

# A **Pari/GP** Package for Continued Fractions

Henri Cohen

November 15, 2022, Séminaire LFANT

LFANT INRIA, IMB, Université de Bordeaux

# Continued Fractions I

A continued fraction associated to two sequences  $a(n)$  and  $b(n)$  with  $b(n) \neq 0$  for all  $n$  is an expression

$$S = a(0) + \frac{b(0)}{a(1) + \frac{b(1)}{a(2) + \frac{b(2)}{a(3) + \ddots}}}$$

assuming the limit of  $p(n)/q(n)$  exists, where  $p(n)/q(n)$  is the fraction obtained by truncating at  $b(n-1)/a(n)$ .

Almost all continued fractions occurring in the literature associated either to **fixed real numbers** or to **functions** are such that  $a(n)$  and  $b(n)$  are rational functions of  $n$  for  $n$  sufficiently large, possibly depending on the parity of  $n$  or even on  $n$  modulo  $N \geq 3$ .

# Continued Fractions I

A continued fraction associated to two sequences  $a(n)$  and  $b(n)$  with  $b(n) \neq 0$  for all  $n$  is an expression

$$S = a(0) + \frac{b(0)}{a(1) + \frac{b(1)}{a(2) + \frac{b(2)}{a(3) + \ddots}}}$$

assuming the limit of  $p(n)/q(n)$  exists, where  $p(n)/q(n)$  is the fraction obtained by truncating at  $b(n-1)/a(n)$ .

Almost all continued fractions occurring in the literature associated either to **fixed real numbers** or to **functions** are such that  $a(n)$  and  $b(n)$  are rational functions of  $n$  for  $n$  sufficiently large, possibly depending on the parity of  $n$  or even on  $n$  modulo  $N \geq 3$ .

# Continued Fractions II

Trivial that if  $r(0) = 1$  and  $r(n) \neq 0$  for all  $n$  is any sequence,  $(a(n)r(n), b(n)r(n)r(n+1))$  give the same **convergents**  $p(n)/q(n)$ , so we may always assume that  $a(n)$  and  $b(n)$  are **polynomials** in  $n$  for  $n$  sufficiently large, again possibly depending on a congruence of  $n$ . Such a continued fraction will be said to be of **polynomial type**, and are the only ones we consider.

Examples with no congruence and congruence modulo 2:

$$e^z = 1 + \frac{2z}{2 - z + \frac{z^2}{6 + \frac{z^2}{10 + \frac{z^2}{14 + \ddots}}}}$$

# Continued Fractions II

Trivial that if  $r(0) = 1$  and  $r(n) \neq 0$  for all  $n$  is any sequence,  $(a(n)r(n), b(n)r(n)r(n+1))$  give the same **convergents**  $p(n)/q(n)$ , so we may always assume that  $a(n)$  and  $b(n)$  are **polynomials** in  $n$  for  $n$  sufficiently large, again possibly depending on a congruence of  $n$ . Such a continued fraction will be said to be of **polynomial type**, and are the only ones we consider.

Examples with no congruence and congruence modulo 2:

$$e^z = 1 + \frac{2z}{2 - z + \frac{z^2}{6 + \frac{z^2}{10 + \frac{z^2}{14 + \ddots}}}}$$

# Continued Fractions III

$$\frac{1+z}{z} \log(1+z) = 1 + \frac{z}{2 + \frac{1.2z}{3 + \frac{1.2z}{4 + \frac{2.3z}{5 + \frac{2.3z}{6 + \frac{3.4z}{7 + \ddots}}}}}}$$

- In the literature, the **speed of convergence** is almost never given, or requires many pages. The package gives it immediately up to a multiplicative constant  $C$ .
- The package computes **numerically** the limit  $S$ , but also (less trivially) the multiplicative constant  $C$ .
- The package can **simplify** in several ways (removing denominators, contraction, etc...) a given continued fraction.

- In the literature, the **speed of convergence** is almost never given, or requires many pages. The package gives it immediately up to a multiplicative constant  $C$ .
- The package computes **numerically** the limit  $S$ , but also (less trivially) the multiplicative constant  $C$ .
- The package can **simplify** in several ways (removing denominators, contraction, etc...) a given continued fraction.



- In the literature, the **speed of convergence** is almost never given, or requires many pages. The package gives it immediately up to a multiplicative constant  $C$ .
- The package computes **numerically** the limit  $S$ , but also (less trivially) the multiplicative constant  $C$ .
- The package can **simplify** in several ways (removing denominators, contraction, etc...) a given continued fraction.

# Goals II

- The package can **accelerate** the convergence of a continued fraction, and thus obtain new ones.
- The most surprising and important fact is that it can even automatically do what I call **Apéry acceleration**, converting a slowly convergent series into an exponentially convergent one, and incredibly enough, this works for **most** slowly convergent series in the literature, and even transforms an exponentially convergent one into one converging exponentially better.
- A final goal is to help create a compendium of the most important continued fractions, together with both their speed of convergence, and whenever possible, their accelerated counterparts.

# Goals II

- The package can **accelerate** the convergence of a continued fraction, and thus obtain new ones.
- The most surprising and important fact is that it can even automatically do what I call **Apéry acceleration**, converting a slowly convergent series into an exponentially convergent one, and incredibly enough, this works for **most** slowly convergent series in the literature, and even transforms an exponentially convergent one into one converging exponentially better.
- A final goal is to help create a compendium of the most important continued fractions, together with both their speed of convergence, and whenever possible, their accelerated counterparts.

# Goals II

- The package can **accelerate** the convergence of a continued fraction, and thus obtain new ones.
- The most surprising and important fact is that it can even automatically do what I call **Apéry acceleration**, converting a slowly convergent series into an exponentially convergent one, and incredibly enough, this works for **most** slowly convergent series in the literature, and even transforms an exponentially convergent one into one converging exponentially better.
- A final goal is to help create a compendium of the most important continued fractions, together with both their speed of convergence, and whenever possible, their accelerated counterparts.

# Speeds of Convergence I

It can be shown that there are five different types of convergence, which can be summarized by three formulas: either

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{n!^k E^n n^P}$$

for some constants  $(C, k, E, P)$  with  $CE \neq 0$ , or

$$S - \frac{p(n)}{q(n)} \sim \varepsilon^n \frac{C}{e^{\sqrt{Dn}}}$$

for some  $\varepsilon = \pm 1$  and constants  $(C, D)$  with  $C \neq 0$  and  $D > 0$ , or (very exceptionally)

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{\log(n)}$$

with  $C \neq 0$ .

# Speeds of Convergence I

It can be shown that there are five different types of convergence, which can be summarized by three formulas: either

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{n!^k E^n n^P}$$

for some constants  $(C, k, E, P)$  with  $CE \neq 0$ , or

$$S - \frac{p(n)}{q(n)} \sim \varepsilon^n \frac{C}{e^{\sqrt{Dn}}}$$

for some  $\varepsilon = \pm 1$  and constants  $(C, D)$  with  $C \neq 0$  and  $D > 0$ , or (very exceptionally)

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{\log(n)}$$

with  $C \neq 0$ .

# Speeds of Convergence I

It can be shown that there are five different types of convergence, which can be summarized by three formulas: either

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{n!^k E^n n^P}$$

for some constants  $(C, k, E, P)$  with  $CE \neq 0$ , or

$$S - \frac{p(n)}{q(n)} \sim \varepsilon^n \frac{C}{e^{\sqrt{Dn}}}$$

for some  $\varepsilon = \pm 1$  and constants  $(C, D)$  with  $C \neq 0$  and  $D > 0$ , or (very exceptionally)

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{\log(n)}$$

with  $C \neq 0$ .

# Speeds of Convergence II

I mention five types because the first one

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{n!^k E^n n^P}$$

is divided into three subtypes: first  $k > 0$  (so that  $n!^k$  dominates), **factorial** convergence, second  $k = 0$  and  $|E| > 1$  (so that  $E^n$  dominates), **exponential** convergence, and third  $k = 0$ ,  $|E| = 1$ , and  $P > 0$ , **polynomial** convergence. The second  $\varepsilon^n C / e^{\sqrt{Dn}}$  and third types  $C / \log(n)$  are called **subexponential** and **logarithmic** convergence respectively.

We will ignore the very rare logarithmic convergence (although it **is** detected by the package), and denote by the four-component vector  $[k, E, D, P]$  the type of convergence.



# Speeds of Convergence II

I mention five types because the first one

$$S - \frac{p(n)}{q(n)} \sim \frac{C}{n!^k E^n n^P}$$

is divided into three subtypes: first  $k > 0$  (so that  $n!^k$  dominates), **factorial** convergence, second  $k = 0$  and  $|E| > 1$  (so that  $E^n$  dominates), **exponential** convergence, and third  $k = 0$ ,  $|E| = 1$ , and  $P > 0$ , **polynomial** convergence. The second  $\varepsilon^n C / e^{\sqrt{Dn}}$  and third types  $C / \log(n)$  are called **subexponential** and **logarithmic** convergence respectively.

