Algorithms for the Functional Decomposition of Laurent Polynomials

Stephen M. Watt

University of Western Ontario, London, Ontario, Canada n6a 5b7
`http://www.csd.uwo.ca/~watt`

Abstract. Recent work has detailed the conditions under which univariate Laurent polynomials have functional decompositions. This paper presents algorithms to compute such univariate Laurent polynomial decompositions efficiently and gives their multivariate generalization.

One application of functional decomposition of Laurent polynomials is the functional decomposition of so-called "symbolic polynomials." These are polynomial-like objects whose exponents are themselves integer-valued polynomials rather than integers. The algebraic independence of X, X^n, $X^{n^2/2}$, *etc*, and some elementary results on integer-valued polynomials allow problems with symbolic polynomials to be reduced to problems with multivariate Laurent polynomials. Hence we are interested in the functional decomposition of these objects.

1 Introduction

Determining whether a univariate polynomial may be written as the functional composition of others of lower degree is a question that has been studied for almost a century. Ritt [1] considered the case of polynomials with complex coefficients and showed the decomposition factors and their degrees are unique up to certain transformations. Engstrom [2] and Levi [3] generalized Ritt's results, showing they hold for arbitrary fields of characteristic zero.

Polynomial decompositions can be useful because they reveal the structure of a problem. This may allow certain problems to be solved explicitly that otherwise could not be. Decomposable polynomials of a given degree form a low-dimensional subspace of the space of all polynomials of that degree. A polynomial that is the composition of two others of degrees r and s has degree rs, but instead of requiring $rs + 1$ coefficients to describe, it can be specified by the $r + s$ independent coefficients of its composition factors.

Algorithms by Barton and Zippel [4] and more recently by Kozen and Landau [5] have been incorporated in many computer algebra systems. Generalizations have been studied for functional decomposition of rational functions [6], algebraic functions [7] and multivariate polynomials [8]. More recent work by Zieve [9] has shown the conditions under which univariate Laurent polynomials may be decomposed, and gives results analogous to those of Ritt.

Separately, we have been interested in the problem of reasoning about and performing algebraic operations on families of polynomials parameterized their exponents [10–12]. This work explores algorithms that work in the generic case, and can be specialized uniformly by evaluating the exponent parameters. Other work considers case-based structure [13–15].

As defined more precisely in [10, 11], the so-called "symbolic polynomials" resemble ordinary polynomials with exponents that are integer-valued polynomials. For example, $X^{(n^2-n)/2} - X^{2nm}Y^m - 4$ would belong to a particular ring of symbolic polynomials. Taking the integer-valued polynomials as an abelian group gives the symbolic polynomials an obvious group ring structure. Using the fact that integer-valued polynomials have integer coefficients when written in a binomial basis (in this example, $\binom{n}{i}\binom{m}{j}$ for $i, j \geq 0$) and on the algebraic independence of the polynomial variables raised to different monomial powers (in this example $X, X^n, X^{\binom{n}{2}}, X^{nm}, Y^m$), it is possible to reduce many problems on symbolic polynomial to problems on multivariate Laurent polynomials.

Most recently, the problem of functional decomposition of symbolic polynomials has been studied, and reduced to the functional decomposition of multivariate Laurent polynomials [16]. In this article we now explore the algorithmic aspects of finding such functional decompositions of Laurent polynomials. We present two algorithms for univariate Laurent polynomial decomposition: one that reduces the problem to polynomial decomposition and one that solves the problem directly. We also present their multivariate generalization.

The paper is organized as follows: First, Section 2 gives some initial definitions and notations. Then Section 3 presents the decomposition problem for univariate Laurent polynomials. We note that the important case, from an algorithms point of view, is when a Laurent polynomial f decomposes as $f = g \circ h$ with g a polynomial and h a Laurent polynomial. The main body of the paper is devoted to showing how to compute such decompositions. The first method uses the leading and trailing coefficients of f to find the leading and trailing coefficients of h. Section 4 gives the required mathematical justification for the method and Section 5 gives the algorithm. This method has the advantage that it may be implemented using an existing polynomial decomposition library, but it has the disadvantage that it may require trying multiple candidate values for h. The second method avoids this problem and determines the coefficients of h from a single triangular system involving the leading coefficients of f. Section 6 gives the mathematical background for the method and Section 7 presents the algorithm. The multivariate generalization of these methods is discussed in Section 8 and Section 9 concludes the paper.

2 Preliminaries

We begin by establishing certain notation and conventions we use throughout.

Notation 1 (Univariate Laurent polynomials) *For a ring R R, we denote by $R[(X)]$ the ring of Laurent polynomials $R[X, X^{-1}]/\langle XX^{-1} - 1\rangle$.*

Notation 2 (Multivariate Laurent polynomials) *For a ring R R, we denote by $R[(X_1,\ldots,X_n)]$ the ring of multivariate Laurent polynomials*

$$R[X_1,\ldots,X_n,X_1^{-1},\ldots,X_n^{-1}]/\langle X_1X_1^{-1}-1,\ldots,X_nX_n^{-1}-1\rangle.$$

Notation 3 (Coefficient) *Given $f\in R[(X)]$, we denote the coefficient of X^k in f as $[X^k]f$ or f_k.*

Notation 4 (Multiplication time) *We denote by $M(m,n)$ the time to multiply polynomials of degrees m and n. If $m=n$, we write $M(n)$.*

Definition 1 (Degree of univariate Laurent polynomial).
Let $h\in R[(X)]$ be a Laurent polynomial with a pole of order t at 0 and of order s at ∞. Then the degree *of h is $\deg h=\langle -t,s\rangle$.*

Definition 2 (Degree of multivariate Laurent polynomial).
Let $h\in R[(X_1,\ldots,X_v)]$ be a Laurent polynomial with poles in X_i of order t_i at 0 and of order s_i at ∞. Then the degree *of h is $\deg h=\langle(-t_1,\ldots,-t_v),(s_1,\ldots,s_v)\rangle$.*

