Full Partial Fraction Decomposition of Rational Functions

Manuel Bronstein
Wissenschaftliches Rechnen
ETH - Zentrum
CH-8092 Zürich, Switzerland
bronstein@inf.ethz.ch

Bruno Salvy
Algorithms Project
INRIA
F-78153 Le Chesnay Cedex, France
Bruno.Salvy@inria.fr

Abstract

We describe a rational algorithm that computes the full partial fraction expansion of a rational function over the algebraic closure of its field of definition. The algorithm uses only gcd operations over the initial field but the resulting decomposition is expressed with linear denominators. We give examples from its **Axiom** and **Maple** implementations.

Introduction

The partial fraction decomposition of a rational function is a form where both the local and global behaviour of the function are easy to find. This is used when computing a primitive by hand, or any linear operation which is most easily done on a pole. An example is the efficient computation of asymptotic expansion of the solutions of a linear recurrence with constant coefficients [4].

Let $f = A/D$ be a rational function in some field $K(z)$. By the fundamental theorem of algebra, it is clear that f admits a partial fraction decomposition of the form

$$f = P + \sum_{D(\alpha)=0} \sum_{i=1}^{n_\alpha} \frac{b_{\alpha,i}}{(z-\alpha)^i}, \tag{1}$$

where P is a polynomial in $K[z]$ and n_α is the multiplicity of the pole α. Because the actual computation of the numerical values of the singularities α is difficult, it is customary to use various partial forms of this expression where the denominators in the right-hand side are not required to be of degree 1.

In general, the numbers α involved in (1) belong to the algebraic closure of K. What we show in this paper is that it is possible to compute the $b_{\alpha,i}$ in (1) without any factorisation, and using only the operations of the field K. We describe our algorithm in Section 1, provide several examples of its use in Section 2, and conclude with a note on its use for symbolic integration.

ACM-ISSAC '93-7/93/Kiev, Ukraine
© 1993 ACM 0-89791-604-2/93/0007/0157...$1.50

1 The Algorithm

The algorithm is based on the following result of [1], which gives a formula for $b_{\alpha,i}$ as a polynomial in α. While this was used in [1] to compute polynomials whose roots were the $b_{\alpha,i}$'s, we use it here to compute the expansion (1) entirely rationally (the algorithm for partial fraction expansion presented in [1] had to compute with the $b_{\alpha,i}$'s).

Theorem 1 *Let K be a field of characteristic 0 and z be an indeterminate over K. Let $f \in K(z)$, D be its denominator, $D = D_1 D_2^2 \cdots D_m^m$ be a squarefree factorisation of D, and $H \in K[z]$ be any nontrivial monic squarefree factor of D_n for $n > 0$. Then, using only rational operations over K, we can compute $H_1, \ldots, H_n \in K[z]$ such that for $H(\alpha) = 0$, the $b_{\alpha,i}$'s of (1) satisfy $b_{\alpha,i} = H_i(\alpha)$.*

The proof of this result is contained in the proof of Theorem 1 of [1]. We describe here the explicit construction of the H_i's: let u be a differential indeterminate over $K(z)$, *i.e.* u and its formal derivatives $u', u'', \ldots$ are all independent variables over $K(z)$, and write

$$h = \frac{fH^n}{u^n} \in K(z)\langle u \rangle. \tag{2}$$

For $0 \le j < n$, compute $h^{(j)}/j!$ and write it as

$$\frac{h^{(j)}}{j!} = \frac{P_j(z, u, u', \ldots, u^{(j)})}{Q_j(z, u)} \tag{3}$$

where P_j and Q_j are multivariate polynomials over K. Substituting for u and its derivatives, compute the following polynomials:

$$\tilde{P}_j = P_j(z, H', \frac{H''}{2}, \ldots, \frac{H^{(j)}}{j}) \in K[z] \tag{4}$$

$$\tilde{Q}_j = Q_j(z, H') \in K[z]. \tag{5}$$

Let α be any root of H in the algebraic closure of K. It was shown in [1] that $\gcd(\tilde{Q}_j, H) = 1$, hence that $\tilde{Q}_j(\alpha) \neq 0$, and that

$$b_{\alpha,n-j} = \frac{\tilde{P}_j(\alpha)}{\tilde{Q}_j(\alpha)}. \tag{6}$$

Since $\gcd(\tilde{Q}_j, H) = 1$, we use the extended Euclidean algorithm in $K[z]$ to compute polynomials B_j and C_j satisfying