We will ignore the very rare logarithmic convergence (although it **is** detected by the package), and denote by the four-component vector  $[k, E, D, P]$  the type of convergence.

# Speeds of Convergence III

Warning concerning limits and asymptotics:

- In case of logarithmic convergence, the package is unable to do anything, I do not know of any algorithm to compute **numerically** the limit and a fortiori the asymptotics.
- In case of polynomial convergence  $P$ , we use extrapolating techniques which work only if  $P$  is a rational number with small denominator  $d$  (in practice  $d \leq 4$ ). If  $d \geq 5$ , the result may either be completely wrong (for instance  $10^{250}$  instead of  $1$ ), or have only very few correct decimals. The value of  $P$  can be checked using the function **cftype**, see below.

# Creating Continued Fractions I

From now on, I will introduce the functions of the package and give a large number of examples. A continued fraction (CF) is a pair  $(a, b)$  of closures in the sense of Pari/GP. For instance the continued fraction for  $e^z$  given above is created by

```
a=(n->if(n==0,1,if(n==1,2-z,4*n-2)));  
b=(n->if(n==0,2*z,z^2));
```

and the continued fraction itself is the 2-component vector  $[a, b]$ . For the convenience of the user, GP also accepts

```
a=[1,2-z,4*n-2];b=[2*z,z^2];
```

and a conversion back and forth to and from closures is done internally using the functions `cfvectoclos` and `cfclostovec`.

Important function `cfsubst`: since we cannot directly replace in a closure, specific function: `ab3=cfsubst([a,b],z,3)` gives the above CF for  $z = 3$ .

# Creating Continued Fractions I

From now on, I will introduce the functions of the package and give a large number of examples. A continued fraction (CF) is a pair  $(a, b)$  of closures in the sense of Pari/GP. For instance the continued fraction for  $e^z$  given above is created by

```
a=(n->if(n==0,1,if(n==1,2-z,4*n-2)));  
b=(n->if(n==0,2*z,z^2));
```

and the continued fraction itself is the 2-component vector  $[a, b]$ . For the convenience of the user, GP also accepts

```
a=[1,2-z,4*n-2];b=[2*z,z^2];
```

and a conversion back and forth to and from closures is done internally using the functions `cfvectoclos` and `cfclustovec`.

Important function `cfsubst`: since we cannot directly replace in a closure, specific function: `ab3=cfsubst([a,b],z,3)` gives the above CF for  $z = 3$ .

# Creating Continued Fractions II

There also exist a number of **creators** which create the necessary closures: most useful is **cffromser**, which converts a series into a continued fraction (due to Euler). Example:

```
? ab=cffromser(n->n^3,n->n^3);
/* Create CF corresponding to \sum_{n\ge 1}1/n^3 */
? cfclostovec(ab,'n)
% = [[0, 2*n^3 - 3*n^2 + 3*n - 1], [1, -n^6]]
? cftochar(ab)
% = "1/(1-1/(9-64/(35-729/(91-4096/(189-15625/(341))))))"
? print(cftotex(ab, 4))
\dfrac{1}{1-\dfrac{1}{9-\dfrac{64}{35-\dfrac{729}{91-\ddots}}}}
? cflimit(ab)
% = 1.2020569.... /* zeta(3) */
```

Above, 3 different ways of looking at the trivial continued fraction for  $\zeta(3)$ .

# Creating Continued Fractions III

`cffromser` can also be used in other ways:

```
? ab=cffromser(n->1/(n+1),-1);
/* CF corresponding to \log(2)=\sum_{n\ge 0}(-1)^n/(n+1) */
? ab=cffromser(exp(x)+O(x^11))
% = [Vecsmall([0, 1]), [1, -1, 1/2, -1/6, 1/6, -1/10]]
/* CF expansion of exp(x)
```

Also available is `cffromquad`:

```
? z=Mod(x,x^2-x-1);ab=cffromquad(z);cftochar(ab)
% = "1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1)))))))"
/* CF expansion of golden ratio, what else? */
? ab=cffromquad(z,-1);cftochar(ab)
% = "2-1/(3-1/(3-1/(3-1/(3-1/(3-1/(3)))))))"
/* CF expansion with -1 numerators, much faster convergence
```

# Creating Continued Fractions III

`cffromser` can also be used in other ways:

```
? ab=cffromser(n->1/(n+1),-1);
/* CF corresponding to \log(2)=\sum_{n\ge 0}(-1)^n/(n+1) */
? ab=cffromser(exp(x)+O(x^11))
% = [Vecsmall([0, 1]), [1, -1, 1/2, -1/6, 1/6, -1/10]]
/* CF expansion of exp(x)
```

Also available is `cffromquad`:

```
? z=Mod(x,x^2-x-1);ab=cffromquad(z);cftochar(ab)
% = "1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1)))))))"
/* CF expansion of golden ratio, what else? */
? ab=cffromquad(z,-1);cftochar(ab)
% = "2-1/(3-1/(3-1/(3-1/(3-1/(3-1/(3)))))))"
/* CF expansion with -1 numerators, much faster convergence
```

# Basic Functions on Continued Fractions I

`cfpnqn` gives the  $n$ th partial quotient, with different options:

```
? z=Mod(x,x^2-x-1);ab=cffromquad(z);cfpnqn(ab,5)
% = [[2, 1], [3, 2], [5, 3], [8, 5], [13, 8]]
? cfpnqn(ab,5,1) /* matrix [p(n),p(n-1);q(n),q(n-1)] */
% =
[13 8]

[ 8 5]
? cfpnqn(ab,5,2) /* p(n)/q(n) */
% = 13/8
? cfpnqn(ab,5,3)
% = 1/40 /* p(n)/q(n)-p(n-1)/q(n-1) */
```



# Basic Functions on Continued Fractions II

```
? a=[1,2-z,4*n-2];b=[2*z,z^2]; /* CF for e^z */  
? cflimit(cfstubst([a,b],z,1))  
% = 2.71828182845...  
? cftype([a,b])  
% = [[1, 4, 0, -1/2], [2, -16/z^2, 0, 0], 1]
```

The `cfstubst` command makes the substitution in the closures, `cflimit` computes the limit, and `cftype` the speed of convergence. The second vector gives  $[k, E, D, P] = [2, -16/z^2, 0, 0]$ , so that

$$e^z - \frac{p(n)}{q(n)} \sim \frac{C}{n!^2(-16/z^2)^n}$$

for some constant  $C$  (the first vector gives the asymptotics of  $q(n)$ , and the third is a convergence type number).

# Basic Functions on Continued Fractions III

The `cfasymp` command is more powerful and computes the necessary unknown constants in the asymptotics:

```
? cfasymp(cfsubst([a,b],z,1))
% = [2.71828..., [1, 4, 0, -1/2, 0.342...],
      [2, -16, 0, 0, 4.2698...], 1]
? lndep([-log(%[3][5]),1,log(Pi),log(2)])
% = [-1, -1, -1, 1]
```

We knew that  $e - p(n)/q(n) \sim C/n!^2(-16)^n$ , and the above command tells us that (numerically)  $C = (\pi/2)e$ . We could also check that the constant  $0.342\dots$  which occurs in the asymptotics of  $q(n)$  is equal to  $1/(e\pi)^{1/2}$ , so that  $q(n) \sim n!4^n n^{-1/2}/(e\pi)^{1/2}$ .

Note that `cftype` can handle unknown variables as above, but of course not `cflimit` or `cfasymp`.

# Basic Functions on Continued Fractions III

The `cfasymp` command is more powerful and computes the necessary unknown constants in the asymptotics:

```
? cfasymp(cfsubst([a,b],z,1))
% = [2.71828..., [1, 4, 0, -1/2, 0.342...],
      [2, -16, 0, 0, 4.2698...], 1]
? lndep([-log(%[3][5]),1,log(Pi),log(2)])
% = [-1, -1, -1, 1]
```

We knew that  $e - p(n)/q(n) \sim C/n!^2(-16)^n$ , and the above command tells us that (numerically)  $C = (\pi/2)e$ . We could also check that the constant  $0.342\dots$  which occurs in the asymptotics of  $q(n)$  is equal to  $1/(e\pi)^{1/2}$ , so that  $q(n) \sim n!4^n n^{-1/2}/(e\pi)^{1/2}$ .

Note that `cftype` can handle unknown variables as above, but of course not `cflimit` or `cfasymp`.