Definition 3 (Total degree of multivariate Laurent polynomial).
Let $h=\sum_i c_iX_1^{e_{1i}}\cdots X_n^{e_{ni}}\in R[(X_1,\ldots,X_n)]$ and $w\in\mathbb{Z}^n_{>0}$. Then the total degree *of h with* weight vector *w is $\operatorname{tdeg}_w\ h=\max_i\sum_{j=1}^n e_{ji}w_j$. If no weight vector is specified, then $w=(1,\ldots,1)$ is assumed.*

Convention 1 (Empty sequence) *The sequence $h_a,\ldots,h_b$ is empty if $b<a$.*

3 Univariate Decomposition

We phrase the functional decomposition problem for univariate Laurent polynomials over a field K as follows:

Problem 1 (Univariate Laurent polynomial decomposition).
Given $f\in K[(X)]$, K a field, and $r\geq 2\in\mathbb{Z}$, do there exist $g\in K[X]$ of degree r and $h\in K[(X)]$ such that $f=g\circ h$? If so find such g and h.

We justify below why we consider $g\in K[X]$ and $h\in K[(X)]$ as opposed to $g,h\in K(X)$ or $g,h\in K[(X)]$.

For the discussion in later sections we fix the following: We let $\deg f=\langle -rt,rs\rangle$. Supposing g and h exist, we let

$$g=\sum_{i=0}^{r}g_iX^i,\quad h=\sum_{i=-t}^{s}h_iX^i\quad f=\sum_{i=-rt}^{rs}f_iX^i\,. \tag{1}$$

We place certain conditions on r, s and t to concentrate on the problem of interest. We assume $t>0$ since otherwise $f,g,h\in K[X]$ and we have the usual polynomial decomposition. We require the inverse of r in K. In the following, we let $\ell=s+t$ and $N=\ell r$. Then h has $\ell+1$ coefficients, g has $r+1$ and f has $N+1$.

We now discuss our restriction that $g\in K[X],h\in K[(X)]$. This relates to the ways in which a Laurent polynomial may decompose. The following result of Zieve [9] describes the situation when $K=\mathbb{C}$.

Lemma 1 (Zieve). *For $f \in \mathbb{C}(X)\backslash\mathbb{C}$, the fields between $\mathbb{C}(X)$ and $\mathbb{C}(f)$ are precisely the fields $\mathbb{C}(h)$, where $g, h \in \mathbb{C}(X)$ satisfy $f = g \circ h$; moreover, for $h, H \in \mathbb{C}(X)$, we have $\mathbb{C}(h) = \mathbb{C}(H)$ if and only if there is a degree-one $\mu \in \mathbb{C}(X)$ such that $h = \mu \circ H$. If f is a Laurent polynomial (respectively, polynomial) and $f = g \circ h$ with $g, h \in \mathbb{C}(X)$, then there is a degree-one $\mu \in \mathbb{C}(X)$ such that both $g \circ \mu$ and $\mu^{-1} \circ h$ are Laurent polynomials (respectively, polynomials).*

With this result we may show the following:

Lemma 2. *For $f = g \circ h \in \mathbb{C}[(X)], g, h \in \mathbb{C}(X)$, there is a degree-one $\mu \in \mathbb{C}(X)$ such that both $g \circ \mu \in \mathbb{C}[(X)]$ and $\mu^{-1} \circ h \in \mathbb{C}[(X)]$ and either (i) $g \circ \mu \in \mathbb{C}[X]$ or (ii) $\mu^{-1} \circ h = X^s$ for some $s \in \mathbb{Z}$, or both.*

Proof. Let $\hat{g} = g \circ \mu, \hat{h} = \mu^{-1} \circ h \in \mathbb{C}[(X)]$ which exist by Lemma 1. Suppose that $\hat{h}$ is not a monomial and $\hat{g} \notin \mathbb{C}[X]$. We then have $\hat{g} = X^{-n}G, G \in \mathbb{C}[X], n > 0 \in \mathbb{Z}$. Because $\hat{h}$ is not a monomial, it will have a finite non-zero root. This will be a pole of f due to the X^{-n} factor of $\hat{g}$. This contradicts the fact that f can have poles only at zero and infinity. □

The case where h is a monomial may be handled trivially, so we restrict our attention to the situation where $g \in K[X]$.

4 Facts about Univariate Laurent Polynomials

We now present some elementary facts about Laurent polynomials that are required to justify our first algorithm.

Engstrom [2] observed that for polynomial composition the leading coefficients of f and $g_r h^r$ agree and, if $h(0) = 0$, give a triangular system for the coefficients of h. The polynomial decomposition algorithm of Kozen and Landau [5] is based on this fact. We develop generalizations of these ideas for Laurent polynomials. We begin by showing that both the leading s terms and trailing t terms of f and $g_r h^r$ agree.

Lemma 3. *The coefficients of X^i in f and $g_r h^r$ agree for $i > rs - s$ and for $i < -rt + t$.*

Proof. Let $f = g_r h^r + F$ for $F = \sum_{i=0}^{r-1} g_i h^i$. The degree of F is $\langle -t(r-1), s(r-1)\rangle$ so F has vanishing support for X^i, $i > rs - s$ and $i < -rt + t$. □

Next we show that the leading and trailing terms of f depend, respectively, only on the positive and negative degree terms of h.

