SYMBOLIC COMPUTATION OF FORMAL POWER SERIES WITH MACSYMA

WOLFRAM KOEPF

Fachbereich Mathematik, Freie Universität Berlin
Arnimallee 3, W-1000 Berlin 33

ABSTRACT

In this lecture we show how to calculate formal power series expansions of analytic functions, and to manipulate general formal power series using MACSYMA.

1. MACSYMA's Built-in Capabilities

For the purpose to calculate formal power series expansions of analytic functions, MACSYMA[5] provides the command **powerseries**($expr,x,x_0$) that calculates the power series expansion for *expr* with respect to the variable x at the point of development x_0. Examples are

```
(C1) powerseries(exp(x),x,0);
                                    INF
                                    ====    I1
                                    \      X
(D1)                                 >     ---
                                    /      I1!
                                    ====
                                    I1 = 0
(C2) powerseries(sin(x),x,0);
                             INF
                             ====        I2  2 I2 + 1
                             \      (- 1)   X
(D2)                          >     -----------------
                             /         (2 I2 + 1)!
                             ====
                             I2 = 0
(C3) powerseries(1/(1-x)^alpha,x,0);
                     INF
                     ====
                     \                                 I3  I3
(D3)                  >     BINOMIAL(- ALPHA, I3) (- 1)   X
                     /
                     ====
                     I3 = 0
```

```
(C4) powerseries(sin(x+y),x,0);

        INF                              INF
        ====         I4  2 I4            ====         I4  2 I4 + 1
        \      (- 1)   X                 \      (- 1)   X
(D4)   ( >     -------------) SIN(Y) + ( >     -----------------) COS(Y)
        /         (2 I4)!                /         (2 I4 + 1)!
        ====                             ====
        I4 = 0                           I4 = 0
```

While the last result is not exactly what we requested, i.e. the power series expansion of a sum (namely the simplified expression $\sin(x+y) = \cos x \sin y + \sin x \cos y$) rather than the sum of two power series expansions, and cannot be brought into a single power series object by Macsyma's simplifiers, this is apparently a reasonable result as it represents the function by its odd and even parts (with respect to x). The situation is much more inappropriate in the example

```
(C5) powerseries(exp(x)*sin(x),x,0);

                              INF         INF
                              ====    I5  ====        I5  2 I5 + 1
                              \      X    \      (- 1)   X
(D5)                         ( >     ---)  >     -----------------
                              /      I5!  /         (2 I5 + 1)!
                              ====        ====
                              I5 = 0      I5 = 0
```

where the product of two power series expansions is returned rather than the power series expansion of the product as requested. The Macsyma implementation of **powerseries**(f,x,x_0) is not based on an algorithm but uses a chain of certain heuristic steps which gives the result in some cases. Macsyma's procedure is at follows:

1. Macsyma tries to expand f in the variable $x-x_0$, e.g. using addition theorems,
2. logarithms $\ln f$ are handled by the rule $\ln f = \int \frac{f'}{f}$,
3. for rational functions a (real nonalgebraic) partial fraction decomposition is used,
4. the power series expansions of the standard elementary functions with point of development $x_0 = 0$ are implemented.

This procedure has the following disadvantages: It fails

1. to find the result for all rational functions, e.g. for $f := \frac{1}{x^2+2x+2}$, because of the use of a *real* partial fraction decomposition,

2. to find the result for $f := \frac{1}{x^2-2x-2}$, e.g., as the partial fraction implementation fails to find nonrational roots of the denominator,

3. to solve `powerseries(exp(x)*exp(y),x,0)` as the internal simplifier changes the input into e^{x+y} before processing,

4. to get `powerseries(atan(x+a),x,0)` or `powerseries(atan(x),x,b)` because of the lack of an addition formula for the inverse tangent function,

5. to solve the problem for products correctly. Usually as above a product of power series is returned rather than the power series of the product as requested.

2. Algorithmic Approach

In four papers we developed an algorithmic approach for the given and connected problems[1–4]. We implemented this algorithm in the symbolic algebra system MATHEMATICA[7], and a MAPLE[6] implementation was done by Dominik Gruntz. In this section we will give some step-by-step examples for calculations of *exp-like* and *rational type*[1] using MACSYMA.

First we present a MACSYMA session searching for the power series of the expression `f:exp(x)*sin(x)`. The underlying idea of the method presented can be generalized to many other examples. We try first to find a second order homogeneous, linear differential equation with polynomial coefficients for f.

```
(C1) f:exp(x)*sin(x);
                                       X
(D1)                                 %E  SIN(X)

(C2) fp:diff(f,x);
                                X            X
(D2)                          %E  SIN(X) + %E  COS(X)

(C3) fpp:diff(fp,x);
                                        X
(D3)                                2 %E  COS(X)

(C4) eq:fpp+q[1]*fp+q[0]*f;

                X            X                X              X
(D4)      Q  (%E  SIN(X) + %E  COS(X)) + Q  %E  SIN(X) + 2 %E  COS(X)
           1                              0

(C5) solve([q[1]+q[0]=0,q[1]+2=0],[q[1],q[0]]);

(D5)                           [[Q  = - 2, Q  = 2]]
                                  1         0
```

By setting the coefficients of the rationally independent functions $e^x \sin x$ and $e^x \cos x$ to zero, we find that f is a solution of the differential equation

$$y'' - 2y' + 2y = 0 \; .$$

It is now a reasonable idea to use MACSYMA's **ode** command with the **series** option to solve this differential equation getting a series solution. However, this method does not apply for the examples of this section.