# Basic Functions on Continued Fractions IV

Two useful functions to go back and forth between closures and vectors are `cfvectoclos` and `cfclostovec`:

```
? [a,b]=[[1,2-z,4*n-2],[2*z,z^2]];ab=cfvectoclos([a,b])
%= [[(v1)->_cftoclos(v1,3,[1,-z+2,4*n-2],4*n-2),
      (v1)->_cftoclos(v1,2,[2*z,z^2],z^2)]
/* A pair of closures */
? cfclostovec(ab)
%= [[1,-z+2,4*n-2],[2*z,z^2]]
```

A simpler function `cftopol` gives only the generic entries (i.e., for  $n$  sufficiently large), but is useful since it applies to any vector of closures.

```
? cftopol(ab,'n')
%= [[4*n-2],[z^2]]
```

# Basic Functions on Continued Fractions IV

Two useful functions to go back and forth between closures and vectors are `cfvectoclos` and `cfclostovec`:

```
? [a,b]=[[1,2-z,4*n-2],[2*z,z^2]];ab=cfvectoclos([a,b])
%= [[(v1)->_cftoclos(v1,3,[1,-z+2,4*n-2],4*n-2),
      (v1)->_cftoclos(v1,2,[2*z,z^2],z^2)]
/* A pair of closures */
? cfclostovec(ab)
%= [[1,-z+2,4*n-2],[2*z,z^2]]
```

A simpler function `cftopol` gives only the generic entries (i.e., for  $n$  sufficiently large), but is useful since it applies to any vector of closures.

```
? cftopol(ab,'n')
%= [[4*n-2],[z^2]]
```