Lemma 4. *Let $h_+ = \sum_{i=1}^{s} h_i X^i$ and $h_- = \sum_{i=1}^{t} h_{-i} X^{-i}$ so $h = h_+ + h_0 + h_-$. Then the coefficients of X^i in f and $g_r h_+^r$ agree for $i > rs - s$. Likewise, the coefficients of X^i in f and $g_r h_-^r$ agree for $i < -rt + t$.*

Proof. Let $f = g_r h^r + F$. The only f_i with $i < -rt + t$ or $i > rs - s$ arise from $g_r h^r = g_r \sum_{r_+ + r_0 + r_- = r} \binom{r}{r_+ \; r_0 \; r_-} h_+^{r_+} h_0^{r_0} h_-^{r_-}$. If both r_+ and$r_0 + r_-$ are non-zero, then $1 \leq \deg(h_+^{r_+}) \leq (r-1)s$ and $-(r-1)t \leq \deg(h_0^{r_0} h_-^{r_-}) \leq 0$ so $-rt+t+1 \leq \deg(h_+^{r_+} h_0^{r_0} h_-^{r_-}) \leq rs-s$. Only when $r_+ = r$ can the degree exceed $rs-s$. Therefore $[X^i]f = [X^i]g_r h_+^r$ when $i > rs-s$. Similarly, $[X^i]f = [X^i]g_r h_-^r$ when $i < rt - t$. □

The following is the observation of Engstrom, where we leave h_s unrestricted in order to make certain statements easier later.

Lemma 5. *The coefficients of h_+ are determined, up to a choice of h_s, by the triangular system*

$$\left.\begin{array}{ll} g_r & = f_{rs}/h_s^r \\ h_{s-i} & = f_{rs-i}/(rg_r h_s^{r-1}) + P_{s-i}(h_s, \ldots, h_{s-i+1}, g_r),\ 1 \leq i \leq s-1 \end{array}\right\} \quad (2)$$

where P_{s-i} is a polynomial function of $i+1$ variables.

Proof. Lemma 4 and multinomial expansion of $g_r h_+^r$. □

A similar result holds for the trailing terms:

Lemma 6. *The coefficients of h_- are determined, up to a choice of h_{-t}, by the triangular system*

$$\left.\begin{array}{ll} g_r & = f_{-rt}/h_{-t}^r \\ h_{-t+i} & = f_{-rt+i}/(rg_r h_{-t}^{r-1}) + P_{-t+i}(h_{-t}, \ldots, h_{-t+i-1}, g_r),\ 1 \leq i \leq t-1 \end{array}\right\} \quad (3)$$

where P_{-t+i} is a polynomial function of $i+1$ variables.

Proof. As for Lemma 5. □

We will also require the following simple fact.

Lemma 7. *Given $k \neq 0 \in K$, there exist $\hat{g} \in K[X]$, $\hat{h} \in K[(X)]$, such that $f = \hat{g} \circ \hat{h}$, $\hat{h}_s = k$, $\hat{h}_0 = 0$, $\deg g = \deg \hat{g}$, $\deg h = \deg \hat{h}$.*

Proof. Take $\hat{g} = g \circ (\frac{X}{a} - \frac{b}{a})$ and $\hat{h} = (aX+b) \circ h$ where $a = k/h_s$ and $b = -h_0/h_s$. Then $\hat{h}_s = k, \hat{h}_0 = 0$ as desired, and $\hat{g} \circ \hat{h} = g \circ h$ by the associativity of $\circ$. □

5 The Two-Ended Algorithm

5.1 Finding *h*

It is possible to find the decomposition of Laurent polynomials using the ideas presented in Section 4. Given $f \in K[(X)]$ of degree $\langle -rt, rs \rangle$ we may find a candidate inner composition factor h_{cand} of degree $\langle -t, s \rangle$ by independently finding the positive degree terms, $h_{\text{cand}+}$, and negative degree terms, $h_{\text{cand}-}$. By Lemma 7, the constant term, $h_{\mathrm{cand}0}$, can be set to zero. Once h_{cand} is chosen, the outer composition factor g, if it exists, may be found easily by a number of methods.

There is one point that requires particular attention, however. While it is possible to specify an arbitrary leading coefficient or trailing coefficient for h_{cand}, they may not be chosen independently. Lemmas 5, 6 and 7 show that we are free to choose h_0 and we can find all the other coefficients of h if we know h_s and h_{-t}. We choose $h_s = 1$ and set $h_0 = 0$. Then requiring g_r to be the same in the systems for both leading and trailing coefficients gives

$$h_{-t}^r = f_{-rt}/f_{rs}\,. \tag{4}$$

Depending on the field K, there may be up to r possible values for h_{-t} satisfying this equation. These do not normally all lead to decompositions of f.

Example 1. Let

$$f = X^4 + 4X^3 + 4X^2 + 6X + 3 - 20X^{-1} + 9X^{-2} - 30X^{-3} + 25X^{-4}.$$

We set $r = 2$, $h_2 = 1$ and find $h_+ = X^2 + 2X$. The possibilities for h_{-2} are then $\pm\sqrt{25}$. Choosing $h_{-2} = -5$ gives $f = (X^2 + 1) \circ (X^2 + 2X + 3X^{-1} - 5X^{-2})$. Choosing $h_{-2} = +5$ gives $h_{\mathrm{trial}} = X^2 + 2X - 3X^{-1} + 5X^{-2}$. Composing with generic g and equating coefficients with f gives an inconsistent system. There is therefore no g such that $f = g \circ h$ with $h_2 = 1$ and $h_{-2} = 5$. □

It is possible to try each of the r possible choices for h_{-t} until one leads to a decomposition. This is the main idea of our "two-ended" algorithm. We shall explain this in more detail shortly. We first present a few pre-requisites.

The first component is an algorithm to find a candidate h_+, given f the degree and the desired leading coefficient for h. This is used twice in the two-ended algorithm — once to find h_+ from f and once to find $\hat{h}_- = h_-/h_{-t}$ from $f(1/X)$.