$$B_j\tilde{Q}_j + C_jH = 1. \tag{7}$$

We then have $b_{\alpha,n-j} = H_{n-j}(\alpha)$ where

$$H_{n-j} = \tilde{P}_jB_j \bmod H. \tag{8}$$

We note that is is possible for $b_{\alpha,n-j}$ to be 0 in formula (6), but if and only if $\tilde{P}_j(\alpha) = 0$, which means that $\tilde{P}_j$ and H have a nontrivial gcd, and that α is a root of that gcd. Our algorithm takes advantage of this fact to return the decomposition (1) in a form where all the terms in the sum are nonzero. We proceed as follows: given $f = A/D \in K(z)$ with $\gcd(A, D) = 1$ and D monic, we compute first the squarefree factorisation $D_1D_2^2 \ldots D_m^m$ of D. Then, for each D_k, we compute h by formula (2) with $H = D_k$ and $n = k$. For $0 \leq j < k$, as we compute the successive derivatives of h, we obtain $\tilde{P}_j$ and $\tilde{Q}_j$ by formulas (3,4,5). At this point we compute $G_j = \gcd(\tilde{P}_j, D_k)$ and replace D_k by $D_{kj} = D_k/G_j$. Since any root α of D_k is either a root of G_j or of D_{kj}, and $b_{\alpha,k-j} = 0$ iff $G_j(\alpha) = 0$, we replace H by D_{kj} in the computation of H_{k-j} by formulas (7,8). Thus, each j yields a summand of the form

$$\sigma_j = \sum_{D_{kj}(\alpha)=0} \frac{H_{k-j}(\alpha)}{(z-\alpha)^{k-j}} \tag{9}$$

and the full partial fraction decomposition (1) is obtained by adding to the polynomial part of f, all the σ_j's contributed by each D_k over all the k's. Of course, the sum in equation (9) is represented as a formal sum, for example as the triple $(k-j, D_{kj}, H_{k-j})$, so algebraic numbers are not introduced in the computation.

2 Examples and Applications

Suppose that $f = A/D$ has only simple poles, *i.e.* that D is squarefree. Then, formula (2) with $H = D$ and $n = 1$ yields $h = A/u$, so applying formulas (3) to (6) we get $P_0 = A = \tilde{P}_0$, $Q_0 = u$ so $\tilde{Q}_0 = D'$ and $b_{\alpha,1} = A(\alpha)/D'(\alpha)$. This is of course the classical formula [5] for the residues of f, so the partial fraction expansion of f is

$$f = (A \text{ quo } D) + \sum_{D(\alpha)=0} \frac{A(\alpha)/D'(\alpha)}{z-\alpha}.$$

The following example with multiple poles illustrate the lazy factoring feature of our algorithm: even though we only compute the squarefree factorisation of the denominator, the presence of zero terms in the expansion gives further splittings of the squarefree factors. Consider

$$f = \frac{36}{z^5 - 2z^4 - 2z^3 + 4z^2 + z - 2}.$$

The squarefree factorisation of the denominator of f is

$$D = (z-2)(z^2-1)^2 = D_1D_2^2.$$

Taking $H = D_1 = z - 2$ and $n = 1$, we get:

1. $h = f(z-2)/u = 36/((z^2-1)^2u)$,
2. $H' = 1$,
3. $\tilde{P}_0 = \text{numer}(h) = 36$ so $G_0 = \gcd(\tilde{P}_0, H) = 1$,
4. $\tilde{Q}_0 = \text{eval}(\text{denom}(h), u \to H') = (z^2-1)^2$,
5. $H^* = H/G_0 = z - 2$,
6. $\frac{1}{9}\tilde{Q} - \frac{1}{9}(z^3 + 2z^2 + 2z + 4)H^* = 1$, so $B_0 = 1/9$,
7. $H_1 = 36\,B_0 = 4$, so $\sigma_0 = \sum_{\alpha-2=0} 4/(z-\alpha) = 4/(z-2)$.

Taking $H = D_2 = z^2 - 2$ and $n = 2$, we get:

1. $h = f(z^2-1)^2/u^2 = 36/((z-2)u^2)$,
2. $H' = 2z$,
3. $\tilde{P}_0 = \text{numer}(h) = 36$ so $G_0 = \gcd(\tilde{P}_0, z^2-1) = 1$,
4. $\tilde{Q}_0 = \text{eval}(\text{denom}(h), u \to H') = 4z^2(z-2)$,
5. $H^* = H/G_0 = z^2 - 1$,
6. $(-\frac{z}{12} - \frac{1}{6})\tilde{Q}_0 + (\frac{z^2}{3} - 1)H^* = 1$, so $B_0 = -z/12 - 1/6$,
7. $H_2 = 36\,(-z/12 - 1/6) = -3z - 6$, so $\sigma_0 = \sum_{\alpha^2-1=0}(-3\alpha-6)/(z-\alpha)^2$,
8. $h' = ((-72z + 144)u' - 36u)/((z-2)^2u^3)$,
9. $H''/2 = 1$,
10. $\tilde{P}_1 = \text{eval}(\text{numer}(h'), u \to H', u' \to H''/2) = -144z + 144$, so $G_1 = \gcd(\tilde{P}_1, z^2-1) = z - 1$,
11. $\tilde{Q}_1 = \text{eval}(\text{denom}(h'), u \to H') = 8z^3(z-2)^2$,

12. $H^* = H/G_1 = (z^2-1)/(z-1) = z+1$,

13. $-\frac{1}{72}\tilde{Q}_1 + (\frac{z^4}{9} - \frac{5z^3}{9} + z^2 - z + 1)H^* = 1$,
so $B_1 = -1/72$,

14. $H_1 = (-144z + 144)\, B_1 \bmod (z+1) = -4$,
so $\sigma_1 = \sum_{\alpha+1=0} -4/(z-\alpha) = \sigma - 4/(z+1)$.