So we proceed trying to find a recurrence equation for the coefficients. Therefore we set $y = \sum_{n=0}^{\infty} \frac{b_n}{n!} x^n$ which is an appropriate choice (and will lead to a recurrence equation similar to the differential equation) if the differential equation has constant coefficients. We get

```
(C6) y:sum(b[n]*x^n/n!,n,0,inf);
                                   INF       N
                                   ====   B  X
                                   \       N
(D6)                                >    -----
                                   /       N!
                                   ====
                                   N = 0

(C7) diff(y,x);
                                INF         N - 1
                                ====  N B  X
                                \        N
(D7)                             >    -----------
                                /          N!
                                ====
                                N = 0

(C8) yp:changevar(%,k=n-1,k,n);
                            INF                   K
                            ====     (K + 1) B      X
                            \                 K + 1
(D8)                         >      -----------------
                            /           (K + 1)!
                            ====
                            K = - 1

(C9) diff(y,x,2)$

(C10) ypp:changevar(%,k=n-2,k,n);
                         INF       2                  K
                         ====     (K  + 3 K + 2) B      X
                         \                        K + 2
(D10)                     >      ------------------------
                         /               (K + 2)!
                         ====
                         K = - 2
```

Now the left-hand side of the differential equation reads

```
(C11) ypp-2*yp+2*y;

         INF       N   INF       2                   K
         ====  B  X    ====    (K  + 3 K + 2) B      X
         \      N      \                      K + 2
(D11) 2   >    ----- +  >      ------------------------
         /      N!     /               (K + 2)!
         ====          ====
         N = 0         K = - 2
                                                               INF                     K
                                                               ====    (K + 1) B      X
                                                               \                K + 1
                                                           - 2  >      -----------------
                                                               /           (K + 1)!
                                                               ====
                                                               K = - 1
(C12) intosum(%);

       INF         N   INF       2                   K
       ====  2 B  X    ====    (K  + 3 K + 2) B      X
       \       N       \                      K + 2
(D12)   >    ------- +  >      ------------------------
       /       N!      /               (K + 2)!
       ====            ====
       N = 0           K = - 2
                                                         INF                      K
                                                         ====       2 (K + 1) B      X
                                                         \                     K + 1
                                                       +  >      (- -------------------)
                                                         /                (K + 1)!
                                                         ====
                                                         K = - 1
(C13) sumcontract(%);

         INF      2                    N                      N         N
         ====   (N  + 3 N + 2) B      X    2 (N + 1) B      X    2 B  X
         \                      N + 2                  N + 1        N
(D13)     >    (------------------------ - ------------------- + -------)
         /              (N + 2)!               (N + 1)!          N!
         ====
         N = 0
```

As the right-hand side of the differential equation is zero, all coefficients of this power series must equal zero. This leads to

```
(C14) coeff(part(%,1),x,n)=0;
```

```
                  2
              (N   + 3 N + 2) B          2 (N + 1) B         2 B
                              N + 2                N + 1        N
(D14)         --------------------- - ---------------- + ---- = 0
                    (N + 2)!              (N + 1)!          N!

(C15) factcomb(%);
                          B         2 B         2 B
                           N + 2     N + 1       N
(D15)                     ------ - -------- + ---- = 0
                            N!        N!        N!

(C16) eq1:ratsimp(%*n!);
(D16)                     B      - 2 B      + 2 B  = 0
                           N + 2      N + 1      N
```

so that we get a constant coefficient recurrence equation for b_n. For this kind of recurrence equations the setup $b_n := t^n$ leads to a solution.

```
(C17) define(b[n],t^n);
                                                N
(D17)                                    B  := T
                                          N

(C18) eq2:ev(eq1);
                           N + 2      N + 1      N
(D18)                     T      - 2 T      + 2 T  = 0

(C19) ratsimp(%/t^n);
                                2
(D19)                          T  - 2 T + 2 = 0

(C20) sol:solve(%,t);
(D20)                      [T = 1 - %I, T = %I + 1]
```

Thus we find two complex solutions for the recurrence equation which is linear, so each linear combination gives also a solution. The special solution to our problem is derived by introducing the initial values. In `(D24)` below, we know the result should be real (because f is real), and **rectform** is used to obtain the real result (and simultaneously verifies that the result is real!).

```
(C21) remarray(b)$

(C22) define(b[n],a*(rhs(first(sol)))^n+b*(rhs(last(sol)))^n);

                                      N              N
(D22)                    B  := (%I + 1)  B + (1 - %I)  A
                          N

(C23) solve([b[0]=subst(x=0,f),b[1]=subst(x=0,fp)],[a,b]);
```

```
                                %I        %I
(D23)                      [[A = --, B = - --]]
                                2         2

(C24) subst(%,b[n]);
                              N                 N
                         (1 - %I)  %I   %I (%I + 1)
(D24)                    ------------ - ------------
                              2              2

(C25) result:rctform(%);
                              N/2     %PI N
(D25)                        2    SIN(-----)
                                        4

(C26) remarray(b)$

(C27) define(b[n],result);
                                N/2     %PI N
(D27)                      B  := 2    SIN(-----)
                            N               4

(C28) b[n]$

(C29) ev(y);
                           INF    N/2     %PI N   N
                           ====  2    SIN(-----) X
                           \                4
(D29)                       >    ------------------
                           /            N!
                           ====
                           N = 0
```

Line `(C27)` defines the array `b` as a pointer pointing to **define**'s second argument. Only the invocation `(C28)` gives the symbol $b[n]$ for the special argument n the desired value that can be used for the evaluation `(C29)`. These evaluation schemes are considered in another lecture.