Algorithm 1 (Positive Degree Terms of h)

```
INPUT:
    $f \in K[(X)]$ of degree $\langle -rt, rs\rangle$ and $r \geq 2 \in \mathbb{Z}$.
OUTPUT:
    A monic polynomial $h_+ \in K[X]$, such that if there exist $g \in K[X]$,
    $h \in K[(X)]$, $\deg g = r$, $f = g \circ h$, then a choice of $h$ has $\sum_{i=1}^{s} h_i X^i = h_+$.
    Note, it may be that there do not exist $g$, $h$ of the required degrees such
    that $f = g \circ h$.
METHOD:
 1. Let $p := X^s$.
 2. For $k$ from 1 to $s - 1$,
    (a) Let $c := \frac{1}{r}[X^{rs-k}](f/f_{rs} - p^r)$.
    (b) Let $p := p + cX^{s-k}$.
 3. Return $h_+ = p$.
```

Theorem 1. *Algorithm 1 solves the polynomial system (2).*

Proof. Let $c_{(k)}$ and $p_{(k)}$ be the values of c and p after k iterations of the loop. We have $f_{rs-k} = [X^{rs-k}] g_r h_+^r$ by Lemma 4 so step 2a computes

$$c_{(k)} = \frac{1}{r}[X^{rs-k}] \left(\left(p_{(k-1)} + h_{s-k}X^{s-k} + O(X^{s-k-1})\right)^r - p_{(k-1)}^r \right)$$

Induction on k shows $p_{(k)} = \sum_{i=0}^{k} h_{s-i}X^{s-i}$ so $p_{(s-1)} = h_+$. The system (2) is triangular and introduces each variable linearly so the solution is unique. $\square$

5.2 Finding g

In the case of polynomials, Kozen and Landau find g by solving the triangular linear system $\mathbf{A} \cdot \mathbf{g} = \mathbf{f}$ with entries $\mathbf{A}_{ij} = [X^{is}]h^j$, $\mathbf{g}_i = g_i$, $\mathbf{f}_i = f_{is}$, $0 \le i, j \le r$. They observe that the coefficients $\mathbf{A}_{ij}$ can be saved during the construction of h and that $h_{(k)}^r$ may be computed using values from previous iterations.

For Laurent polynomials, finding g by solving a linear system would require the coefficients $\mathbf{A}_{ij} = [X^{is}](h_+ + h_{-t}\hat{h}_-)^j$ for a choice of h_{-t}. These are not immediately available as the two applications of Algorithm 1 produce $[X^{is}]h_+^j$ and $[X^{is}]\hat{h}_-^j$. We may nevertheless compute the matrix $\mathbf{A}$, given h_+, $\hat{h}_-$ and h_{-t}, but the advantage of using saved values from the construction of h is lost. Moreover, we generally need to construct this matrix for several choices of h_{-t}. While it is possible to do this, depending on the field, it may be more convenient to find g by interpolation.

Finding g by linear system solving

Suppose we have h_+, $\hat{h}_-$, h_{-t} and f and wish to find g by solving a linear system.

1. Find the $(r+1)^2$ coefficients $\mathbf{A}_{ij} = [X^{is}](h_+ + h_{-t}\hat{h}_-)^j$ for $0 \le i, j \le r$. Computing $\mathbf{A}_{ij}$ can be done in time $\sum_{i=1}^{r} M(\ell, i\ell)$. This can be done in time $O(r^2\ell^2) = O(M(r\ell))$ with classical polynomial multiplication or time $O(r^2\ell \log(r\ell)) = O(rM(r\ell))$ with fast arithmetic.
2. Solve the triangular system $\mathbf{A} \cdot \mathbf{g} = \mathbf{f}$, which can be done in time $O(r^2)$.

If up to r such systems must be solved, with $\mathbf{A}_{ij}$ being computed afresh each time, then time $O(r^3\ell^2)$ is required for classical arithmetic or $O(r^3\ell \log(r\ell))$ for fast arithmetic.

Finding g by interpolation

Suppose we have h_+, $\hat{h}_-$, h_{-t} and f and wish to find g by interpolation

1. Evaluate h_+, $\hat{h}_-$ and f at points, $\alpha_1, \ldots, \alpha_q \in K$, until $r+1$ distinct values are found for $h_+ + h_{-t}\hat{h}_-$. This requires $2q(r+1)\ell$ operations.

2. Interpolate the points $\{(h_+(\alpha_j) + h_{-t}\hat{h}_-(\alpha_j), f(\alpha_j)) \mid 1 \le j \le q\}$ to obtain g. This requires $O(r^2) = O(M(r))$ operations with classical arithmetic or $O(r \log^2 r) = O(\log r M(r))$ operations for fast arithmetic.

If multiple such interpolations must be performed, the values of $h_+, \hat{h}_-, f$ need not be recomputed. Only the q sums $h_+ + h_{-t}\hat{h}_-$ need be recomputed, requiring $2q$ operations. If up to r interpolations are required, the total time is then $O(qr\ell + r^3)$ for classical arithmetic or $O(qr\ell + r^2 \log^2 r)$ for fast arithmetic. If the field is large enough, $q = r + 1$ with high probability. Thus we have expected time $O(r^2\ell + r^3)$ with classical arithmetic and expected time $O(r^2\ell + r^2 \log^2 r)$ with fast arithmetic. In the worst case, because there may be up to ℓ values of X such that $h = \alpha$, it is theoretically possible to require as many as $q = (r + 1)\ell$ evaluations of h. The worst case is thus $O(r^2\ell^2 + r^3)$ for classical arithmetic or $O(r^2\ell^2 + r^2 \log^2 r)$ for fast arithmetic.

Comparison

The complexity of finding the outer composition factor g by linear system solving and by interpolation is summarized in the table below. The first two columns give the time complexity if only one candidate for h is tried and the second pair of columns give the time complexity if $O(r)$ possibilities for h_{-t} must be tried.