# Basic Functions on Continued Fractions V

Other useful functions are `cfmul` and `cfsimplify`. Example for `cfmul`:

```
? ab4=cfsbst([a,b],z,1/4)
% = [[1, 7/4, 4*n - 2], [1/2, 1/16]]
? cftochar(ab4)
% = "1+1/2/(7/4+1/16/(6+1/16/(10+1/16/(14+1/16/(18+1/16/(22+1/16/(26+1/16/(30+1/16/(34+1/16/(38+1/16/(42+1/16/(46+1/16/(50+1/16/(54+1/16/(58+1/16/(62+1/16/(66+1/16/(70+1/16/(74+1/16/(78+1/16/(82+1/16/(86+1/16/(90+1/16/(94+1/16/(98+1/16/(102+1/16/(106+1/16/(110+1/16/(114+1/16/(118+1/16/(122+1/16/(126+1/16/(130+1/16/(134+1/16/(138+1/16/(142+1/16/(146+1/16/(150+1/16/(154+1/16/(158+1/16/(162+1/16/(166+1/16/(170+1/16/(174+1/16/(178+1/16/(182+1/16/(186+1/16/(190+1/16/(194+1/16/(198+1/16/(202+1/16/(206+1/16/(210+1/16/(214+1/16/(218+1/16/(222+1/16/(226+1/16/(230+1/16/(234+1/16/(238+1/16/(242+1/16/(246+1/16/(250+1/16/(254+1/16/(258+1/16/(262+1/16/(266+1/16/(270+1/16/(274+1/16/(278+1/16/(282+1/16/(286+1/16/(290+1/16/(294+1/16/(298+1/16/(302+1/16/(306+1/16/(310+1/16/(314+1/16/(318+1/16/(322+1/16/(326+1/16/(330+1/16/(334+1/16/(338+1/16/(342+1/16/(346+1/16/(350+1/16/(354+1/16/(358+1/16/(362+1/16/(366+1/16/(370+1/16/(374+1/16/(378+1/16/(382+1/16/(386+1/16/(390+1/16/(394+1/16/(398+1/16/(402+1/16/(406+1/16/(410+1/16/(414+1/16/(418+1/16/(422+1/16/(426+1/16/(430+1/16/(434+1/16/(438+1/16/(442+1/16/(446+1/16/(450+1/16/(454+1/16/(458+1/16/(462+1/16/(466+1/16/(470+1/16/(474+1/16/(478+1/16/(482+1/16/(486+1/16/(490+1/16/(494+1/16/(498+1/16/(502+1/16/(506+1/16/(510+1/16/(514+1/16/(518+1/16/(522+1/16/(526+1/16/(530+1/16/(534+1/16/(538+1/16/(542+1/16/(546+1/16/(550+1/16/(554+1/16/(558+1/16/(562+1/16/(566+1/16/(570+1/16/(574+1/16/(578+1/16/(582+1/16/(586+1/16/(590+1/16/(594+1/16/(598+1/16/(602+1/16/(606+1/16/(610+1/16/(614+1/16/(618+1/16/(622+1/16/(626+1/16/(630+1/16/(634+1/16/(638+1/16/(642+1/16/(646+1/16/(650+1/16/(654+1/16/(658+1/16/(662+1/16/(666+1/16/(670+1/16/(674+1/16/(678+1/16/(682+1/16/(686+1/16/(690+1/16/(694+1/16/(698+1/16/(702+1/16/(706+1/16/(710+1/16/(714+1/16/(718+1/16/(722+1/16/(726+1/16/(730+1/16/(734+1/16/(738+1/16/(742+1/16/(746+1/16/(750+1/16/(754+1/16/(758+1/16/(762+1/16/(766+1/16/(770+1/16/(774+1/16/(778+1/16/(782+1/16/(786+1/16/(790+1/16/(794+1/16/(798+1/16/(802+1/16/(806+1/16/(810+1/16/(814+1/16/(818+1/16/(822+1/16/(826+1/16/(830+1/16/(834+1/16/(838+1/16/(842+1/16/(846+1/16/(850+1/16/(854+1/16/(858+1/16/(862+1/16/(866+1/16/(870+1/16/(874+1/16/(878+1/16/(882+1/16/(886+1/16/(890+1/16/(894+1/16/(898+1/16/(902+1/16/(906+1/16/(910+1/16/(914+1/16/(918+1/16/(922+1/16/(926+1/16/(930+1/16/(934+1/16/(938+1/16/(942+1/16/(946+1/16/(950+1/16/(954+1/16/(958+1/16/(962+1/16/(966+1/16/(970+1/16/(974+1/16/(978+1/16/(982+1/16/(986+1/16/(990+1/16/(994+1/16/(998+1/16/(1002+1/16/(1006+1/16/(1010+1/16/(1014+1/16/(1018+1/16/(1022+1/16/(1026+1/16/(1030+1/16/(1034+1/16/(1038+1/16/(1042+1/16/(1046+1/16/(1050+1/16/(1054+1/16/(1058+1/16/(1062+1/16/(1066+1/16/(1070+1/16/(1074+1/16/(1078+1/16/(1082+1/16/(1086+1/16/(1090+1/16/(1094+1/16/(1098+1/16/(1102+1/16/(1106+1/16/(1110+1/16/(1114+1/16/(1118+1/16/(1122+1/16/(1126+1/16/(1130+1/16/(1134+1/16/(1138+1/16/(1142+1/16/(1146+1/16/(1150+1/16/(1154+1/16/(1158+1/16/(1162+1/16/(1166+1/16/(1170+1/16/(1174+1/16/(1178+1/16/(1182+1/16/(1186+1/16/(1190+1/16/(1194+1/16/(1198+1/16/(1202+1/16/(1206+1/16/(1210+1/16/(1214+1/16/(1218+1/16/(1222+1/16/(1226+1/16/(1230+1/16/(1234+1/16/(1238+1/16/(1242+1/16/(1246+1/16/(1250+1/16/(1254+1/16/(1258+1/16/(1262+1/16/(1266+1/16/(1270+1/16/(1274+1/16/(1278+1/16/(1282+1/16/(1286+1/16/(1290+1/16/(1294+1/16/(1298+1/16/(1302+1/16/(1306+1/16/(1310+1/16/(1314+1/16/(1318+1/16/(1322+1/16/(1326+1/16/(1330+1/16/(1334+1/16/(1338+1/16/(1342+1/16/(1346+1/16/(1350+1/16/(1354+1/16/(1358+1/16/(1362+1/16/(1366+1/16/(1370+1/16/(1374+1/16/(1378+1/16/(1382+1/16/(1386+1/16/(1390+1/16/(1394+1/16/(1398+1/16/(1402+1/16/(1406+1/16/(1410+1/16/(1414+1/16/(1418+1/16/(1422+1/16/(1426+1/16/(1430+1/16/(1434+1/16/(1438+1/16/(1442+1/16/(1446+1/16/(1450+1/16/(1454+1/16/(1458+1/16/(1462+1/16/(1466+1/16/(1470+1/16/(1474+1/16/(1478+1/16/(1482+1/16/(1486+1/16/(1490+1/16/(1494+1/16/(1498+1/16/(1502+1/16/(1506+1/16/(1510+1/16/(1514+1/16/(1518+1/16/(1522+1/16/(1526+1/16/(1530+1/16/(1534+1/16/(1538+1/16/(1542+1/16/(1546+1/16/(1550+1/16/(1554+1/16/(1558+1/16/(1562+1/16/(1566+1/16/(1570+1/16/(1574+1/16/(1578+1/16/(1582+1/16/(1586+1/16/(1590+1/16/(1594+1/16/(1598+1/16/(1602+1/16/(1606+1/16/(1610+1/16/(1614+1/16/(1618+1/16/(1622+1/16/(1626+1/16/(1630+1/16/(1634+1/16/(1638+1/16/(1642+1/16/(1646+1/16/(1650+1/16/(1654+1/16/(1658+1/16/(1662+1/16/(1666+1/16/(1670+1/16/(1674+1/16/(1678+1/16/(1682+1/16/(1686+1/16/(1690+1/16/(1694+1/16/(1698+1/16/(1702+1/16/(1706+1/16/(1710+1/16/(1714+1/16/(1718+1/16/(1722+1/16/(1726+1/16/(1730+1/16/(1734+1/16/(1738+1/16/(1742+1/16/(1746+1/16/(1750+1/16/(1754+1/16/(1758+1/16/(1762+1/16/(1766+1/16/(1770+1/16/(1774+1/16/(1778+1/16/(1782+1/16/(1786+1/16/(1790+1/16/(1794+1/16/(1798+1/16/(1802+1/16/(1806+1/16/(1810+1/16/(1814+1/16/(1818+1/16/(1822+1/16/(1826+1/16/(1830+1/16/(1834+1/16/(1838+1/16/(1842+1/16/(1846+1/16/(1850+1/16/(1854+1/16/(1858+1/16/(1862+1/16/(1866+1/16/(1870+1/16/(1874+1/16/(1878+1/16/(1882+1/16/(1886+1/16/(1890+1/16/(1894+1/16/(1898+1/16/(1902+1/16/(1906+1/16/(1910+1/16/(1914+1/16/(1918+1/16/(1922+1/16/(1926+1/16/(1930+1/16/(1934+1/16/(1938+1/16/(1942+1/16/(1946+1/16/(1950+1/16/(1954+1/16/(1958+1/16/(1962+1/16/(1966+1/16/(1970+1/16/(1974+1/16/(1978+1/16/(1982+1/16/(1986+1/16/(1990+1/16/(1994+1/16/(1998+1/16/(2002+1/16/(2006+1/16/(2010+1/16/(2014+1/16/(2018+1/16/(2022+1/16/(2026+1/16/(2030+1/16/(2034+1/16/(2038+1/16/(2042+1/16/(2046+1/16/(2050+1/16/(2054+1/16/(2058+1/16/(2062+1/16/(2066+1/16/(2070+1/16/(2074+1/16/(2078+1/16/(2082+1/16/(2086+1/16/(2090+1/16/(2094+1/16/(2098+1/16/(2102+1/16/(2106+1/16/(2110+1/16/(2114+1/16/(2118+1/16/(2122+1/16/(2126+1/16/(2130+1/16/(2134+1/16/(2138+1/16/(2142+1/16/(2146+1/16/(2150+1/16/(2154+1/16/(2158+1/16/(2162+1/16/(2166+1/16/(2170+1/16/(2174+1/16/(2178+1/16/(2182+1/16/(2186+1/16/(2190+1/16/(2194+1/16/(2198+1/16/(2202+1/16/(2206+1/16/(2210+1/16/(2214+1/16/(2218+1/16/(2222+1/16/(2226+1/16/(2230+1/16/(2234+1/16/(2238+1/16/(2242+1/16/(2246+1/16/(2250+1/16/(2254+1/16/(2258+1/16/(2262+1/16/(2266+1/16/(2270+1/16/(2274+1/16/(2278+1/16/(2282+1/16/(2286+1/16/(2290+1/16/(2294+1/16/(2298+1/16/(2302+1/16/(2306+1/16/(2310+1/16/(2314+1/16/(2318+1/16/(2322+1/16/(2326+1/16/(2330+1/16/(2334+1/16/(2338+1/16/(2342+1/16/(2346+1/16/(2350+1/16/(2354+1/16/(2358+1/16/(2362+1/16/(2366+1/16/(2370+1/16/(2374+1/16/(2378+1/16/(2382+1/16/(2386+1/16/(2390+1/16/(2394+1/16/(2398+1/16/(2402+1/16/(2406+1/16/(2410+1/16/(2414+1/16/(2418+1/16/(2422+1/16/(2426+1/16/(2430+1/16/(2434+1/16/(2438+1/16/(2442+1/16/(2446+1/16/(2450+1/16/(2454+1/16/(2458+1/16/(2462+1/16/(2466+1/16/(2470+1/16/(2474+1/16/(2478+1/16/(2482+1/16/(2486+1/16/(2490+1/16/(2494+1/16/(2498+1/16/(2502+1/16/(2506+1/16/(2510+1/16/(2514+1/16/(2518+1/16/(2522+1/16/(2526+1/16/(2530+1/16/(2534+1/16/(2538+1/16/(2542+1/16/(2546+1/16/(2550+1/16/(2554+1/16/(2558+1/16/(2562+1/16/(2566+1/16/(2570+1/16/(2574+1/16/(2578+1/16/(2582+1/16/(2586+1/16/(2590+1/16/(2594+1/16/(2598+1/16/(2602+1/16/(2606+1/16/(2610+1/16/(2614+1/16/(2618+1/16/(2622+1/16/(2626+1/16/(2630+1/16/(2634+1/16/(2638+1/16/(2642+1/16/(2646+1/16/(2650+1/16/(2654+1/16/(2658+1/16/(2662+1/16/(2666+1/16/(2670+1/16/(2674+1/16/(2678+1/16/(2682+1/16/(2686+1/16/(2690+1/16/(2694+1/16/(2698+1/16/(2702+1/16/(2706+1/16/(2710+1/16/(2714+1/16/(2718+1/16/(2722+1/16/(2726+1/16/(2730+1/16/(2734+1/16/(2738+1/16/(2742+1/16/(2746+1/16/(2750+1/16/(2754+1/16/(2758+1/16/(2762+1/16/(2766+1/16/(2770+1/16/(2774+1/16/(2778+1/16/(2782+1/16/(2786+1/16/(2790+1/16/(2794+1/16/(2798+1/16/(2802+1/16/(2806+1/16/(2810+1/16/(2814+1/16/(2818+1/16/(2822+1/16/(2826+1/16/(2830+1/16/(2834+1/16/(2838+1/16/(2842+1/16/(2846+1/16/(2850+1/16/(2854+1/16/(2858+1/16/(2862+1/16/(2866+1/16/(2870+1/16/(2874+1/16/(2878+1/16/(2882+1/16/(2886+1/16/(2890+1/16/(2894+1/16/(2898+1/16/(2902+1/16/(2906+1/16/(2910+1/16/(2914+1/16/(2918+1/16/(2922+1/16/(2926+1/16/(2930+1/16/(2934+1/16/(2938+1/16/(2942+1/16/(2946+1/16/(2950+1/16/(2954+1/16/(2958+1/16/(2962+1/16/(2966+1/16/(2970+1/16/(2974+1/16/(2978+1/16/(2982+1/16/(2986+1/16/(2990+1/16/(2994+1/16/(2998+1/16/(3002+1/16/(3006+1/16/(3010+1/16/(3014+1/16/(3018+1/16/(3022+1/16/(3026+1/16/(3030+1/16/(3034+1/16/(3038+1/16/(3042+1/16/(3046+1/16/(3050+1/16/(3054+1/16/(3058+1/16/(3062+1/16/(3066+1/16/(3070+1/16/(3074+1/16/(3078+1/16/(3082+1/16/(3086+1/16/(3090+1/16/(3094+1/16/(3098+1/16/(3102+1/16/(3106+1/16/(3110+1/16/(3114+1/16/(3118+1/16/(3122+1/16/(3126+1/16/(3130+1/16/(3134+1/16/(3138+1/16/(3142+1/16/(3146+1/16/(3150+1/16/(3154+1/16/(3158+1/16/(3162+1/16/(3166+1/16/(3170+1/16/(3174+1/16/(3178+1/16/(3182+1/16/(3186+1/16/(3190+1/16/(3194+1/16/(3198+1/16/(3202+1/16/(3206+1/16/(3210+1/16/(3214+1/16/(3218+1/16/(3222+1/16/(3226+1/16/(3230+1/16/(3234+1/16/(3238+1/16/(3242+1/16/(3246+1/16/(3250+1/16/(3254+1/16/(3258+1/16/(3262+1/16/(3266+1/16/(3270+1/16/(3274+1/16/(3278+1/16/(3282+1/16/(3286+1/16/(3290+1/16/(3294+1/16/(3298+1/16/(3302+1/16/(3306+1/16/(3310+1/16/(3314+1/16/(3318+1/16/(3322+1/16/(3326+1/16/(3330+1/16/(3334+1/16/(3338+1/16/(3342+1/16/(3346+1/16/(3350+1/16/(3354+1/16/(3358+1/16/(3362+1/16/(3366+1/16/(3370+1/16/(3374+1/16/(3378+1/16/(3382+1/16/(3386+1/16/(3390+1/16/(3394+1/16/(3398+1/16/(3402+1/16/(3406+1/16/(3410+1/16/(3414+1/16/(3418+1/16/(3422+1/16/(3426+1/16/(3430+1/16/(3434+1/16/(3438+1/16/(3442+1/16/(3446+1/16/(3450+1/16/(3454+1/16/(3458+1/16/(3462+1/16/(3466+1/16/(3470+1/16/(3474+1/16/(3478+1/16/(3482+1/16/(3486+1/16/(3490+1/16/(3494+1/16/(3498+1/16/(3502+1/16/(3506+1/16/(3510+1/16/(3514+1/16/(3518+1/16/(3522+1/16/(3526+1/16/(3530+1/16/(3534+1/16/(3538+1/16/(3542+1/16/(3546+1/16/(3550+1/16/(3554+1/16/(3558+1/16/(3562+1/16/(3566+1/16/(3570+1/16/(3574+1/16/(3578+1/16/(3582+1/16/(3586+1/16/(3590+1/16/(3594+1/16/(3598+1/16/(3602+1/16/(3606+1/16/(3610+1/16/(3614+1/16/(3618+1/16/(3622+1/16/(3626+1/16/(3630+1/16/(3634+1/16/(3638+1/16/(3642+1/16/(3646+1/16/(3650+1/16/(3654+1/16/(3658+1/16/(3662+1/16/(3666+1/16/(3670+1/16/(3674+1/16/(3678+1/16/(3682+1/16/(3686+1/16/(3690+1/16/(3694+1/16/(3698+1/16/(3702+1/16/(3706+1/16/(3710+1/16/(3714+1/16/(3718+1/16/(3722+1/16/(3726+1/16/(3730+1/16/(3734+1/16/(3738+1/16/(3742+1/16/(3746+1/16/(3750+1/16/(3754+1/16/(3758+1/16/(3762+1/16/(3766+1/16/(3770+1/16/(3774+1/16/(3778+1/16/(3782+1/16/(3786+1/16/(3790+1/16/(3794+1/16/(3798+1/16/(3802+1/16/(3806+1/16/(3810+1/16/(3814+1/16/(3818+1/16/(3822+1/16/(3826+1/16/(3830+1/16/(3834+1/16/(3838+1/16/(3842+1/16/(3846+1/16/(3850+1/16/(3854+1/16/(3858+1/16/(3862+1/16/(3866+1/16/(3870+1/16/(3874+1/16/(3878+1/16/(3882+1/16/(3886+1/16/(3890+1/16/(3894+1/16/(3898+1/16/(3902+1/16/(3906+1/16/(3910+1/16/(3914+1/16/(3918+1/16/(3922+1/16/(3926+1/16/(3930+1/16/(3934+1/16/(3938+1/16/(3942+1/16/(3946+1/16/(3950+1/16/(3954+1/16/(3958+1/16/(3962+1/16/(3966+1/16/(3970+1/16/(3974+1/16/(3978+1/16/(3982+1/16/(3986+1/16/(3990+1/16/(3994+1/16/(3998+1/16/(4002+1/16/(4006+1/16/(4010+1/16/(4014+1/16/(4018+1/16/(4022+1/16/(4026+1/16/(4030+1/16/(4034+1/16/(4038+1/16/(4042+1/16/(4046+1/16/(4050+1/16/(4054+1/16/(4058+1/16/(4062+1/16/(4066+1/16/(4070+1/16/(4074+1/16/(4078+1/16/(4082+1/16/(4086+1/16/(4090+1/16/(4094+1/16/(4098+1/16/(4102+1/16/(4106+1/16/(4110+1/16/(4114+1/16/(4118+1/16/(4122+1/16/(4126+1/16/(4130+1/16/(4134+1/16/(4138+1/16/(4142+1/16/(4146+1/16/(4150+1/16/(4154+1/16/(4158+1/16/(4162+1/16/(4166+1/16/(4170+1/16/(4174+1/16/(4178+1/16/(4182+1/16/(4186+1/16/(4190+1/16/(4194+1/16/(4198+1/16/(4202+1/16/(4206+1/16/(4210+1/16/(4214+1/16/(4218+1/16/(4222+1/16/(4226+1/16/(4230+1/16/(4234+1/16/(4238+1/16/(4242+1/16/(4246+1/16/(4250+1/16/(4254+1/16/(4258+1/16/(4262+1/16/(4266+1/16/(4270+1/16/(4274+1/16/(4278+1/16/(4282+1/16/(4286+1/16/(4290+1/16/(4294+1/16/(4298+1/16/(4302+1/16/(4306+1/16/(4310+1/16/(4314+1/16/(4318+1/16/(4322+1/16/(4326+1/16/(4330+1/16/(4334+1/16/(4338+1/16/(4342+1/16/(4346+1/16/(4350+1/16/(4354+1/16/(4358+1/16/(4362+1/16/(4366+1/16/(4370+1/16/(4374+1/16/(4378+1/16/(4382+1/16/(4386+1/16/(4390+1/16/(4394+1/16/(4398+1/16/(4402+1/16/(4406+1/16/(4410+1/16/(4414+1/16/(4418+1/16/(4422+1/16/(4426+1/16/(4430+1/16/(4434+1/16/(4438+1/16/(4442+1/16/(4446+1/16/(4450+1/16/(4454+1/16/(4458+1/16/(4462+1/16/(4466+1/16/(4470+1/16/(4474+1/16/(4478+1/16/(4482+1/16/(4486+1/16/(4490+1/16/(4494+1/16/(4498+1/16/(4502+1/16/(4506+1/16/(4510+1/16/(4514+1/16/(4518+1/16/(4522+1/16/(4526+1/16/(4530+1/16/(4534+1/16/(4538+1/16/(4542+1/16/(4546+1/16/(4550+1/16/(4554+1/16/(4558+1/16/(4562+1/16/(4566+1/16/(4570+1/16/(4574+1/16/(4578+1/16/(4582+1/16/(4586+1/16/(4590+1/16/(4594+1/16/(4598+1/16/(4602+1/16/(4606+1/16/(4610+1/16/(4614+1/16/(4618+1/16/(4622+1/16/(4626+1/16/(4630+1/16/(4634+1/16/(4638+1/16/(4642+1/16/(4646+1/16/(4650+1/16/(4654+1/16/(4658+1/16/(4662+1/16/(4666+1/16/(4670+1/16/(4674+1/16/(4678+1/16/(4682+1/16/(4686+1/16/(4690+1/16/(4694+1/16/(4698+1/16/(4702+1/16/(4706+1/16/(4710+1/16/(4714+1/16/(4718+1/16/(4722+1/16/(4726+1/16/(4730+1/16/(4734+1/16/(4738+1/16/(4742+1/16/(4746+1/16/(4750+1/16/(4754+1/16/(4758+1/16/(4762+1/16/(4766+1/16/(4770+1/16/(4774+1/16/(4778+1/16/(4782+1/16/(4786+1/16/(4790+1/16/(4794+1/16/(4798+1/16/(4802+1/16/(4806+1/16/(4810+1/16/(4814+1/16/(4818+1/16/(4822+1/16/(4826+1/16/(4830+1/16/(4834+1/16/(4838+1/16/(4842+1/16/(4846+1/16/(4850+1/16/(4854+1/16/(4858+1/16/(4862+1/16/(4866+1/16/(4870+1/16/(4874+1/16/(4878+1/16/(4882+1/16/(4886+1/16/(4890+1/16/(4894+1/16/(4898+1/16/(4902+1/16/(4906+1/16/(4910+1/16/(4914+1/16/(4918+1/16/(4922+1/16/(4926+1/16/(4930+1/16/(4934+1/16/(4938+1/16/(4942+1/16/(4946+1/16/(4950+1/16/(4954+1/16/(4958+1/16/(4962+1/16/(4966+1/16/(4970+1/16/(4974+1/16/(4978+1/16/(4982+1/16/(49
```