Also for some rational functions **powerseries** fails to obtain an expansion. But finding the complex partial fraction decomposition and using the binomial series will always lead to the desired result. An example is

```
(C1) f:1/(x^2+2*x+2);
                                    1
(D1)                           ------------
                                2
                               X  + 2 X + 2

(C2) powerseries(f,x,0);
(D2)                         Unable to expand
```

The next procedure calculates the complex partial fraction decomposition of f.

```
(C3) gfactor(f);
                                      1
(D3)                     -------------------------
                         (X - %I + 1) (X + %I + 1)

(C4) f:partfrac(%,x);
                            %I               %I
(D4)                  -------------- - --------------
                      2 (X + %I + 1)   2 (X - %I + 1)
```

Next we ascertain if MACSYMA is able to find the power series of f in the above form.

```
(C5) powerseries(f,x,0);
(D5)                          Unable to expand
```

As this is not the case, we handle the two summands separately.

```
(C6) powerseries(first(f),x,0)+powerseries(last(f),x,0);

                 INF                         INF
                 ====        I3  I3          ====        I4  I4
                 \       (- 1)   X           \       (- 1)   X
              %I  >      --------------   %I  >      --------------
                 /              I3 + 1       /              I4 + 1
                 ====    (%I + 1)            ====    (1 - %I)
                 I3 = 0                      I4 = 0
(D6)          ------------------------ - ------------------------
                          2                          2

(C7) sol:sumcontract(intosum(%));

                  INF
                  ====             I4  I4              I4  I4
                  \         %I (- 1)   X       %I (- 1)   X
(D7)               >      (---------------- - ----------------)
                  /                I4 + 1              I4 + 1
                  ====     2 (%I + 1)         2 (1 - %I)
                  I4 = 0
(C8) rectform(sol);

Is   X   positive or negative?
p;
                 INF     - I4 - 1
                 ====    --------
                 \          2          I4     %PI (I4 + 1)   I4
(D8)              >     2         (- 1)   SIN(------------) X
                 /                                  4
                 ====
                 I4 = 0
```

If a power series expansion at the origin exists, it is valid in some open interval containing the origin. So there is no loss of generality to introduce x to be a positive variable at MACSYMA's request.

Another example is the arctan function whose derivative is rational and the same procedure applies.

```
(C1) f:atan(x-x0);
(D1)                          - ATAN(X0 - X)

(C2) powerseries(f,x,0);
(D2)                           Unable to expand

(C3) fp:diff(f,x);
                                      1
(D3)                          -------------
                                        2
                               (X0 - X)  + 1

(C4) gfactor(fp);
                                      1
(D4)                   ---------------------------
                       (X0 - X - %I) (X0 - X + %I)

(C5) fp:partfrac(%,x);
                             %I                  %I
(D5)                ----------------- - -----------------
                    2 (- X0 + X + %I)   2 (- X0 + X - %I)

(C6) powerseries(fp,x,0);
(D6)                           Unable to expand

(C7) powerseries(first(fp),x,0)+powerseries(last(fp),x,0);

               INF                           INF
               ====          I3  I3          ====            I4  I4
               \        (- 1)   X            \          (- 1)   X
            %I  >      ---------------    %I  >      -----------------
               /              I3 + 1         /                I4 + 1
               ====   (%I - X0)              ====   (- X0 - %I)
               I3 = 0                        I4 = 0
(D7)        ------------------------- - ---------------------------
                         2                            2

(C8) sumcontract(intosum(%));
                INF
                ====             I3  I3               I3  I3
                \         %I (- 1)   X          %I (- 1)   X
(D8)             >      (----------------- - -------------------)
                /                 I3 + 1                 I3 + 1
                ====     2 (%I - X0)         2 (- X0 - %I)
                I3 = 0
```

```
(C9) sol:-atan(x0)+integrate(%,x);

     INF
     ====                I3   I3 + 1                     I3   I3 + 1
     \           %I (- 1)    X                   %I (- 1)    X
(D9)  >     (------------------------- - ---------------------------)
     /                          I3 + 1                           I3 + 1
     ====    2 (I3 + 1) (%I - X0)         2 (I3 + 1) (- X0 - %I)
     I3 = 0
                                                                     - ATAN(X0)
(C10) rectform(sol);
Is   X   positive or negative?
p;
                                        - I3 - 1
                                        --------
      INF          I3   I3 + 1    2        2                          %PI
      ====   (- 1)    X        (X0  + 1)         SIN((I3 + 1) (ATAN(X0) + ---))
      \                                                                    2
(D10)  >     ----------------------------------------------------------------
      /                                  I3 + 1
      ====
      I3 = 0
                                                                     - ATAN(X0)
(C11) changevar(%,i4=i3+1,i4,i3);

            INF          I4   I4      2 I4 ATAN(X0) + %PI I4
            ====    (- 1)    X    SIN(----------------------)
            \                                  2
(D11)     -  >      --------------------------------------- - ATAN(X0)
            /                           2     I4/2
            ====               I4 (X0   + 1)
            I4 = 1
```