	1 Linear Sys.	1 Interp.	r Linear Sys.	r Interp.
Expected Classical	$O(r^2\ell^2)$	$O(r^2\ell)$	$O(r^3\ell^2)$	$O(r^2\ell + r^3)$
Expected Fast	$O(r^2\ell \log(r\ell))$	$O(r^2\ell)$	$O(r^3\ell \log(r\ell))$	$O(r^2\ell + r^2 \log^2 r)$
Worst Case Class.	$O(r^2\ell^2)$	$O(r^2\ell^2)$	$O(r^3\ell^2)$	$O(r^2\ell^2 + r^3)$
Worst Case Fast	$O(r^2\ell \log(r\ell))$	$O(r^2\ell^2)$	$O(r^3\ell \log(r\ell))$	$O(r^2\ell^2 + r^2 \log^2 r)$

Provided the field has sufficiently many elements, the only situation where solving a linear system is superior to interpolation is when *all* of the following conditions hold:

1. the worst case number of evaluations is required (unlikely),
2. $O(r)$ candidates for h must be tried,
3. fast arithmetic is used, and
4. $O(r \log(r\ell)) < O(\ell)$, *e.g.* when searching for g of fixed low degree.

Under normal circumstances, therefore, interpolation should be used. This may be done as described in Algorithm 2.

Algorithm 2 (Interpolation of g)
Input:
$f \in K[(X)]$ *with* $\deg f = \langle -rt, rs\rangle$,
$h_+ \in K[X]$ *with* h_+ *monic,* $\deg h_- = s$,
$\hat{h}_- \in K[(X)]$ *with* $\hat{h}_-(X^{-1}) \in K[X]$, $\hat{h}_-(X^{-1})$ *monic,* $\deg \hat{h}_-(X^{-1}) = t$
T *a finite set of values* $\{\tau_i \in K\}$.

```
OUTPUT:
    If there exit $g \in K[X]$ and $\tau \in T$, such that $f = g \circ (h_+ + \tau\hat{h}_-)$, then
    returns $g$ and $\tau$. Otherwise returns FAIL.
METHOD:
  1. Choose $r+1$ values $\alpha_j \in K$, and compute $F_j = f(\alpha_j)$, $H_{+j} = h_+(\alpha_j)$,
     $H_{-j} = \hat{h}_-(\alpha_j)$, $j = 1, \ldots, r+1$.
  2. For each value $\tau_i \in T$,
     (a) Compute the values $H_j = H_{+j} + \tau_i H_{-j}, j = 1, \ldots, r+1$
     (b) While the values $H_j$ are not all distinct, say $H_{j_1} = H_{j_2}$, choose a new
         for $\alpha_{j_1}$ and recompute $F_{j_1}, H_{+j_1}, H_{-j_1}, H_{j_1}$.
     (c) Form $g$ by interpolating the points $(H_j, F_j), j = 1, \ldots, r+1$.
     (d) Test whether $f = g \circ (h_+ + \tau_i\hat{h}_-)$. If so, return $g$ and $\tau_i$.
  3. Return FAIL.
```

5.3 Two-Ended Univariate Laurent Polynomial Decomposition

The above results may be combined to give an algorithm for the decomposition of univariate Laurent polynomials. The leading coefficients for h_+ and the trailing coefficients for a multiple of h_- are found, and the possible values of h_{-t} are tried to put them together.

Algorithm 3 (Two-Ended Univariate Laurent Polynomial Decomposition)

```
INPUT:
    $f \in K[(X)]$ of degree $\langle -rt, rs \rangle$ and $r \geq 2 \in \mathbb{Z}$.
OUTPUT:
    If there exist $g \in K[X], h \in K[(X)]$ such that $\deg g = r$, $f = g \circ h$,
    returns a choice of $g$ and $h$. Otherwise, returns FAIL.
METHOD:
  1. Apply Algorithm 1 to $f(X)$ and $r$ to compute monic $h_+(X) \in K[X]$.
  2. Apply Algorithm 1 to $f(\frac{1}{X})$ and $r$ to compute monic $\hat{h}_-(\frac{1}{X}) \in K[X]$.
  3. Compute the set $T = \{\tau \in K \mid \tau^r = f_{-rt}/f_{rs}\}$.
  4. Apply Algorithm 2 to $f(X), h_+(X), \hat{h}_-(X)$ and $T$ to find $g$ and $\tau$.
     If Algorithm 2 returns FAIL, return FAIL.
  5. Let $h_{-t} = \tau$ and return $g$ and $h_+ + h_{-t}\hat{h}_-$.
```

Although this method requires up to r attempts to find the inner composition factor h, it is easy to implement in a setting where polynomial decomposition is already provided. Also, in some important cases the trailing coefficient equation has only a few solutions, and possibly only one. For example, when $K = \mathbb{R}$ there are one or two alternatives for h_{-t} according as r is odd or even.

If implementing Laurent polynomial decomposition *ab initio*, it is possible to find a candidate for h by examining only the leading coefficients of f and without having to try alternatives. For this we need a few more properties of Laurent polynomials.

6 Further Facts about Univariate Laurent Polynomials

For the second algorithm for Laurent polynomial decomposition it is useful to consider more leading and trailing coefficients than contemplated by Lemma 3. The following obviously generalizes to $i > (r-k)s$ and $i < -(r-k)t$, but we need only $k = 2$.

Lemma 8. *The coefficients of X^i in f and $g_r h^r + g_{r-1}h^{r-1}$ agree for $i > (r-2)s$ and for $i < -(r-2)t$.*

Proof. As for Lemma 3.