# Basic Functions on Continued Fractions VI

```
? ab=[[1,1/n],[1]];cftochar(ab)
% = "1+1/(1+1/(1/2+1/(1/3+1/(1/4+1/(1/5+1/(1/6))))))"
/* CF for Pi/2 */
/* Again lots of fractions.
We could write ab2=cfmul(ab,n), but simpler: */
? ab2=cfsimplify(ab)
% = [[1, 1], [1, n^2 + n]]
? cftochar(ab2)
% = "1+1/(1+2/(1+6/(1+12/(1+20/(1+30/(1))))))"
/* Much neater */
```

Note that **cfsimplify** removes denominators (and additional simplifications), but only as functions of the **reserved** variable **n**, not of **z**, or scalar fractions, for this need to use **cfmul** explicitly.

# Period 2 Continued Fractions I

We have seen above the period 2 continued fraction for

$$f(z) = (1 + z) \log(1 + z)/z:$$

$$f(z) = 1 + z/(2 + 1.2z/(3 + 1.2z/(4 + 2.3z/(5 + 2.3z/(6 + \dots))))))$$

This is entered in GP with 2-component vectors when necessary:

$$? a=[n+1]; b=z*[[1,2], [n*(n+1), (n+1)*(n+2)]]$$

Note that we would have preferred to write

$b=z*[1, [n*(n+1), n*(n+1)]]$ , but this (for now) is not permitted by the package: in  $a$  or  $b$ , all entries must be either scalars, or 2-component vectors. A mixture is not allowed (but  $a$  can be scalar and  $b$  2-component as above, or conversely), and neither are  $N$ -component vectors for  $N \geq 3$ .



