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

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

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

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

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:

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

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:

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

Continued Fractions III

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

Goals I

- 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 I

- 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 I

- 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 `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 II

There also exist a number of creators which create the necesssary 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\ge1}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\ge0}(-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\ge0}(-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(cfsubst([a,b],z,1))
% = 2.71828182845...
? cftype([a,b])
% = [[1, 4, 0, -1/2], [2, -16/z^2, 0, 0], 1]
```

The `cfsubst` 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).

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]
? lindep([-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 $0.342...$ 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]
? lindep([-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... 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=cfsubst([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
/* Lots of fractions, CF for exp(1/4) */
? ab5=cfmul(ab4,4)
% = [[1, 7, 16*n - 8], [2, 1]]
? cftochar(ab5)
% = "1+2/(7+1/(24+1/(40+1/(56+1/(72+1/(88))))))"
/* Much neater, and equivalent CF for exp(1/4) */
```

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

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

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 `cfsimplify` 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 `cfcontract` command can also do this on scalar-valued CF.
- The `cftype` 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 nth tail of a CF (which can be computed numerically by the command `cflimit([a,b],n)`) is

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

and that the limit S is given by

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

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 $(p(n+1)+r(n)p(n))/(q(n+1)+r(n)q(n))$ 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 nth tail of a CF (which can be computed numerically by the command `cflimit([a,b],n)`) is

$$\rho(n) = b(n)/(a(n{+}1){+}b(n{+}1)/(a(n{+}2){+}b(n{+}2)/(a(n{+}3){+}\cdots))) ,$$

and that the limit S is given by

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

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 $(p(n+1) + r(n)p(n))/(q(n+1) + r(n)q(n))$ 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:

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

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

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

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

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:

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

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

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

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

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 lth 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 lth 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 `lindep` 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 `lindep` 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 π, $\pi/\sqrt{3}$, and $\pi/\sqrt{2}$, but I know of no other e.g., for $\pi/\sqrt{d}$ with $d > 3$ squarefree.
- There exist CF for π^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" (π^3, $\pi^3/\sqrt{3}$, and $\zeta(3)$ and no others), and a single one of degree 4 for π^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 π, $\pi/\sqrt{3}$, and $\pi/\sqrt{2}$, but I know of no other e.g., for $\pi/\sqrt{d}$ with $d > 3$ squarefree.

• There exist CF for π^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" (π^3, $\pi^3/\sqrt{3}$, and $\zeta(3)$ and no others), and a single one of degree 4 for π^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 π, $\pi/\sqrt{3}$, and $\pi/\sqrt{2}$, but I know of no other e.g., for $\pi/\sqrt{d}$ with $d > 3$ squarefree.
- There exist CF for π^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" (π^3, $\pi^3/\sqrt{3}$, and $\zeta(3)$ and no others), and a single one of degree 4 for π^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 π, $\pi/\sqrt{3}$, and $\pi/\sqrt{2}$, but I know of no other e.g., for $\pi/\sqrt{d}$ with $d > 3$ squarefree.
- There exist CF for π^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" (π^3, $\pi^3/\sqrt{3}$, and $\zeta(3)$ and no others), and a single one of degree 4 for π^4 that I found in 1980 with G. Rhin. None in higher degree.

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 π, $\pi/\sqrt{3}$, and $\pi/\sqrt{2}$, but I know of no other e.g., for $\pi/\sqrt{d}$ with $d > 3$ squarefree.
- There exist CF for π^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" (π^3, $\pi^3/\sqrt{3}$, and $\zeta(3)$ and no others), and a single one of degree 4 for π^4 that I found in 1980 with G. Rhin. None in higher degree.

The Dictionary II

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
% = 2.000000000000000000000000000000000 /* 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 ? */
```