The leading coefficients are related as follows:

Lemma 9. *Let $T = \min(t, s-1)$. The coefficients of g, h and the leading $s+T+1$ coefficients of f are related by a system of polynomial equations of the form*

$$\begin{aligned}
f_{rs} &= g_r h_s^r \\
f_{rs-i} &= r g_r h_s^{r-1} h_{s-i} + P_{s-i}(h_s, \dots, h_{s-i+1}, g_r), && 1 \le i \le s-1 \\
f_{rs-i} &= r g_r h_s^{r-1} h_{s-i} + g_{r-1} h_s^{r-1} + P_{s-i}(h_s, \dots, h_{s-i+1}, g_r) && i = s \\
f_{rs-i} &= r g_r h_s^{r-1} h_{s-i} + P_{s-i}(h_s, \dots, h_{s-i+1}, g_r, g_{r-1}), && s+1 \le i \le s+T\,.
\end{aligned}$$

Proof. Lemma 8 and multinomial expansion of $g_r h^r + g_{r-1}h^{r-1}$. □

The key observation that allows a one-ended algorithm is that the triangular system (2) can be extended, *as a triangular system*, if h_0 is restricted to be 0. We see this as follows: From Lemma 8 we know $f_{rs-s} = [X^{rs-s}](g_r h^r + g_{r-1}h^{r-1})$. A degree counting argument shows that this coefficient can depend only on $h_i, i \ge 0$, g_r and g_{r-1}. Higher degree coefficients of f give all of these but h_0 and g_{r-1} by (2). Then restricting $h_0 = 0$ determines g_{r-1}. We then have a triangular system that introduces each of the coefficients of h and g_{r-1} linearly.

Lemma 10. *If $f \in K[(X)]$ and $r \ge 2 \in \mathbb{Z}$ invertible in K, such that $f = g \circ h$ for some $g \in K[X]$ of degree r and $h \in K[(X)]$ of degree $\langle -t, s\rangle$, then g_r, g_{r-1} and all coefficients of h, save possibly h_{-t}, can be determined by a triangular system of the form:*

$$\left.\begin{aligned}
g_r &= Q_s\ (f_{rs}) \\
h_{s-i} &= Q_{s-i}(f_{rs-i}, h_{s-1}, \dots, h_{s-i+1}, g_r^{-1}, g_r) && 1 \le i \le s-1 \\
g_{r-1} &= Q_0\ (f_{rs-s}, h_{s-1}, \dots, h_1, \quad g_r^{-1}, g_r) \\
h_{s-i} &= Q_{s-i}(f_{rs-i}, h_{s-1}, \dots, h_1, h_{-1}, \dots, h_{s-i+1}, g_r^{-1}, g_r, g_{r-1}) \\
& && s+1 \le i \le s+T
\end{aligned}\right\} \tag{5}$$

where $T = \min(t, s-1)$ and each Q_{s-i} is a polynomial function of $i+1$ variables. The coefficient h_{-t} is also determined if $t < s$.

Proof. As allowed by Lemma 7, we set $h_s = 1$, $h_0 = 0$ and specialize the system of Lemma 9. □

The above results are sufficient for our purposes when $t < s$, but the following will be necessary when $t = s$.

Lemma 11. *If $f \in K[(X)]$ is of degree $\langle -rs, rs\rangle$, and $f = g\circ(h_sX^s + h_{-s}X^{-s})$, then*

$$f_{is} = \sum_{n=0}^{\lfloor \frac{r-i}{2}\rfloor} \binom{2n+i}{n+i} g_{2n+i} h_s^{n+i} h_{-s}^n \qquad 0 \le i \le r\,, \tag{6}$$

$$h_{-s}^i f_{is} = h_s^i f_{-is} \qquad -r \le i \le r\,, \tag{7}$$

$$f_j = 0 \qquad j \ne is,\ -r \le i \le r\,. \tag{8}$$

Proof. Use induction on r, noting $\sum_{n=\lfloor\frac{r-1-i}{2}\rfloor+1}^{\lfloor\frac{r-i}{2}\rfloor}$ is empty if $r - i$ is odd and otherwise gives one term with $n = (r-i)/2$. □

7 The One-Ended Algorithm

We now show how to decompose a Laurent polynomial by solving a triangular system derived from its leading coefficients. In the following we assume $0 < t \le s$. This does not exclude any Laurent polynomials: If $t = 0$, the problem reduces to ordinary polynomial decomposition. If $t > s$, the algorithm can be applied to $f(\frac{1}{X})$. Under these assumptions, we are able to determine all the coefficients of h, except possibly h_{-t}, from the leading $2s$ coefficients of f. The coefficient h_{-t} is also found if $t < s$. The following algorithm computes h, possibly minus its trailing term.

Algorithm 4 (Determining $h - \eta$)

```
INPUT:
    $f \in K[(X)]$ and $r \ge 2 \in \mathbb{Z}$, with $\deg f = \langle -rt, rs\rangle, s \ge t$.
OUTPUT:
    If there exist $g \in K[X]$, $\deg g = r$ and $h \in K[(X)]$ such that $f = g \circ h$,
    returns a choice of $h - \eta$, where $\eta = h_{-s}X^{-s}$. (Note $\eta = 0$ if $s > t$.)
METHOD:
  1. Let $p := X^s$.
  2. For k from 1 to $s - 1$,
     (a) Let $c := \frac{1}{r}[X^{rs-k}](f/f_{rs} - p^r)$.
     (b) Let $p := p + cX^{s-k}$.
  3. Let $g1 := [X^{rs-s}](f/f_{rs} - p^r)$.
  4. For k from $s + 1$ to $s + \min(s - 1, t)$,
     (a) Let $c := \frac{1}{r}[X^{rs-k}](f/f_{rs} - p^{r-1}(p + g1))$.
     (b) Let $p := p + cX^{s-k}$.
  5. Return $h = p$.
```