Thus, the full partial fraction decomposition of f is

$$\frac{36}{z^5 - 2z^4 - 2z^3 + 4z^2 + z - 2} = \frac{4}{z-2} - \frac{4}{z+1} - 3\sum_{\alpha^2-1=0} \frac{\alpha+2}{(z-\alpha)^2}.$$

This is exactly the form of the answer returned by our programs. In **Maple** :

```
> f;

                          36
           -------------------------------
            5      4      3      2
           x  - 2 x  - 2 x  + 4 x  + x - 2

> fullparfrac(f,x);

                       / -----              \
     4       4         |  \          - 3 alpha - 6|
   ----- - ----- +     |   )         -------------|
   x - 2   x + 1       |  /                     2|
                       | -----        (x - alpha) |
                       \alpha = %1                /

                                                  2
%1 :=             RootOf(_Z  - 1)
```

And in **Axiom** :

```
(11) -> f

                          36
   (11)   -----------------------------
           5     4     3     2
          x  - 2x  - 2x  + 4x  + x - 2

Type: Fraction UnivariatePolynomial(x,
                              Fraction Integer)

(12) -> fullPartialFraction f

           4       4         --+       - 3%A - 6
   (12)  ----- - ----- +     >         ---------
         x - 2   x + 1       --+               2
                           2           (x - %A)
                        %A  - 1= 0
```

An application of this full partial fraction decomposition is the resolution of linear recurrences with rational coefficients and initial conditions. If u_n is a solution of such an equation, the generating function $\sum_n u_n z^n$ is a rational function of z. The coefficients of the partial fraction decomposition are the coefficients of the basis of exponential polynomial solutions found by the characteristic polynomial method. One can use this approach to compute efficiently an asymptotic estimate of these solutions without actually computing the roots of the polynomial. This is the subject of [4].

Our algorithm can also be applied in the reverse direction: algorithm 4.1 of [2] computes a hypergeometric recurrence relation from a rational function. The first step of that algorithm is "Compute a **complex** PFD of f ...which can be algorithmically done at least if the denominator has a rational factorisation". Since the remaining steps of algorithm 4.1 can be carried on a formal sum of the form (9), we allow that algorithm to run on any rational function.

Another application is the integration of rational functions. Our algorithm makes the classical partial fraction integration algorithm rational: it produces a sum of terms, each of the form (9), so the integral of the initial rational function is the sum of the integrals of the individual terms. Each term is easily integrated:

$$\int \sum_{Q(\alpha)=0} \frac{H(\alpha)}{(z-\alpha)^m} dz =$$

$$\begin{cases} \sum_{Q(\alpha)=0} \frac{1}{1-m}\frac{H(\alpha)}{(z-\alpha)^{m-1}}, & \text{if } m > 1 \\ \sum_{Q(\alpha)=0} H(\alpha)\log(z-\alpha), & \text{if } m = 1 \end{cases}$$

and as previously, the summands of $\int f(z)dz$ are represented as formal sums, so no algebraic numbers are introduced. While this approach gives a non-factoring algorithm for integration, it yields the full partial fraction decomposition of the answer, so we do not suggest it as a replacement for the other rational integration algorithms (Ostrogradski, Hermite, Horowitz, Mack, Trager, Lazard & Rioboo). It is however interesting that after all those newer algorithms, the original local approach, which dates back to Newton, can be made rational.

Finally, another application area for our algorithm which has yet to be investigated is the search for Liouvillian solutions of linear ordinary differential equations. Kovacic's algorithm [3] for solving the differential equation $y''(z) + r(z)y(z) = 0$ requires the full partial fraction expansion of $r(z)$ over the complex numbers. Kovacic's necessary conditions for the 3$^{\text{rd}}$ case of his algorithm can be tested easily using our algorithm (an alternative test based on the same idea is in [1]). Whether the rest of his algorithm can be made rational, and whether the efficient computation of full partial fraction expansions can improve the algorithms for higher order equations are still open problems.

References

[1] M. Bronstein, *Formulas for Series Computation*, Applied Algebra in Engineering, Communication and Computing **2**, 195–206 (1992).

[2] W. Koepf, *Power Series in Computer Algebra*, Journal of Symbolic Computation **13**, 581–603 (1992).

[3] J. J. Kovacic, *An Algorithm for Solving Second Order Linear Homogeneous Differential Equations*, Journal of Symbolic Computation **2**, 3–43 (1986).

[4] X. Gourdon & B. Salvy, *Asymptotics of Linear Recurrences with Rational Coefficients*, manuscript submitted to the 5^{th} Formal Power Series and Algebraic Combinatorics Conference, Florence, 1993.

[5] B. Trager, *Algebraic Factoring and Rational Function Integration*, in Proceedings SYMSAC '76, 219–226 (1976).