## Period 2 Continued Fractions II

Such a CF is treated like the others, but:

- The `cf_simplify` command **contracts** the CF by computing the CF corresponding to  $p(2n)/q(2n)$ , and then applies the usual simplifications that it can find. The `cf_contract` command can also do this on scalar-valued CF.
- The `cf_type` command giving the speed of convergence is applied to the contracted CF, and then  $n$  is changed into  $n/2$ . Thus, if  $p(2n+1)/q(2n+1)$  is asymptotically very different from  $p(2n)/q(2n)$ , care must be applied.

## Period 2 Continued Fractions III

Example:

```
? a=[n+1];b=z*[[1,2],[n*(n+1),(n+1)*(n+2)]];
/* CF for (1+z)\log(1+z)/z seen above */
? cftochar([a,b])
% = "1+z/(2+2*z/(3+2*z/(4+6*z/(5+6*z/(6+12*z/(7))))))"
? cd=cfsimplify([a,b])
% = [[1, (4*z + 8)*n^2 + (-2*z - 2)],
      [3*z, -4*z^2*n^4 - 8*z^2*n^3 - z^2*n^2 + 3*z^2*n]]
? cftochar(cd,3)
%122 = "1+3*z/(2*z+6-10*z^2/(14*z+30-126*z^2/(34*z+70)))"
```

This “simplified” CF is not really simpler than the period 2 CF, so in practice this is not recorded as an interesting CF.

# Convergence Acceleration I

This is perhaps the most spectacular aspect of the package. Initial method due to **Bauer–Muir**. Recall that the  $n$ th **tail** of a CF (which can be computed numerically by the command `cflimit([a,b],n)`) is

$$\rho(n) = b(n)/(a(n+1)+b(n+1)/(a(n+2)+b(n+2)/(a(n+3)+\dots))) ,$$

and that the limit  $S$  is given by

$$S = \frac{\rho(n+1) + \rho(n)p(n)}{q(n+1) + \rho(n)q(n)} .$$

Since trivially  $\rho(n)(a(n+1) + \rho(n+1)) - b(n) = 0$ , if we choose some  $r(n)$  such that

$$d(n) = r(n)(a(n+1) + r(n+1)) - b(n)$$

is small, we can hope that

$(\rho(n+1) + r(n)p(n))/(q(n+1) + r(n)q(n))$  is a better approximation to  $S$ .

# Convergence Acceleration I

This is perhaps the most spectacular aspect of the package. Initial method due to **Bauer–Muir**. Recall that the  $n$ th **tail** of a CF (which can be computed numerically by the command `cflimit([a,b],n)`) is

$$\rho(n) = b(n)/(a(n+1)+b(n+1)/(a(n+2)+b(n+2)/(a(n+3)+\dots))) ,$$

and that the limit  $S$  is given by

$$S = \frac{\rho(n+1) + \rho(n)p(n)}{q(n+1) + \rho(n)q(n)} .$$

Since trivially  $\rho(n)(a(n+1) + \rho(n+1)) - b(n) = 0$ , if we choose some  $r(n)$  such that

$$d(n) = r(n)(a(n+1) + r(n+1)) - b(n)$$

is small, we can hope that

$(\rho(n+1) + r(n)p(n))/(q(n+1) + r(n)q(n))$  is a better approximation to  $S$ .

# Convergence Acceleration II

Easy to write explicit formulas for  $(a'(n), b'(n))$  so that the corresponding CF is such that the partial quotients are  $(p(n+1) + r(n)p(n))/(q(n+1) + r(n)q(n))$ : the formulas involve  $d(n)/d(n-1)$ , with

$$d(n) = r(n)(a(n+1) + r(n+1)) - b(n)$$

as above.

Ideal situation: by choosing  $r(n)$  suitably, make  $d(n)$  a nonzero **constant**, but even otherwise can be useful. Since  $a(n), b(n)$  polynomials, choose  $r(n)$  a polynomial with unknown coefficients and solve for each by decreasing degree (no need for Gröbner bases, only linear or quadratic equations).

# Convergence Acceleration II

Easy to write explicit formulas for  $(a'(n), b'(n))$  so that the corresponding CF is such that the partial quotients are  $(p(n+1) + r(n)p(n))/(q(n+1) + r(n)q(n))$ : the formulas involve  $d(n)/d(n-1)$ , with

$$d(n) = r(n)(a(n+1) + r(n+1)) - b(n)$$

as above.

Ideal situation: by choosing  $r(n)$  suitably, make  $d(n)$  a nonzero **constant**, but even otherwise can be useful. Since  $a(n), b(n)$  polynomials, choose  $r(n)$  a polynomial with unknown coefficients and solve for each by decreasing degree (no need for Gröbner bases, only linear or quadratic equations).

# Convergence Acceleration III

When  $a(n)$  is of degree less than equal to 2 and  $b(n)$  less than or equal to 4, almost always possible. When  $a(n)$  has degree 3 and  $b(n)$  degree 6, possible in some cases. In higher degree very rare, but method can be modified.

Depending on the five rates of convergence, can see if we obtain acceleration using  $r(n)$  a polynomial with rational coefficients (assuming  $a(n)$  and  $b(n)$  same assumptions):

- Polynomial convergence: always possible, usually changes  $P$  into  $P + 2$ , most useful since convergence slow.
- Exponential convergence: always possible if  $E$  is rational, does not change  $E$  but again  $P$  into  $P + 2$ . Less useful since rapid convergence, but essential for Apéry's method.
- Subexponential convergence: never possible, but can give new CF with same convergence.

# Convergence Acceleration III

When  $a(n)$  is of degree less than equal to 2 and  $b(n)$  less than or equal to 4, almost always possible. When  $a(n)$  has degree 3 and  $b(n)$  degree 6, possible in some cases. In higher degree very rare, but method can be modified.

Depending on the five rates of convergence, can see if we obtain acceleration using  $r(n)$  a polynomial with rational coefficients (assuming  $a(n)$  and  $b(n)$  same assumptions):

- **Polynomial** convergence: always possible, usually changes  $P$  into  $P + 2$ , most useful since convergence slow.
- **Exponential** convergence: always possible if  $E$  is rational, does not change  $E$  but again  $P$  into  $P + 2$ . Less useful since rapid convergence, but **essential** for **Apéry's method**.
- **Subexponential** convergence: never possible, **but** can give **new** CF with same convergence.



# The `cfbauer` Command I

All this is done by the command `cfbauer`, which outputs  $[a'(n), b'(n), r(n), d(n)]$  with flag `1`, and only  $[a'(n), b'(n)]$  with the default flag `0`. Slight subtlety: the indices of the new CF are shifted by 1 compared to the initial one, compensate by contracting the first 2 coeffs, done by `cfbauer`.

```
? ab=cfclostovec(cffromser(n->n^3,n->n^3))
% = [[0, 2*n^3 - 3*n^2 + 3*n - 1], [1, -n^6]]
/* Trivial CF for zeta(3) */
? cftype(ab)[2]
% = [0, 1, 0, 2] /* Convergence in C/n^2 */
? abrd1=cfbauer(ab,1)
% = [[1, 2*n^3 - 3*n^2 + 11*n - 5], [1, -n^6],
      [1, -n^3 + 2*n^2 - 2*n + 1], [0, 1]]
/* Use r(n)=-n^3+2*n^2-2*n+1 to accelerate, get d(n)=1 */
? ab1=abrd1[1..2];cftype(ab1)[2]
% = [0, 1, 0, 6] /* Convergence in C/n^6 */
```



# The `cfbauer` Command I

All this is done by the command `cfbauer`, which outputs  $[a'(n), b'(n), r(n), d(n)]$  with flag `1`, and only  $[a'(n), b'(n)]$  with the default flag `0`. Slight subtlety: the indices of the new CF are shifted by 1 compared to the initial one, compensate by contracting the first 2 coeffs, done by `cfbauer`.

```
? ab=cfclostovec(cffromser(n->n^3,n->n^3))
% = [[0, 2*n^3 - 3*n^2 + 3*n - 1], [1, -n^6]]
/* Trivial CF for zeta(3) */
? cftype(ab)[2]
% = [0, 1, 0, 2] /* Convergence in C/n^2 */
? abrd1=cfbauer(ab,1)
% = [[1, 2*n^3 - 3*n^2 + 11*n - 5], [1, -n^6],
      [1, -n^3 + 2*n^2 - 2*n + 1], [0, 1]]
/* Use r(n)=-n^3+2*n^2-2*n+1 to accelerate, get d(n)=1 */
? ab1=abrd1[1..2];cftype(ab1)[2]
% = [0, 1, 0, 6] /* Convergence in C/n^6 */
```

# The `cfbauer` Command II

We can continue as long as we like:

```
? ab2=cfbauer(ab1)
% = [[9/8, 2*n^3 - 3*n^2 + 27*n - 13], [1, -n^6]]
? cftype(ab2)[2]
% = [0, 1, 0, 10] /* Convergence in C/n^10 */
```

This iteration, followed by a diagonal process, is the basis for the `cfapery` command.