Theorem 2. *Algorithm 4 solves the polynomial system (5).*

Proof. We take $h_s = 1, h_0 = 0$. Step 2 gives the values for $h_{s-1}, \ldots, h_1$ by Theorem 1. A similar argument shows that Step 3 computes $g1 = g_{r-1}$ and that Step 4a computes $c_{(k)} = [X^{rs-k}]\left(f/f_{rs} - (h^r + g_{r-1}h^{r-1})\right)$. By Lemma 8, these give the unique values for $h_{-1}, \ldots, h_{-T}$, $T = \min(s-1, t)$. □

Algorithm 4 gives h if $s > t$, but if $s = t$ the coefficient h_{-t} is not found. Depending on the form of h, it is possible to find this remaining coefficient in one of two ways. If the $h - \eta$ computed by Algorithm 4 has more than one term, then we may compute decompositions of $f(X)$ and $f(\frac{1}{X})$ and use the ratio of a pair of corresponding interior coefficients to determine h_{-t}. Otherwise, a special method is used for $h = X^s + h_{-s}/X^s$. These two procedures are described below.

```
Algorithm 5 (Determining $h_{-s}$ when $s = t$, $h \neq h_s X^s + h_0 + h_{-s}X^{-s}$)
Input:
    $f \in K[(X)]$ of degree $\langle -rs, rs\rangle$, $h - h_{-s}X^{-s} \in K[(X)]$ such that $f = g \circ h$,
    $g \in K[X]$, $h \neq h_s X^s + h_0 + h_{-s}X^{-s}$.
Output:
    Returns $h_{-s}$.
Method:
  1. Find the smallest $i$, $s-1 \leq i \leq -s+1$, such that $h_i \neq 0$.
  2. Apply Algorithm 4 to compute $\bar{h} - \bar{h}_{-s}X^{-s}$ from $f(\frac{1}{X})$ and $r$.
     Algorithm 4 may be terminated early, as soon as $\bar{h}_{-i}$ is computed.
  3. Return $h_{-s} = h_{-i}/\bar{h}_i$.
```

Note that here $h_0 = \bar{h}_0 = 0$ and one $h_i \neq 0$ by the input requirements.

```
Algorithm 6 (Determining $h_{-1}$ when $h = X + h_{-1}X^{-1}$)
Input:
    $f \in K[(X)]$ of degree $\langle -r, r\rangle$ such that $f = g \circ h$ for some $g \in K[X]$ and
    $h = X + h_{-1}X^{-1}$.
Output:
    Returns $h_{-1}$.
Method:
  1. Let $m = \gcd_{i \in I}(i)$ where $I = \{i \mid i > 0, f_i \neq 0\}$.
  2. If $m = 1$,
     (a) Compute $c_i = f_{-i}/f_i$,  $i \in I$. Note $c_i = {h_{-1}}^i$, by (7).
     (b) Use the extended Euclidean algorithm to find $m_i$, $\sum_{i \in I} m_i i = 1$.
     (c) Return $h_{-1} = a$ where $a = \prod_{i \in I} c_i^{m_i}$. Note $\prod_{i \in I} c_i^{m_i} = {h_{-1}}^{\sum_{i \in I} m_i i}$.
  3. If $m > 1$,
     (a) Recursively find $G \circ H = \sum_{i=-r/m}^{r/m} f_{mi} X^i$, $\deg G = r/m$, $H = X + A/X$.
     (b) Return $h_{-1} = a$ for any $a$ such that $a^m = A$.
```

We now have all the ingredients of the one-ended algorithm for univariate Laurent polynomial decomposition. We require $s \geq t$ so that, with the restriction $h_0 = 0$ and $h_s = 1$, the first $2s$ coefficients of f give a triangular system for g_r, g_{r-1} and all the coefficients of h, except possibly h_{-s}. As stated earlier, if $s < t$ we apply the algorithm to $f(\frac{1}{X})$.

Algorithm 7 (One-Ended Univariate Laurent Polynomial Decomposition)

```
INPUT:
    $f \in K[(X)]$ of degree $\langle -rt, rs \rangle$, $s \geq t$ and $r \geq 2 \in \mathbb{Z}$.
OUTPUT:
    If there exist $g \in K[X], h \in K[(X)]$ such that $\deg g = r$, $f = g \circ h$,
    returns a choice of $g$ and $h$. Otherwise, returns FAIL.
METHOD:
 1. Apply Algorithm 4 to $f$ and $r$ to obtain $h - \eta$.
 2. If $s > t$, then $\eta = 0$ and we have $h$.
 3. If $s = t$, then
    (a) If $h - \eta$ is a monomial, then
          i. If any $f_j \neq 0$ for $s \not| j$, return FAIL.
         ii. Form $F = \sum_{i=-r} r f_{is} X^i$.
        iii. Apply Algorithm 6 to $F$ to compute $h_{-s}$
    (b) If $h - \eta$ is not a monomial, then
          i. Apply Algorithm 5 to $f$ and $h - \eta$ to compute $h_{-s}$.
    We now have a candidate for $h$.
 4. Construct the corresponding $g$ by interpolation or by solving the linear system
    $\mathbf{A} \cdot \mathbf{g} = \mathbf{f}$ where $\mathbf{A}_{ij} = [X^{is}]h^j$, $\mathbf{g}_i = g_i$, $\mathbf{f}_i = f_{is}$, $0 \leq i, j \leq r$.
    The coefficients $\mathbf{A}_{ij}$ computed by Algorithm 4 in Step 1 may be reused.
 5. Test whether $f = g \circ h$. If so, return $g$ and $h$. Otherwise return FAIL.
```

8 Multivariate Laurent Polynomial Decomposition

The functional decomposition of Laurent polynomials can be extended to the multivariate case. We consider the following problem:

Problem 2 (Multivariate Laurent polynomial decomposition).
Given $f \in K[(X_1, \ldots, X_v)]$, K a field, and $r \geq 2 \in \mathbb{Z}$, do there exist $g \in K[Y]$ of degree r and $h \in K[(X_1, \ldots, X_v)]$ such that $f = g \circ h$? If so find such g and h.