We present another example finding the power series expansion $\sum_{n=0}^{\infty} a_n x^n$ for $f(x) = \frac{x}{x^2-x-1}$, and showing that the coefficients a_n fulfill the recurrence equation

$$a_n = a_{n-1} + a_{n-2} \quad (n \geq 2)$$

and the initial conditions

$$a_0 = 0 \ ,$$

$$a_1 = a_2 = 1 \ ,$$

and so are the *Fibonacci numbers*. We then calculate the 1000^{th} Fibonacci number a_{1000} using the coefficient formula found. As the **gfactor** command does not apply with nonrational factors, we find the factors of the denominator by the **solve** command. Further, as the **powerseries** command does not apply for the summands that are the output of **partfrac**, we calculate the partial fraction decomposition ourselves.

The following MACSYMA code solves the given example:

```
f:x/(1-x-x^2);
solve(denom(f),x);
f1:a/(x-rhs(first(%)))+b/(x-rhs(last(%)));
ratsimp(f1/f);
expr:%*2*x;
solve([coeff(expr,x,0)=0,coeff(expr,x,1)=2],[a,b]);
f:subst(%,f1);
                     SQRT(5) - 5           SQRT(5) + 5
                 -------------------- - --------------------
                         SQRT(5) - 1            SQRT(5) + 1
                 10 (X - -----------)   10 (X + -----------)
                              2                      2
powerseries(first(f),x,0)+powerseries(last(f),x,0);
sol:sumcontract(intosum(%));
coeff(part(%,1),x,i2);
subst(i2=n,%);
                                N  N                          N  N
             (- SQRT(5) - 5) (- 1)  2    (SQRT(5) - 5) (- 1)  2
             ------------------------- + -----------------------
                            N + 1                       N + 1
               5 (SQRT(5) + 1)            5 (1 - SQRT(5))

/* calculate fib(1000) */
ratsimp(subst(n=1000,%));

434665576869374564356885276750406258025646605173717804024817290895365554179490#

189040387984007925516929592259308032263477520968962323987332247116164299644090#

5331879382989696499285160037044761377951668492288 75

/* find the recurrence relation */
fp:diff(f,x);
eq:fp+q*f;
solve(eq,q);
                                      2
                                     X  + 1
                               [Q = -----------]
                                     3    2
                                    X  + X  - X
y:sum(a[n]*x^n,n,0,inf);
yp:diff(y,x);
(x^3+x^2-x)*yp+(x^2+1)*y;
expand(%);
summ:intosum(%);
f1:changevar(part(summ,1),k=n+2,k,n);
f2:changevar(part(summ,2),k=n+2,k,n);
f3:changevar(part(summ,3),k=n+1,k,n);
f4:part(summ,4);
f5:part(summ,5);
sumcontract(f1+f2+f3+f4+f5);
```

```
part(%,1);
coeff(part(%,1),x,n)=0;
solve(%,a[n]);
                                      [A  = A      + A     ]
                                        N    N - 1    N - 2
```

3. Formal Power Series of the Hypergeometric Type

In the last section we developed a method with which we were able to extend MACSYMA's capabilities to find the not truncated power series expansions of certain analytic functions. Here we proceed with this matter in extending the described method to a very large class of hypergeometric type functions[1], what this exactly means, will be discussed later. As a motivation we generate a MACSYMA session searching for the power series development of the arccos function. Note that MACSYMA was not able to expand `powerseries(acos(x),x,0)` (until Version 415), even if MACSYMA was able to get the correct answer for `powerseries(asin(x),x,0)` (and $\operatorname{acos} x = \pi/2 - \operatorname{asin} x$). The arccos function is an especially nice example to show what's going on. We shall introduce the necessary new steps when they appear.

We begin to search again for a second order linear, homogeneous differential equation with polynomial coefficients for $f := \operatorname{acos} x$.

```
(C1) f:acos(x);
(D1)                                ACOS(X)

(C2) fp:diff(f,x);
                                         1
(D2)                              - ------------
                                              2
                                    SQRT(1 - X )
(C3) fpp:diff(fp,x);
                                         X
(D3)                              - -----------
                                          2 3/2
                                    (1 - X )
```

Obviously there will no differential equation exist in which f explicitly occurs, so we look for one between f' and f''.

```
(C4) fpp+q*fp;
                                   Q              X
(D4)                       - ------------ - -----------
                                        2            2 3/2
                             SQRT(1 - X )   (1 - X )

(C5) sol:solve(%=0,q);
```

```
                                        X
(D5)                             [Q = ------]
                                       2
                                      X  - 1
```

The above procedure showed that f is a solution of the differential equation

$$(x^2 - 1)\, y'' + xy' = 0 \; .$$

The MACSYMA built-in function **ode** with the **series** option is able to solve this differential equation — as other hypergeometric type equations — having the desired series as output, and it remains to use the initial conditions to eliminate the two generated constants (this is only true until version 416. From version 417 on the **series** package tries to find a closed form solution, and does not give a series output for the examples of this section.). On the other hand, this output contains Pochhammer symbols, and cannot be brought into a simpler form.