The above was `cfbauer` applied to a CF with polynomial convergence. We will see below an example of an application to a CF with exponential convergence: the exponent stays the same, but the polynomial part increases as above, and the `Apéry` acceleration will increase the exponent.

# The `cfbauer` Command II

We can continue as long as we like:

```
? ab2=cfbauer(ab1)
% = [[9/8, 2*n^3 - 3*n^2 + 27*n - 13], [1, -n^6]]
? cftype(ab2)[2]
% = [0, 1, 0, 10] /* Convergence in C/n^10 */
```

This iteration, followed by a diagonal process, is the basis for the `cfapery` command.

The above was `cfbauer` applied to a CF with polynomial convergence. We will see below an example of an application to a CF with exponential convergence: the exponent stays the same, but the polynomial part increases as above, and the `Apéry` acceleration will increase the exponent.

# The `cfbauer` Command III

Finally, an example of `cfbauer` applied to **subexponential** convergence: no acceleration at all, but a **new** CF. Here is a beautiful CF:

$$\int_0^{\infty} \frac{e^{-t}}{t+1} dt = 1/(2 - 1^2/(4 - 2^2/(6 - 3^2/(8 - 4^2/(10 - \dots))))))$$

Speed of convergence  $2\pi e/e^{4\sqrt{n}}$ .

Applying `cfbauer`, we arrive at the new CF:

$$\int_0^{\infty} \frac{e^{-t}}{t+1} dt = 1 - 1/(3 - 1.2/(5 - 2.3/(7 - 3.4/(9 - 4.5/(10 - \dots))))))$$

exactly the same speed of convergence. Of course we can iterate.

# The `cfbauer` Command III

Finally, an example of `cfbauer` applied to **subexponential** convergence: no acceleration at all, but a **new** CF. Here is a beautiful CF:

$$\int_0^{\infty} \frac{e^{-t}}{t+1} dt = 1/(2 - 1^2/(4 - 2^2/(6 - 3^2/(8 - 4^2/(10 - \dots))))))$$

Speed of convergence  $2\pi e/e^{4\sqrt{n}}$ .

Applying `cfbauer`, we arrive at the new CF:

$$\int_0^{\infty} \frac{e^{-t}}{t+1} dt = 1 - 1/(3 - 1.2/(5 - 2.3/(7 - 3.4/(9 - 4.5/(10 - \dots))))))$$

exactly the same speed of convergence. Of course we can iterate.

# Apéry Acceleration I

We have seen that the `cfbauer` command can be **iterated**. This is the basis of Apéry acceleration, given in the package as `cfapery`. Starting from a CF  $(a(n), b(n), p(n), q(n))$  we set  $(a(n, 0), b(n, 0), p(n, 0), q(n, 0)) = (a(n), b(n), p(n), q(n))$ , and denote by  $(a(n, l), b(n, l), p(n, l), q(n, l))$  the  $l$ th CF obtained by successive Bauer–Muir accelerations using  $(r(n, l), d(n, l))$ . We thus obtain **two-dimensional arrays**, all linked by **3-term linear recursions**, and hopefully with not too complicated coefficients (in particular, if possible, with  $(a(n, l), b(n, l), r(n, l), d(n, l))$  **polynomials** in  $n$  and  $l$ ).

The game then consists in **traveling** in this array as one likes. Ideal acceleration would be in traveling along the **diagonal**  $l = n$ : however, complicated coefficients. Instead, travel along a **staircase** either above or below the diagonal.

# Apéry Acceleration I

We have seen that the `cfbauer` command can be **iterated**. This is the basis of Apéry acceleration, given in the package as `cfapery`. Starting from a CF  $(a(n), b(n), p(n), q(n))$  we set  $(a(n, 0), b(n, 0), p(n, 0), q(n, 0)) = (a(n), b(n), p(n), q(n))$ , and denote by  $(a(n, l), b(n, l), p(n, l), q(n, l))$  the  $l$ th CF obtained by successive Bauer–Muir accelerations using  $(r(n, l), d(n, l))$ . We thus obtain **two-dimensional arrays**, all linked by **3-term linear recursions**, and hopefully with not too complicated coefficients (in particular, if possible, with  $(a(n, l), b(n, l), r(n, l), d(n, l))$  **polynomials** in  $n$  and  $l$ ).

The game then consists in **traveling** in this array as one likes. Ideal acceleration would be in traveling along the **diagonal**  $l = n$ : however, complicated coefficients. Instead, travel along a **staircase** either above or below the diagonal.



# Apéry Acceleration II

Everything explicit, except for the initial terms. The `cfapery` command returns the accelerated CF (and if desired the two-dimensional arrays), but since the initial terms may not be correct, the limit will be of the form  $(aS + b)/(cS + d)$  with small integers  $(a, b, c, d)$ , where  $S$  is the limit of the initial CF.

The command `fracdep` (analogous to `linddep` and `algdep`) finds these small integers, and it is then trivial to modify the CF output by `cfapery` to obtain one which converges to  $S$ .

# Apéry Acceleration II

Everything explicit, except for the initial terms. The `cfapery` command returns the accelerated CF (and if desired the two-dimensional arrays), but since the initial terms may not be correct, the limit will be of the form  $(aS + b)/(cS + d)$  with small integers  $(a, b, c, d)$ , where  $S$  is the limit of the initial CF.

The command `fracdep` (analogous to `linddep` and `algdep`) finds these small integers, and it is then trivial to modify the CF output by `cfapery` to obtain one which converges to  $S$ .

# Examples of the `cfapery` Command I

First, the “canonical” example:

```
? A = cfclostovec(cffromser(n->n^3,n->n^3))
% = [[0, 2*n^3 - 3*n^2 + 3*n - 1], [1, -n^6]]
/* Naive continued fraction for zeta(3),
   term by term from series. */
? B = cfsimplify(cfapery(A))
% = [[0, 34*n^3 - 51*n^2 + 27*n - 5], [6, -n^6]]
/* Apery works, but miraculously
   cfsimplify gives something simple */
? cflimit(B)-zeta(3)
% = 0.E-38 /* No need for fracdep */
? cftype(B)[2]
% = [0, 1153.999..., 0, 0]
/* Very fast convergence in (1+sqrt(2))^(−8*n) */
```

## Examples of the `cfapery` Command II

Here is a more complicated and spectacular example:

```
? ab=[[1/2,7*n-5],[1,-4*n*(3*n-2)]];cftype(ab)[2]
%= [0, 4/3, 0, 5/3]
/* Continued fraction for 2^(1/3), exponential convergence
   in C/(4/3)^nn^(5/3). Since 4/3 rational, try apery */
? AB=cfapery(ab)
%= [[1/2,6],[8*n,8*n+4]],
   [[1,-20],[-12*n^2+8*n,-12*n^2-32*n-20]]]
? cftype(AB)[2]
%= [0, -3, 0, 0]
/* Apery works, faster exp. conv. in C/(-3)^n */
? AB2=cfsimplify(AB)
%= [[1/2, 28, 40*n - 20], [8, -144*n^2 + 64]]
? AB2=cfmul(AB2,1/4)
%= [[1/2, 7, 10*n - 5], [2, -9*n^2 + 4]]
/* Simplification much simpler, exp. conv. in C/9^n */
```

## Examples of the `cfapery` Command II

Here is a more complicated and spectacular example:

```
? ab=[[1/2,7*n-5],[1,-4*n*(3*n-2)]];cftype(ab)[2]
%= [0, 4/3, 0, 5/3]
/* Continued fraction for 2^(1/3), exponential convergence
   in C/(4/3)^nn^(5/3). Since 4/3 rational, try apery */
? AB=cfapery(ab)
%= [[1/2,6],[8*n,8*n+4]],
   [[1,-20],[-12*n^2+8*n,-12*n^2-32*n-20]]]
? cftype(AB)[2]
%= [0, -3, 0, 0]
/* Apery works, faster exp. conv. in C/(-3)^n */
? AB2=cfsimplify(AB)
%= [[1/2, 28, 40*n - 20], [8, -144*n^2 + 64]]
? AB2=cfmul(AB2,1/4)
%= [[1/2, 7, 10*n - 5], [2, -9*n^2 + 4]]
/* Simplification much simpler, exp. conv. in C/9^n */
```

## Examples of the `cfapery` Command II

Here is a more complicated and spectacular example:

```
? ab=[[1/2,7*n-5],[1,-4*n*(3*n-2)]];cftype(ab)[2]
%= [0, 4/3, 0, 5/3]
/* Continued fraction for 2^(1/3), exponential convergence
   in C/(4/3)^nn^(5/3). Since 4/3 rational, try apery */
? AB=cfapery(ab)
%= [[1/2,6],[8*n,8*n+4]],
   [[1,-20],[-12*n^2+8*n,-12*n^2-32*n-20]]]
? cftype(AB)[2]
%= [0, -3, 0, 0]
/* Apery works, faster exp. conv. in C/(-3)^n */
? AB2=cfsimplify(AB)
%= [[1/2, 28, 40*n - 20], [8, -144*n^2 + 64]]
? AB2=cfmul(AB2,1/4)
%= [[1/2, 7, 10*n - 5], [2, -9*n^2 + 4]]
/* Simplification much simpler, exp. conv. in C/9^n */
```

