=1,
                                                     b1=3+1i,
                                                     b2=2-0.5i,
                                                     c=5+0.5i,
                                                     x=table3orig[case, "x"],
                                                     y=table3orig[case, "y"],
                                                     userflag=1L,
                                                     debug=TRUE,
                                                     hyp2f1="michel.stoitsov")$val)
}
table3orig

## look at the (small) differences:
table3orig$f1ser - table3orig$f1ser.michel.stoitsov
table3orig$f1int - table3orig$f1int.michel.stoitsov
## so we have no missing values for Michel & Stoitsov
```

```
## besides that, only very small differences between the two methods:
table3orig$f1ser.michel.stoitsov - table3orig$f1ser.forrey
table3orig$f1int.michel.stoitsov - table3orig$f1int.forrey

## --------------------
## read the original table 4
table4orig <- read.table(file=
                         system.file(file.path("extdata", "table4.dat"),
                                     package="appell"),
                         col.names=
                         c("x", "y", "f1", "hypergeo", "exact", "relErrorf1",
                           "relErrorhypergeo"))

## compute the values here
table4orig <- cbind(table4orig,
                    f1.forrey=0,
                    f1.michel.stoitsov=0,
                    flag=0)

for(case in seq_len(nrow(table4orig)))
{
    ## get Forrey result and flag
    thisRes <- appellf1(a=-0.5,
                        b1=2,
                        b2=1,
                        c=3,
                        x=table4orig[case, "x"],
                        y=table4orig[case, "y"],
                        hyp2f1="forrey")

    table4orig[case, "f1.forrey"] <- abs(thisRes$val)
    table4orig[case, "flag"] <- thisRes$algoflag

    ## get Michel & Stoitsov result
    table4orig[case, "f1.michel.stoitsov"] <-
        abs(appellf1(a=-0.5,
                     b1=2,
                     b2=1,
                     c=3,
                     x=table4orig[case, "x"],
                     y=table4orig[case, "y"],
                     hyp2f1="michel.stoitsov")$val)
}
table4orig

## look at the (small) differences:
table4orig$f1 - table4orig$f1.forrey
## very small errors all over the place!

## and extremely small differences between Forrey and Michel & Stoitsov:
table4orig$f1.michel.stoitsov - table4orig$f1.forrey
```

```
## look at the flags
subset(table4orig,
       select=c(x, y, flag))
```

---

hyp2f1 *Compute the Gaussian hypergeometric function with complex arguments*

---

## Description

Two different algorithms can be used.

## Usage

```
hyp2f1(a, b, c, z,
  algorithm = c("michel.stoitsov", "forrey"))
```

## Arguments

| | |
|---|---|
| a | complex parameter |
| b | complex parameter |
| c | complex parameter |
| z | complex variable |
| algorithm | either "michel.stoitsov" (default) or "forrey" (see the details) |

## Details

The first, default, algorithm uses Fortran code in "hyp_2F1.f90" from N. L. J. Michel and M. V. Stoitsov, which is available at http://cpc.cs.qub.ac.uk/summaries/AEAE. The corresponding background reference is N. L. J. Michel and M. V. Stoitsov (2008): 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.

The second algorithm uses Fortran code in "cyp.f" from R. C. Forrey is used which is available at http://physics.bk.psu.edu/codes/chyp.f. The corresponding background reference is R. C. Forrey (1997): Computing the hypergeometric function, Journal of Computational Physics 137:79-100.

## Value

The complex value of the Gaussian hypergeometric function.

## Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

## Examples

```
## library(appell)

## compare the results of both algorithms
## for random test data.

## todo: add better tests trying to replicate published results?

nTest <- 100L
set.seed(38)

a <- complex(real=rnorm(nTest),
             imaginary=rnorm(nTest))
b <- complex(real=rnorm(nTest),
             imaginary=rnorm(nTest))
c <- complex(real=rnorm(nTest),
             imaginary=rnorm(nTest))
z <- complex(real=rnorm(nTest),
             imaginary=rnorm(nTest))

tableHyp2f1 <- matrix(nrow=nTest,
                      ncol=2L,
                      dimnames=
                      list(NULL,
                           c("forrey", "michel.stoitsov")))

for(i in seq_len(nTest))
{
    tableHyp2f1[i, "forrey"] <- hyp2f1(a[i], b[i], c[i], z[i],
                                       algorithm="forrey")
    tableHyp2f1[i, "michel.stoitsov"] <- hyp2f1(a[i], b[i], c[i], z[i],
                                                algorithm="michel.stoitsov")
}

tableHyp2f1

abs(tableHyp2f1[, "forrey"] - tableHyp2f1[, "michel.stoitsov"])
## so very small differences,
## at least in this range of function parameters.
```

# Index