Therefore — and also because this approach will give us insight about the general situation — we proceed in translating this differential equation into a recurrence equation for the coefficients a_n ($n \in \mathbb{N}$) of f as before.

```
(C6) y:sum(a[n]*x^n,n,0,inf)$

(C7) yp:diff(y,x)$

(C8) ypp:diff(yp,x)$
```

Substituting these expression into the left hand side of the differential equation yields

```
(C9) (x^2-1)*(ypp+q*yp);
                         INF                   INF
                         ====                  ====
           2             \            N - 1    \                   N - 2
(D9)     (X   - 1) (Q     >     N A  X       +  >     (N - 1) N A  X      )
                         /         N           /                  N
                         ====                  ====
                         N = 0                 N = 0

(C10) ratsimp(subst(sol,%));

               INF                         INF
               ====                        ====
               \            N - 1      2   \       2           N - 2
(D10)        X  >     N A  X       + (X  - 1)  >     (N  - N) A  X
               /         N                 /                  N
               ====                        ====
               N = 0                       N = 0

(C11) sum:intosum(expand(%));
```

```
      INF             INF
      ====            ====
      \           N   \      2   2     N - 2         N - 2
(D11)  >    N A  X  +  >    X  (N  A  X       - N A  X     )
      /        N      /             N             N
      ====            ====
      N = 0           N = 0
                                                INF
                                                ====
                                                \            N - 2    2     N - 2
                                              +  >    (N A  X       - N  A  X     )
                                                /        N              N
                                                ====
                                                N = 0
(C12) f1:part(sum,1)$

(C13) f2:part(sum,2)$

(C14) f3:changevar(part(sum,3),k=n-2,k,n)$

(C15) sumcontract(f1+f2+f3)$

(C16) expand(%);
        INF
        ====
        \        2         N               N             N    2     N
(D16)    >    (- N  A      X  - 3 N A      X  - 2 A      X  + N  A  X )
        /            N + 2           N + 2         N + 2          N
        ====
        N = 0
```

This finally gives the left-hand side of the differential equation in power series form. As the right-hand side equals zero, all coefficients must vanish.

```
(C17) coeff(part(%,1),x,n)=0;

                   2                                    2
(D17)           - N  A       - 3 N A      - 2 A      + N  A  = 0
                      N + 2         N + 2       N + 2      N

(C18) solve(%,a[n+2]);
                                               2
                                             N  A
                                                 N
(D18)                             [A       = ------------]
                                    N + 2    2
                                            N  + 3 N + 2

(C19) recursion:part(%,1)/a[n];
```

```
         A              2
          N + 2          N
(D19)    ------ = -----------
           A        2
            N      N  + 3 N + 2
```

Until now the procedure was as in the last section. Now come the new aspects. The point is that there is a procedure for solving the resulting recurrence equation that is always successful if the recurrence equation is of the *hypergeometric type*, i.e., of the form

$$\frac{a_{n+m}}{a_n} = R(n) \ ,$$

where $m \in \mathbb{N}$ is a positive integer, and R is a rational function in the variable n. On the other hand, many examples in applications are of the hypergeometric type.

Suppose first $m = 1$, the ordinary hypergeometric case. Then $\frac{a_{n+1}}{a_n} = R$ is rational in n (so that the ratio test for convergence applies). It is now essential to factor R. Then we get

$$\frac{a_{n+1}}{a_n} = c \cdot \frac{(n+\alpha_1)\cdot(n+\alpha_2)\cdots(n+\alpha_p)}{(n+\beta_1)\cdot(n+\beta_2)\cdots(n+\beta_q)} \ . \tag{1}$$

It is easily established that there is the following explicit representation for the coefficients

$$a_n = c^n \frac{(\alpha_1)_n \cdot (\alpha_2)_n \cdots (\alpha_p)_n}{(\beta_1)_n \cdot (\beta_2)_n \cdots (\beta_q)_n} \cdot a_0 \ , \tag{2}$$

where $(a)_n$ denotes the *Pochhammer symbol* or *shifted factorial* defined by

$$(a)_n := \begin{cases} 1 \ , & \text{if } n = 0 \\ a(a+1)(a+2)\cdots(a+n-1) \ , & \text{otherwise} \end{cases} \ .$$

This can be proved just by substituting the explicit formula Eq. 2 into Eq. 1 and noting that Eq. 1 given a_0 has a unique solution.

If $m > 1$, then one decomposes f by m shifted series

$$f(x) = \sum_{j=0}^{m-1} \left(\sum_{k=0}^{\infty} a_{mk+j} \, x^{mk+j} \right) = \sum_{j=0}^{m-1} x^j \, f_j(x^m)$$

with functions f_j $(j = 0, \ldots, m-1)$ each of which is ordinarily hypergeometric, so that their coefficients can be found by an application of Eq. 2.

