Automatic expansion of C-fractions for solutions of Riccati equations via Guessing and Proving.

The examples from Cuyt *et alii*., Handbook of Continued Fractions for Special Functions.

Maulat & Salvy

This worksheet demonstrates the use of the submodule gfun:-ContFrac, for power series solutions of Riccati differential equations : y'=p y^2 + q y + r with p,q,r rational in the variable z.

The first section gives a few examples, and demonstrates how to print some information along the way.
A comprehensive list of all the examples from Cuyt *et alii* is then given in the following section.

```
> restart;
> starttime := time();
> with(gfun):

  if gfun:-version() < 3.70 then
    error "Old gfun version. Please download the latest version on
  gfun's website, http://perso.ens-lyon.fr/bruno.
  salvy/software/the-gfun-package/."
  fi;

  with(gfun:-ContFrac):
```

First examples.

The procedure Riccati_to_Cfrac takes as input :
- f: a function to be expanded (or a series)
- y: a function name
- z: the variable name

and optionally :
- a number (the series expansion order)

```
> Riccati_to_Cfrac( exp(z), y, z );
```

The infolevel command enables to print details on the computation.
The 'demo' field concerns the main computation lines ; it can be set to 0 or 1.

```
> infolevel[demo]:=1;
> Riccati_to_Cfrac( tan(z), y, z, 20 );
```

```
> infolevel[demo]:=0;
```

More information on how the continued fraction expansion is computed and proved can be printed using the 'gfuncontfrac' information field.

```
> infolevel[gfuncontfrac]:=1;
```

```
> Riccati_to_Cfrac( tan(z), y, z, 20 );
```

```
> infolevel[demo]:=0;
  infolevel[gfuncontfrac]:=0;
```

C-fractions formulas of Riccati solutions in *Cuyt et alii*.

In this section, formulas from the compendium by Cuyt *et alii* (2008) are recovered automatically. All explicit C-fractions solutions of Riccati equations are recovered.

Formulas are stored in a table gfun:-ContFrac:-Cuyt_Cfrac, which is accessed as follows:

```
> chapter:=11;
```

```
> labels(chapter);
```

```
> Cuyt_Cfrac[chapter][1.3];
```

In what follows, only the parameters related to the series exponents are given example values.

Chapter 11

```
> chapter := 11;
```

```
> labels(chapter);
```

```
> N:=20;
```

Generic expansion, with finite series as input.
The order of the series is doubled when the C-fraction coefficients are in z^2.

```
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );
    f := Cuyt_Cfrac[chapter][formula]:

    if formula in {1.3, 2.2, 7.1, 7.2} then
      S := MultiSeries:-series(f,z,N):
    elif formula = 7.4 then
      S := series(f,z,2*N);
    else
      S := MultiSeries:-series(f,z,2*N)
     fi;

    Riccati_to_Cfrac( S, y, z );
```

```
   od;
> f := -(-2*z+(1+z)^2-1)/((1+z)^2-1);
> factor(f);
```

Chapter 12

```
> chapter := 12;
> N:=25:
```

The parameter l in formulas (12.6.21) and (12.6.22) is set to a consistent value.

```
> vall:=3;
> infolevel[gfuncontfrac]:=0;
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );
    f := Cuyt_Cfrac[chapter][formula]:

    if formula in {6.21,6.22} then
      print( f, l=vall );
      f := subs( l=vall, f );
     fi;

    Riccati_to_Cfrac( MultiSeries:-series(f,z,N), y, z )

  od:
```

Chapter 13

In this chapter, one parameter (k) is given a value, and some variable changes are performed for simplicity.

```
> chapter:=13;
> labels(chapter);
> valk := 3;
> N:=34:
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );

    f := Cuyt_Cfrac[chapter][formula];

    if formula = 3.5 then   f := subs(k=valk,f);
    elif formula = 4.9 then  f := subs(z=z/sqrt(I*Pi),f)
    end if;

    print(f);
```

```
    S := MultiSeries:-series(f,z,N);

    Riccati_to_Cfrac( S, y, z )

  od:
```

Chapter 14

```
> chapter := 14;
> N:=20;
> labels(chapter);
```

Parameters n and nu are given example values.

```
> valn := 4;
  valnu := 3/2;
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );

    f := Cuyt_Cfrac[chapter][formula];
    print(f);

    if formula = 1.16 then
      S := MultiSeries:-series(f,z,N) assuming nu::real;
    elif formula = 1.20 then
      S := MultiSeries:-series(f,z,N) assuming 1-nu > 0;
    else
      S := MultiSeries:-series(f,z,N);
    fi;

    Riccati_to_Cfrac( S, y, z )

  od:
```