We reduce this to univariate Laurent polynomial decomposition. The reduction is not entirely trivial because the univariate algorithm sets $h_0 = 0$ and the usual multivariate reduction techniques may require $h_0 \neq 0$.

To discuss the problem we set the following notation. Let $f \in K[(X_1, \ldots, X_v)]$. We seek a decomposition $f = g \circ h$ with $g \in K[Y]$ with $\deg g = r$. We require that r have an inverse in K and let $\deg f = \langle(-rt_1, \ldots, -rt_v), (rs_1, \ldots, rs_v)\rangle$. We use the notation $p_{i_1 \ldots i_v} = [X_1^{i_1} \cdots X_v^{i_v}]\, p$ where convenient.

Our univariate decomposition methods are based on the degrees of monomials. We will therefore employ techniques that preserve monomial degree. The first problem is then to find a weight vector such that no term of f, other than the constant term, has weighted total degree 0. This gives the following problem.

Problem 3 (Finding a constant-isolating weight vector).
Given a finite set of vectors $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(N)} \in \mathbb{Z}^n$, find a vector $\mathbf{w} \in \mathbb{Z}^n$ such that $\mathbf{v}^{(j)} \cdot \mathbf{w} \Leftrightarrow \mathbf{v}^{(j)} = 0$.

Finding such a weight vector is straightforward. Finding such a weight vector that, for efficiency, minimizes the weighted degree of f requires more attention.

Once such a weight vector is found, we may make substitutions $X_i \mapsto \alpha_i X_1^{w_i}, \alpha_i \in K, 2 \le i \le v$ to obtain a univariate problem. Because of the choice of w, setting $h_0 = 0$ in the univariate image omits only the constant term in the multivariate problem. Finding multiple images of h under different substitutions allows h to be constructed by dense or sparse interpolation. The outer composition factor g need be computed only once. As before, it is necessary to test whether the candidate h gives $f = g \circ h$ since not all of the coefficients of f were examined to construct the composition factors.

In practice, we have found it to be more convenient avoid interpolation and to construct a multivariate h candidate directly. This can be achieved by adapting Algorithm 4 to use polynomials of homogeneous weighted degree d wherever a monomial of degree d is used in the original algorithm.

9 Conclusions

Motivated by the desire to reason about symbolic polynomials, we have studied the problem of Laurent polynomial decomposition. We have presented two algorithms to find the functional decomposition, if one exists, of a Laurent polynomial f as $g \circ h$, where g is a polynomial of a specified degree. The "two-ended" method constructs h from the leading and trailing coefficients of f and can be implemented in terms of an existing polynomial decomposition library. The "one-ended" method is more efficient and constructs h from only the leading coefficients of f. Multivariate Laurent polynomial decomposition can be given in terms of either of these methods.

These methods may be used to give the complete decomposition of a Laurent polynomial into irreducible composition factors. Both of these methods are susceptible to the same techniques to improve asymptotic complexity as the polynomial decomposition method of Kozen and Landau. Test implementations have been made in the Maple computer algebra system.

References

1. Ritt, J.: Prime and composite polynomials. Trans. American Math. Society **23**(1) (1922) 51–66
2. Engstrom, H.T.: Polynomial substitutions. American Journal of Mathematics **63**(2) (1941) 249–255
3. Levi, H.: Composite polynomials with coefficients in an arbitrary field of characteristic zero. American Journal of Mathematics **64**(1) (1942) 389–400
4. Barton, D.R., Zippel, R.E.: A polynomial decomposition algorithm. In: Proc. 1976 ACM Symposium on Symbolic and Algebraic Computation, ACM Press (1976) 356–358
5. Kozen, D., Landau, S.: Polynomial decomposition algorithms. J. Symbolic Computation **22** (1989) 445–456
6. Zippel, R.E.: Rational function decomposition. In: Proc. ISSAC 2001, ACM Press (1991) 1–6
7. Kozen, D., Landau, S., Zippel, R.: Decomposition of algebraic functions. J. Symbolic Computation **22**(3) (1996) 235–246
8. von zur Gathen, J., Gutierrez, J., Rubio, R.: Multivariate polynomial decomposition. Applied Algebra in Engineering, Communication and Computing **14** (2003) 11–31
9. Zieve, M.E.: Decompositions of Laurent polynomials (2007) Preprint: arXiv.org:0710.1902v1.
10. Watt, S.M.: Making computer algebra more symbolic. In: Proc. Transgressive Computing 2006: A conference in honor of Jean Della Dora. (2006) 43–49
11. Watt, S.M.: Two families of algorithms for symbolic polynomials. In Kotsireas, I., Zima, E., eds.: Computer Algebra 2006: Latest Advances in Symbolic Algorithms – Proceedings of the Waterloo Workshop, World Scientific (2007) 193–210
12. Watt, S.M.: Symbolic polynomials with sparse exponents. In: Proc. Milestones in Computer Algebra 2008: A conference in honour of Keith Geddes' 60th birthday, Stonehaven Bay, Trinidad and Tobago, University of Western Ontario (2007) 91–97 ISBN 978-0-7714-2682-7.
13. Weispfenning, V.: Gröbner bases for binomials with parametric exponents. Technical report, Universität Passau, Germany (2004)
14. Yokoyama, K.: On systems of algebraic equations with parametric exponents. In: Proc. ISSAC 2004, ACM Press (2004) 312–319
15. Pan, W., Wang, D.: Uniform gröbner bases for ideals generated by polynomials with parametric exponents. In: Proc. ISSAC 2006, ACM Press (2006) 269–276
16. Watt, S.: Functional decomposition of symbolic polynomials. In: Proc. International Conference on Computatioanl Sciences and its Applications (ICCSA 2008), IEEE Computer Society (2008) 353–362