Let us now proceed with our MACSYMA session. We calculate the first few coefficients from the values of the function f (defined in `(C1)`) and its derivatives by Taylor's Theorem: $a_n = \frac{f^{(n)}(0)}{n!}$.

```
(C20) a[0]:subst(x=0,f)/0!;
                                          %PI
(D20)                                     ---
                                           2
```

```
(C21) a[1]:subst(x=0,fp)/1!;
(D21)                                 - 1

(C22) a[2]:subst(x=0,fpp)/2!;
(D22)                                  0
```

As in the given case $m = 2$, and since $a_2 = 0$, it turns out that all even coefficients a_{2k} ($k \in \mathbb{N}$) except a_0 vanish, and so this proves that $f - f(0)$ is odd (without the use of a built-in simplification rule like $\operatorname{acos}(-x) \to \pi - \operatorname{acos} x$). In that case it will be more convenient to work with the power series

$$h(x) := \sum_{k=0}^{\infty} c_k x^k$$

for which $f(x) - f(0) = x h(x^2)$, and so $c_k = a_{2k+1}$ $(k \in \mathbb{N})$. To introduce the new variables, we substitute $2k + 1$ for n. This yields

```
(C23) c[0]:a[1]$

(C24) subst(n=2*k+1,recursion);

                    A                               2
                     2 K + 3             (2 K + 1)
(D24)               -------- = ----------------------------
                    A                    2
                     2 K + 1   (2 K + 1)  + 3 (2 K + 1) + 2
```

We introduce c_k $(k > 0)$:

```
(C25) c[k+1]/c[k]=factor(rhs(%));

                          C                           2
                           K + 1           (2 K + 1)
(D25)                     ------ = -------------------
                            C      2 (K + 1) (2 K + 3)
                             K

(C26) rat:rhs(%);
                                               2
                                      (2 K + 1)
(D26)                            -------------------
                                 2 (K + 1) (2 K + 3)

(C27) solnum:solve(num(rat),k);
                                              1
(D27)                                 [K = - -]
                                              2

(C28) multiplicities[1];
(D28)                                  2
```

```
(C29) solden:solve(denom(rat),k);
                                            3
(D29)                             [K = - -, K = - 1]
                                            2
(C30) leadcoeff:coeff(expand(num(rat)),k,2)/coeff(expand(denom(rat)),k,2),eval;

(D30)                                        1
```

The above calculations with the aid of Eq. 2 lead to the Pochhammer representation of c_k

$$c_k = \frac{\left(\frac{1}{2}\right)_k \cdot \left(\frac{1}{2}\right)_k}{\left(\frac{3}{2}\right)_k \cdot (1)_k} \cdot c_0 \ .$$

What will now be important, is to find ways to simplify expressions like this to bring them in a more familiar form using factorials. We have implemented those simplification rules for the function Pochhammers $(a, k) := (a)_k$ in the file `pochhammer.mac` which we load into our session. (We would have liked to use the MACSYMA built-in function `pochhammer(a,k)`, but from Version 417 on all occurrences of `pochhammer` are replaced by the Gamma function, and our results would neither look very nice nor familiar.) The content of this file will be discussed later.

```
(C31) load("pochhammer.mac")$

(C32) Pochhammers(1/2,k)^2/(Pochhammers(3/2,k)*k!)*c[0];
                                                  2
                                          (2 K)!
(D32)                             - -----------------
                                      K   2
                                     4  K!  (2 K + 1)!

(C33) res:factcomb(%*(2*k+1))/(2*k+1);
                                            (2 K)!
(D33)                               - ----------------
                                                   K   2
                                        (2 K + 1) 4  K!
(C34) remarray(c)$

(C35) define(c[k],res)$

(C36) c[k]$

(C37) h:sum(c[k]*z^k,k,0,inf);
                                  INF
                                  ====             K
                                  \        (2 K)! Z
(D37)                           -  >    ----------------
                                  /                K   2
                                  ====  (2 K + 1) 4  K!
                                  K = 0
```

```
(C38) declare(x,complex)$

(C39) a[0]+x*subst(z=x^2,h);
                               INF
                               ====             2 K
                        %PI    \        (2 K)! X
(D39)                   --- - X  >    ----------------
                         2      /               K   2
                               ====  (2 K + 1) 4  K!
                               K = 0

(C40) f:intosum(%);
                        INF
                        ====            2 K + 1
                        \        (2 K)! X            %PI
(D40)                    >    (- ----------------) + ---
                        /                 K   2       2
                        ====     (2 K + 1) 4  K!
                        K = 0
```

This is the desired power series expansion for the arccos function.

Now we explain how we implemented the simplification rules for the Pochhammer symbol. This is most easily done by pattern matching rules. First we implement the rule Pochhammers $(a, 0) \to 1$ which should hold for arbitrary a. This is done by the following MACSYMA code.

```
matchdeclare(arbitrary_arg,true);

matchdeclare(zero_arg,check_zero);
check_zero(x):=is(x=0);

/* Pochhammers(a,0) -> 1 */
tellsimp(Pochhammers(arbitrary_arg,zero_arg),1);
```

Next, the Pochhammer symbol Pochhammers (n, k) can be easily represented using factorials if $n \in \mathbb{N}$. We suppose here and in the further declarations k to be arbitrary even though MACSYMA substitutes the Gamma function for the factorials whereas the Pochhammer symbol is not defined in the case that k is not a nonnegative integer. But as we intend to use the Pochhammer symbols only for the coefficients of power series with nonnegative integer coefficients that are indexed by a formal variable, we prefer doing so rather than declaring k and all other occurring indices as integers.

```
matchdeclare(pos_int_arg,check_integer);
check_integer(x):=integerp(x) and is(x>0);

/* Pochhammers(n,k) -> (n+k-1)!/(n-1)! */
tellsimp(Pochhammers(pos_int_arg,arbitrary_arg),
        (pos_int_arg+arbitrary_arg-1)!/(pos_int_arg-1)!);
```