Chapter 15

Chapter 15 contains parametrized hypergeometric ratios, which are long to compute with.
For this reason, information is printed along the computation, and additional arguments are used, as detailed below.

```
> chapter := 15;
> N:=20;
> labels(chapter);
> infolevel[demo]:=1;
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );
```

```
    f := Cuyt_Cfrac[chapter][formula];

    print(f);

    S := series(f,z,N);

    Riccati_to_Cfrac( S, y, z, 15 );

  od:
> infolevel[demo]:=0;
```

Chapter 16

```
> chapter := 16;
> N:=20;
> labels(chapter);
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );

    f := Cuyt_Cfrac[chapter][formula];

    print(f);

    S := series(f,z,N);

    Riccati_to_Cfrac( S, y, z );

  od:
```

Chapter 17

In this chapter, some functions require to compute the series by hand, as well as the Riccati equation.
The formulas are still produced and proved automatically.

```
> chapter := 17;
> labels(chapter);
> N:=20;
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );

    f := subs(n=nn, Cuyt_Cfrac[chapter][formula]);
```

```
  if formula = 1.44 then

    lprint("Ad-hoc series computation, and Riccati equation
computation.");
      f := subs(z=z/I,f/exp(-I*Pi));
     print(f);
      f := eval(f, HankelH1 = ContFrac:-myHankelH1);
      f := simplify(f,power,symbolic,exp);
      S := map(simplify,MultiSeries:-series( f, z, N ));

      deq := diff(y(z), z) - (z^2+(2*nu+2)*z^2*y(z)-y(z)^2)
/z^3:
      Riccati_to_Cfrac(  S, y, z, input_riccati = deq );

  elif formula = 1.46 then

    lprint("Ad-hoc series computation.");
      f := subs(z=I*z,f);
     print(f);
      f := eval(f, HankelH2 = ContFrac:-myHankelH2);
      f := simplify(f,power,symbolic,exp);
      S := map(simplify,MultiSeries:-series( f, z, N ));

      Riccati_to_Cfrac(  S, y, z );

  elif formula = 2.35 then

    print(f);
    S := map(factor,series(f,z,N));
    Riccati_to_Cfrac(  S, y, z );

  else

     print(f);
    S := MultiSeries:-series( f, z, N );
    Riccati_to_Cfrac(  S, y, z );

   fi;

od:
```

Chapter 18

```
> chapter := 18;
> labels(chapter);
> N:=40;
```

```
> alpha := 2;
> for formula in labels(chapter) do

    print( sprintf("%a.%a",chapter,formula) );

     f := Cuyt_Cfrac[chapter][formula];
    print(f);
     S := MultiSeries:-series( f, z, N );
     Riccati_to_Cfrac(  S, y, z );

  od:
```

Other examples.

logarithmic derivative of Airy function, at infinity

An indication that this function is of interest appears in the Chudnovsky brothers article of 1991.

We perform a slight linear fractional transformation to remove the first terms, and prove the formula.

```
> f:=AiryAi(x);
> S:=MultiSeries:-asympt(f,x,100):
> dS:=diff(S,x):
> dlog:=asympt(dS/S,x,100):
> St0 := subs(x=1/t^2,dlog):
> series( t*St0 ,t,90):
> St := eval(%,csgn=1):
```

Formula

```
> numtheory[cfrac](St,t,10,simregular):
> numtheory[cfrac](%,t,simregular,quotients);
```

Ad hoc transformation to remove the first terms.

```
> S:=eval( series(1+t^3/4/(1+St),t,90), t=z^(1/3)):
> gfun:-ContFrac:-Riccati_to_Cfrac(subs(t=z,S),y,z,10);
```

dlmf 4.25.5

```
> f:= exp(2*A*arctan(1/z));
> A:=2/3;
> g:='g';
> g:=subs(z=1/z, solve(f=1+2*A/(z-A+g(z)),g(z)));
> G := map(normal,series(g,z,40)):
> gfun:-ContFrac:-Riccati_to_Cfrac(G,y,z,20);
>
```

```
> time()-starttime;
```