

# The contfrac Package

August 22, 2008

**Title** Continued fractions

**Version** 1.1-5

**Author** Robin K. S. Hankin

**Description** Various utilities for evaluating continued fractions

**Maintainer** Robin K. S. Hankin <rksh1@cam.ac.uk>

**License** GPL version 2

## R topics documented:

CF . . . . .	1
as_cf . . . . .	3
contfrac-package . . . . .	4
convergents . . . . .	5
<b>Index</b>	<b>7</b>

---

CF	<i>Continued fraction convergent</i>
----	--------------------------------------

---

## Description

Returns continued fraction convergent using the modified Lenz's algorithm; function `CF ()` deals with continued fractions and `GCF ()` deals with generalized continued fractions.

## Usage

```
CF(a, finite = FALSE, tol=0)
GCF(a,b, b0=0, finite = FALSE, tol=0)
```

**Arguments**

<code>a, b</code>	In function <code>CF()</code> , the elements of <code>a</code> are the partial denominators; in <code>GCF()</code> the elements of <code>a</code> are the partial numerators and the elements of <code>b</code> the partial denominators
<code>finite</code>	Boolean, with default <code>FALSE</code> meaning to iterate Lenz's algorithm until convergence (a warning is given if the sequence has not converged); and <code>TRUE</code> meaning to evaluate the finite continued fraction
<code>b0</code>	In function <code>GCF()</code> , floor of the continued fraction
<code>tol</code>	tolerance, with default 0 silently replaced with <code>.Machine\$double.eps</code>

**Details**

Function `CF()` treats the first element of its argument as the integer part of the convergent.

Function `CF()` is a wrapper for `GCF()`; it includes special dispensation for infinite values (in which case the value of the appropriate finite CF is returned).

The implementation is in C; the real and complex cases are treated separately in the interests of efficiency.

The algorithm terminates when the convergence criterion is achieved irrespective of the value of `finite`.

**Author(s)**

Robin K. S. Hankin

**References**

- W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling 1992. *Numerical recipes 3rd edition: the art of scientific computing*. Cambridge University Press; section 5.2 "Evaluation of continued fractions"
- W. J. Lenz 1976. Generating Bessel functions in Mie scattering calculations using continued fractions. *Applied Optics*, 15(3):668-671

**See Also**

[convergents](#)

**Examples**

```
phi <- (sqrt(5)+1)/2
phi_cf <- CF(rep(1,100))
phi - phi_cf      # should be small

# The tan function:

"tan_cf" <-
function(z,n=20){GCF(c(z, rep(-z^2,n-1)), seq(from=1,by=2, len=n)) }
```

```
z <- 1+1i
tan(z) - tan_cf(z)  # should be small
```

---

as\_cf

*Converts a real number to continued fraction form*

---

### Description

Converts a real number to continued fraction form using a standard simple algorithm

### Usage

```
as_cf(x, n = 10)
```

### Arguments

x                    real number to be converted to continued fraction form  
n                    Number of partial denominators to evaluate; see Notes

### Note

Has difficulties with rational values as expected

### Author(s)

Robin K. S. Hankin

### See Also

[CF,convergents](#)

### Examples

```
phi <- (sqrt(5)+1)/2
as_cf(phi,50)  # loses it after about 38 iterations ... not bad ...

as_cf(pi)  # looks about right
as_cf(exp(1),20)

f <- function(x){CF(as_cf(x,30),TRUE) - x}

x <- runif(40)
plot(sapply(x,f))
```

---

contfrac-package    *Continued fractions*

---

## Description

Various utilities for manipulating continued fractions

## Details

Package:    contfrac  
Type:        Package  
Version:    1.0  
Date:        2008-04-04  
License:    GPL

## Author(s)

Robin K. S. Hankin

Maintainer: <r.hankin@noc.soton.ac.uk>

## References

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling 1992. *Numerical recipes 3rd edition: the art of scientific computing*. Cambridge University Press; section 5.2 “Evaluation of continued fractions”

## Examples

```
# Some convergents of pi:
jj <- convergents(c(3,7,15,1,292))
jj$A / jj$B - pi

# An identity of Euler's:
jj <- GCF(a=seq(from=2,by=2,len=30), b=seq(from=3,by=2,len=30), b0=1)
jj - 1/(exp(0.5)-1)    # should be small

# Now a continued fraction representation of tan(z):
tan_cf <- function(z,n=14){ GCF(c(z,rep(-z^2,n-1)), seq(from=1,by=2,len=n)) }

tan_cf(1+1i) - tan(1+1i)    # should be small
```

convergents

*Partial convergents of continued fractions***Description**

Partial convergents of continued fractions or generalized continued fractions

**Usage**

```
convergents(a)
gconvergents(a,b, b0 = 0)
```

**Arguments**

`a, b` In function `convergents()`, the elements of `a` are the partial denominators (the first element of `a` is the integer part of the continued fraction). In `gconvergents()` the elements of `a` are the partial numerators and the elements of `b` the partial denominators

`b0` The floor of the fraction

**Details**

Function `convergents()` returns partial convergents of the continued fraction

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \frac{1}{a_5 + \ddots}}}}}$$

where  $a = a_0, a_1, a_2, \dots$  (note the off-by-one issue).

Function `gconvergents()` returns partial convergents of the continued fraction

$$b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{b_4 + \frac{a_5}{b_5 + \ddots}}}}}$$

where  $a = a_1, a_2, \dots$

**Value**

Returns a list of two elements, A for the numerators and B for the denominators

**Note**

This classical algorithm generates very large partial numerators and denominators. To evaluate limits, use functions `CF()` or `GCF()`.

**Author(s)**

Robin K. S. Hankin

**References**

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling 1992. *Numerical recipes 3rd edition: the art of scientific computing*. Cambridge University Press; section 5.2 “Evaluation of continued fractions”

**See Also**

[CF](#)

**Examples**

```
# Successive approximations to pi:  
  
jj <- convergents(c(3,7,15,1,292))  
jj$a/jj$b - pi  
  
convergents(rep(1,10))
```

# Index

\*Topic **math**

as\_cf, 3

CF, 1

convergents, 5

\*Topic **package**

contfrac-package, 4

as\_cf, 3

CF, 1, 3, 6

contfrac (*contfrac-package*), 4

contfrac-package, 4

convergents, 2, 3, 5

GCF (*CF*), 1

gconvergents (*convergents*), 5