Now a more sophisticated rule is used. We saw in our example that it may be of some importance to have a rule for Pochhammers (a,k) for numbers a that differ by $1/2$ from an integer. Therefore observe that for $b \neq 0$ and $k \in \mathbb{N}$

$$\begin{aligned}
(2b)_{2k} &= 2b(2b+1)(2b+2)\cdots(2b+2k-1) \\
&= 2b(2b+2)(2b+4)\cdots(2b+2k-2)\cdot(2b+1)(2b+3)(2b+5)\cdots(2b+2k-1) \\
&= 2^k b(b+1)(b+2)\cdots(b+k-1)\cdot 2^k\left(b+\frac{1}{2}\right)\left(b+\frac{3}{2}\right)\left(b+\frac{5}{2}\right)\cdots\left(b+\frac{1}{2}+k-1\right) \\
&= 4^k (b)_k \left(b+\frac{1}{2}\right)_k ,
\end{aligned}$$

so that we get

$$\left(b+\frac{1}{2}\right)_k = \frac{(2b)_{2k}}{4^k\cdot(b)_k} ,$$

and by the substitution $a := b + \frac{1}{2}$

$$(a)_k = \frac{(2a-1)_{2k}}{4^k\cdot\left(\frac{2a-1}{2}\right)_k} .$$

This is implemented by the MACSYMA code

```
matchdeclare(half_int_arg,check_half);
check_half(x):=integerp(x+1/2) and is(x>1);

/* Pochhammers(a,k) -> Pochhammers(2a-1,2k)/(4^k*Pochhammers((2a-1)/2,k) */
tellsimp(Pochhammers(half_int_arg,arbitrary_arg),
        Pochhammers(2*half_int_arg-1,2*arbitrary_arg)/
        (4^arbitrary_arg*Pochhammers((2*half_int_arg-1)/2,arbitrary_arg))
);
```

Note that the above formula holds only for $b \neq 0$ and so for $a \neq \frac{1}{2}$. For this value we have

$$\left(\frac{1}{2}\right)_k = \frac{1}{2}\cdot\frac{3}{2}\cdots\frac{2k-1}{2} = \frac{1}{2^k}\cdot\frac{(2k)!}{2\cdot 4\cdots 2k} = \frac{(2k)!}{4^k\cdot k!} ,$$

which is implemented by the rule

```
matchdeclare(onehalf_arg,check_onehalf);
check_onehalf(x):=is(x=1/2);

/* Pochhammers(1/2,k) -> (2k)!/(4^k*k!) */
tellsimp(Pochhammers(onehalf_arg,arbitrary_arg),
        (2*arbitrary_arg)!/(4^arbitrary_arg*arbitrary_arg!));
```

For negative numbers that differ by $1/2$ from an integer we use the rule

```
matchdeclare(half_int_arg_neg,check_half_neg);
check_half_neg(x):=integerp(x+1/2) and is(x<1);

tellsimp(Pochhammers(half_int_arg_neg,arbitrary_arg),
        product(half_int_arg_neg+j-1,j,1,entier(-half_int_arg_neg))*
        Pochhammers(half_int_arg_neg+entier(-half_int_arg_neg)+1,
        arbitrary_arg-entier(-half_int_arg_neg)-1)
);
```

Last but not least we implement the rule Pochhammers$(a, k) \to a(a+1)\cdots(a+k-1)$ for $k \in \mathbb{N}$ by

```
/* Pochhammers(a,k) -> a(a+1)...(a+k-1) */
tellsimp(Pochhammers(arbitrary_arg,pos_int_arg),
        product(arbitrary_arg+j-1,j,1,pos_int_arg));
```

Many functions in applications are of the hypergeometric type, so that the given procedure often is applicable. Here are some more examples. We give only the important parts of MACSYMA's output.

Example 1. By the following MACSYMA commands we find the power series expansion of the error function erf x with respect to x at the point $x = 0$.

```
/* exercise  erf(x) */
f:erf(x);
powerseries(f,x,0);
fp:diff(f,x);
fpp:diff(fp,x);
deq:ypp+p*yp;
fpp+p*fp;
solve(%=0,p);
deq:subst(%,deq);
                                  YPP + 2 X YP
y:sum(a[n]*x^n,n,0,inf);
yp:diff(y,x);
ypp:diff(yp,x);
ev(deq);
intosum(%);
part(%,1)+changevar(part(%,2),k=n-2,k,n);
sumcontract(%);
coeff(part(%,1),x,n)=0;
solve(%,a[n+2]);
                                         2 N A
                                              N
                          [A      = - ------------]
                            N + 2      2
                                      N  + 3 N + 2
recursion:part(%,1)/a[n];
a[0]:subst(x=0,f)/0!;
a[1]:subst(x=0,fp)/1!;
```

```
c[0]:a[1];
subst(n=2*k+1,recursion);
c[k+1]/c[k]=rhs(%);
factor(%);
                          C
                           K + 1          2 K + 1
                          ------ = - -----------------
                            C        (K + 1) (2 K + 3)
                             K
rat:rhs(%);
solnum:solve(num(rat),k);
solden:solve(denom(rat),k);
leadcoeff:coeff(expand(num(rat)),k,1)/coeff(expand(denom(rat)),k,2),eval;
load("pochhammer.mac");
(-1)^k*Pochhammers(1/2,k)/(Pochhammers(3/2,k)*k!)*c[0];
res:factcomb(%*(2*k+1))/(2*k+1);
remarray(c);
define(c[k],res);
c[k]$
h:sum(c[k]*z^k,k,0,inf);
declare(x,complex);
a[0]+x*subst(z=x^2,h);
f:intosum(%);
                          INF
                          ====           K  2 K + 1
                          \       2 (- 1)  X
                           >    ----------------------
                          /     SQRT(%PI) (2 K + 1) K!
                          ====
                          K = 0
```