## Example of the `cfapery` Command III

```
/* Since 9 rational, try apery again */
? AB3=cfapery(AB2)
% = [[ [1/2, 4], [18*n-9/2, 18*n+9/2] ],
      [ [2, 7/4], [-9*n^2+4, -9*n^2-9*n+7/4] ] ]
? cftype(AB3)[2]
% = [0, -33.97..., 0, 0]
/* Apery works again, even faster exp. conv. in
   C/(-(1+sqrt(2))^4)^n */
? AB4=cfmul(AB3,2)
% = [[ [1/2, 8], [36*n-9, 36*n+9] ],
      [ [4, 7], [-36*n^2+16, -36*n^2-36*n+7] ] ]
/* limit is a Mobius transform of 2^(1/3), we use
   fracdep to find it: */
```

# Example of the `cfapery` Command IV

```
? fracdep(cflimit(AB4),2^(1/3))
% = (-10*x - 3)/(12*x - 22)
? %+10/12
% = -32/(18*x - 33)
? cftochar(AB4)
% = "1/2+4/(8+7/27-..."
? -10/12-32/(-33+18*(1/2+4/(8+x)))
% = (x + 13)/(2*x + 10)
? %-1/2
% = 4/(x + 5)
/* Thus, we set */
? AB5=[[ [1/2,5], [36*n-9,36*n+9] ],
        [ [4,7], [-36*n^2+16,-36*n^2-36*n+7] ] ]
? cflimit(AB5)-2^(1/3)
% = 0.E-38
```



# The Dictionary I

Exploring the literature, using the package, making systematic searches, and applying systematically Apéry acceleration whenever possible, I have compiled a list of almost 300 continued fractions of **polynomial type** (i.e.,  $a(n)$  and  $b(n)$  polynomials of period 1 or 2 for  $n$  sufficiently large) with **rational** coefficients, both for specific real numbers, and for elementary and special functions. Several observations:

- With only 1 or 2 exceptions,  $b(n)$  is always a product of linear factors over  $\mathbb{Q}$  (or  $\mathbb{Q}(i)$  when  $z$  has been changed into  $iz$ ).
- There exist CF for  $\pi$ ,  $\pi/\sqrt{3}$ , and  $\pi/\sqrt{2}$ , but I know of no other e.g., for  $\pi/\sqrt{d}$  with  $d > 3$  squarefree.
- There exist CF for  $\pi^2$ ,  $L(\chi_{-4}, 2)$ , and  $L(\chi_{-3}, 2)$ , but no other “periods of degree 2” or divided by some  $\sqrt{d}$ .
- There exist only a few CF for “periods of degree 3” ( $\pi^3$ ,  $\pi^3/\sqrt{3}$ , and  $\zeta(3)$  and no others), and a single one of degree 4 for  $\pi^4$  that I found in 1980 with G. Rhin. None in higher degree.

# The Dictionary I

Exploring the literature, using the package, making systematic searches, and applying systematically Apéry acceleration whenever possible, I have compiled a list of almost 300 continued fractions of **polynomial type** (i.e.,  $a(n)$  and  $b(n)$  polynomials of period 1 or 2 for  $n$  sufficiently large) with **rational** coefficients, both for specific real numbers, and for elementary and special functions. Several observations:

- With only 1 or 2 exceptions,  $b(n)$  is always a product of linear factors over  $\mathbb{Q}$  (or  $\mathbb{Q}(i)$  when  $z$  has been changed into  $iz$ ).
- There exist CF for  $\pi$ ,  $\pi/\sqrt{3}$ , and  $\pi/\sqrt{2}$ , but I know of no other e.g., for  $\pi/\sqrt{d}$  with  $d > 3$  squarefree.
- There exist CF for  $\pi^2$ ,  $L(\chi_{-4}, 2)$ , and  $L(\chi_{-3}, 2)$ , but no other “periods of degree 2” or divided by some  $\sqrt{d}$ .
- There exist only a few CF for “periods of degree 3” ( $\pi^3$ ,  $\pi^3/\sqrt{3}$ , and  $\zeta(3)$  and no others), and a single one of degree 4 for  $\pi^4$  that I found in 1980 with G. Rhin. None in higher degree.



# The Dictionary I

Exploring the literature, using the package, making systematic searches, and applying systematically Apéry acceleration whenever possible, I have compiled a list of almost 300 continued fractions of **polynomial type** (i.e.,  $a(n)$  and  $b(n)$  polynomials of period 1 or 2 for  $n$  sufficiently large) with **rational** coefficients, both for specific real numbers, and for elementary and special functions. Several observations:

- With only 1 or 2 exceptions,  $b(n)$  is always a product of linear factors over  $\mathbb{Q}$  (or  $\mathbb{Q}(i)$  when  $z$  has been changed into  $iz$ ).
- There exist CF for  $\pi$ ,  $\pi/\sqrt{3}$ , and  $\pi/\sqrt{2}$ , but I know of no other e.g., for  $\pi/\sqrt{d}$  with  $d > 3$  squarefree.
- There exist CF for  $\pi^2$ ,  $L(\chi_{-4}, 2)$ , and  $L(\chi_{-3}, 2)$ , but no other “periods of degree 2” or divided by some  $\sqrt{d}$ .
- There exist only a few CF for “periods of degree 3” ( $\pi^3$ ,  $\pi^3/\sqrt{3}$ , and  $\zeta(3)$  and no others), and a single one of degree 4 for  $\pi^4$  that I found in 1980 with G. Rhin. None in higher degree.



# The Dictionary I

Exploring the literature, using the package, making systematic searches, and applying systematically Apéry acceleration whenever possible, I have compiled a list of almost 300 continued fractions of **polynomial type** (i.e.,  $a(n)$  and  $b(n)$  polynomials of period 1 or 2 for  $n$  sufficiently large) with **rational** coefficients, both for specific real numbers, and for elementary and special functions. Several observations:

- With only 1 or 2 exceptions,  $b(n)$  is always a product of linear factors over  $\mathbb{Q}$  (or  $\mathbb{Q}(i)$  when  $z$  has been changed into  $iz$ ).
- There exist CF for  $\pi$ ,  $\pi/\sqrt{3}$ , and  $\pi/\sqrt{2}$ , but I know of no other e.g., for  $\pi/\sqrt{d}$  with  $d > 3$  squarefree.
- There exist CF for  $\pi^2$ ,  $L(\chi_{-4}, 2)$ , and  $L(\chi_{-3}, 2)$ , but no other “periods of degree 2” or divided by some  $\sqrt{d}$ .
- There exist only a few CF for “periods of degree 3” ( $\pi^3$ ,  $\pi^3/\sqrt{3}$ , and  $\zeta(3)$  and no others), and a single one of degree 4 for  $\pi^4$  that I found in 1980 with G. Rhin. None in higher degree.



# The Dictionary I

Exploring the literature, using the package, making systematic searches, and applying systematically Apéry acceleration whenever possible, I have compiled a list of almost 300 continued fractions of **polynomial type** (i.e.,  $a(n)$  and  $b(n)$  polynomials of period 1 or 2 for  $n$  sufficiently large) with **rational** coefficients, both for specific real numbers, and for elementary and special functions. Several observations:

- With only 1 or 2 exceptions,  $b(n)$  is always a product of linear factors over  $\mathbb{Q}$  (or  $\mathbb{Q}(i)$  when  $z$  has been changed into  $iz$ ).
- There exist CF for  $\pi$ ,  $\pi/\sqrt{3}$ , and  $\pi/\sqrt{2}$ , but I know of no other e.g., for  $\pi/\sqrt{d}$  with  $d > 3$  squarefree.
- There exist CF for  $\pi^2$ ,  $L(\chi_{-4}, 2)$ , and  $L(\chi_{-3}, 2)$ , but no other “periods of degree 2” or divided by some  $\sqrt{d}$ .
- There exist only a few CF for “periods of degree 3” ( $\pi^3$ ,  $\pi^3/\sqrt{3}$ , and  $\zeta(3)$  and no others), and a single one of degree 4 for  $\pi^4$  that I found in 1980 with G. Rhin. None in higher degree.



**Challenge:** it is usually easy to guess numerically the constants  $C$  entering into the asymptotics. I have been unable to do so for the very first CF:

```
? A=[[1/2,7*n-5],[1,-4*n*(3*n-2)]]; cflimit(A)^3
% = 2.0000000000000000000000000000000000000000000000000000000 /* So A=2^(1/3) */
? cfasymp(A)[3]
% = [0, 4/3, 0, 5/3, 2.370194...]
/* Convergence in C/((4/3)^nn^(5/3) with C=2.37... */
/* What is C ? */
```