Example 2. We find the power series expansion of $e^{x^2}\operatorname{erf} x$ with respect to x at the point $x = 0$.

```
/* exercise  exp(x^2)*erf(x) */
f:exp(x^2)*erf(x);
powerseries(f,x,0);
fp:diff(f,x);
fpp:diff(fp,x);
deq:ypp+q*yp+p*y;
fpp+q*fp+p*f;
sol:num(ratsimp(%));
solve([part(sol,1,1)=0,part(sol,2,2)=0],[p,q]);
deq:subst(%,deq);
                               YPP - 2 X YP - 2 Y
y:sum(a[n]*x^n,n,0,inf);
yp:diff(y,x);
ypp:diff(yp,x);
ev(deq);
intosum(%);
part(%,1)+part(%,2)+changevar(part(%,3),k=n-2,k,n);
```

```
sumcontract(intosum(%));
coeff(part(%,1),x,n)=0;
solve(%,a[n+2]);
                                        2 A
                                           N
                              [A       = -----]
                                N + 2   N + 2
recursion:part(%,1)/a[n];
a[0]:subst(x=0,f)/0!;
a[1]:subst(x=0,fp)/0!;
c[0]:a[1];
subst(n=2*k+1,recursion);
c[k+1]/c[k]=rhs(%);
                              C
                               K + 1      2
                              ------ = -------
                                C      2 K + 3
                                 K
load("pochhammer.mac");
Pochhammers(1,k)/(Pochhammers(3/2,k)*k!)*c[0];
res:factcomb(%*(2*k+1))/(2*k+1);
remarray(c);
define(c[k],res);
c[k]$
h:sum(c[k]*z^k,k,0,inf);
declare(x,complex);
a[0]+x*subst(z=x^2,h);
f:intosum(%);
                         INF
                         ====       K     2 K + 1
                         \       2 4  K! X
                          >    --------------------
                         /     SQRT(%PI) (2 K + 1)!
                         ====
                         K = 0
```

Example 3. We find the power series expansion for $f(x) := \frac{e^x \sinh x}{x}$.

```
/* exercise  e^x*sinh(x)/x  */
f:exp(x)*sinh(x)/x;
powerseries(f,x,0);
fp:diff(f,x);
fpp:diff(fp,x);
deq:ypp+q*yp+p*y;
fpp+q*fp+p*f;
sol:num(ratsimp(%));
solve([part(sol,1,1)=0,part(sol,2,1)=0],[p,q]);
subst(%,deq);
deq:ratsimp(%*x);
                         X YPP + (2 - 2 X) YP - 2 Y
y:sum(a[n]*x^n,n,0,inf);
```

```
yp:diff(y,x);
ypp:diff(yp,x);
ev(deq);
expand(%);
intosum(%);
sum:intosum(expand(%));
f1:part(sum,1);
f2:part(sum,2);
f3:changevar(part(sum,3),k=n-1,k,n);
f4:changevar(part(sum,4),k=n-1,k,n);
sumcontract(f1+f2+f3+f4);
coeff(part(%,1),x,n)=0;
solve(%,a[n+1]);
                                             2 A
                                                N
                                  [A       = -----]
                                    N + 1    N + 2
recursion:part(%,1)/a[n];
a[0]:limit(f,x,0);
load("pochhammer.mac");
result:2^n*Pochhammers(1,n)/(Pochhammers(2,n)*n!)*a[0];
                                          N
                                         2
                                      --------
                                      (N + 1)!
```

References

1. W. Koepf, *Power series in Computer Algebra* (Journal of Symbolic Computation **13**, 1992), p. 581–603.

2. W. Koepf, *Algorithmic development of power series* (Proceedings of the *Conference on artificial intelligence and symbolic computations*, Karlsruhe, Germany, August 1992. Lecture Notes in Computer Science, Springer), to appear.

3. W. Koepf, *Examples for the algorithmic calculation of formal Puiseux, Laurent and power series* (SIGSAM Bulletin **27**, 1993), p. 20–32.

4. W. Koepf, *A new algorithm for the development of algebraic functions in Puiseux series* (Preprint B-92-21 Fachbereich Mathematik der Freien Universität Berlin, 1992), 1–16.

5. Macsyma: *Reference Manual, Version 13* (Symbolics, USA, 1988).

6. Maple: *Reference Manual, fifth edition* (Watcom publications, Waterloo, 1988).

7. St. Wolfram, Mathematica. *A system for doing mathematics by Computer* (Addison-Wesley Publ. Comp., Redwood City, CA, 1991).