

Efficient q -Integer Linear Decomposition of Multivariate Polynomials

Mark Giesbrecht

*Symbolic Computation Group, Cheriton School of Computer Science, University of Waterloo,
Waterloo, ON, N2L 3G1, Canada*

Hui Huang

*Symbolic Computation Group, Cheriton School of Computer Science, University of Waterloo,
Waterloo, ON, N2L 3G1, Canada*

George Labahn

*Symbolic Computation Group, Cheriton School of Computer Science, University of Waterloo,
Waterloo, ON, N2L 3G1, Canada*

Eugene Zima

*Physics and Computer Science, Wilfrid Laurier University,
Waterloo, ON, N2L 3C5, Canada*

Abstract

We present two new algorithms for the computation of the q -integer linear decomposition of a multivariate polynomial. Such a decomposition is essential in the q -analogous world of symbolic summation, for example, describing the q -counterpart of Ore-Sato theory or determining the applicability of the q -analogue of Zeilberger's algorithm to a q -hypergeometric term. Both of our algorithms require only basic integer and polynomial arithmetic and work for any unique factorization domain containing the ring of integers. Complete complexity analyses are conducted for both our algorithms and two previous algorithms in the case of multivariate integer polynomials, showing that our algorithms have better theoretical performances. A Maple implementation is also included which suggests that our algorithms are also much faster in practice than previous algorithms.

Keywords: q -Analogue, Integer-linear polynomials, Polynomial decomposition, Newton polytope (Newton polygon), Creative telescoping, Ore-Sato theory

1. Introduction

Many objects in the ordinary shift world of symbolic summation find a natural counterpart commonly called q -analogues. In a typical situation, these are just slight adaptations of the

Email addresses: mwg@uwaterloo.ca (Mark Giesbrecht), h2huang@uwaterloo.ca (Hui Huang),
g1abahn@uwaterloo.ca (George Labahn), ezima@wlu.ca (Eugene Zima)

original objects but with involved variables promoted to exponents of an additional parameter q . Techniques for handling the originals often carry over to their q -analogues with some subtle modifications.

In this paper, we deal with the q -analogue of integer-linear decompositions of polynomials and aim to provide an intensive treatment for its computation in analogy to (Giesbrecht et al., 2019a). Surprisingly, although this q -analogue is obtained by modeling its ordinary shift counterpart, the primary technique used in (Giesbrecht et al., 2019a) can not be easily adapted to compute it due to different structures. A new alternative technique will be presented in this q -shift case.

In order to describe more details, we let D be a ring of characteristic zero and let $R = D[q, q^{-1}]$ be its transcendental ring extension by the indeterminate q . For n discrete indeterminates k_1, \dots, k_n distinct from q , we know that q^{k_1}, \dots, q^{k_n} are transcendental over R . We can then consider polynomials in q^{k_1}, \dots, q^{k_n} over R , all of which form a well-defined ring denoted by $R[q^{k_1}, \dots, q^{k_n}]$. We say an irreducible polynomial $p \in R[q^{k_1}, \dots, q^{k_n}]$ is q -integer linear over R if there exists a univariate polynomial $P(y) \in R[y]$ and two integer-linear polynomials $\sum_{i=1}^n \alpha_i k_i, \sum_{i=1}^n \lambda_i k_i \in \mathbb{Z}[k_1, \dots, k_n]$ such that

$$p(q^{k_1}, \dots, q^{k_n}) = q^{\sum_{i=1}^n \alpha_i k_i} P(q^{\sum_{i=1}^n \lambda_i k_i}).$$

In order to avoid superscripts, we will write the indeterminates q^{k_1}, \dots, q^{k_n} as the variables x_1, \dots, x_n in the sequel of the paper. Then the above definition can be rephrased as follows. An irreducible polynomial $p \in R[x_1, \dots, x_n]$ is called q -integer linear over R if there exists a univariate polynomial $P(y) \in R[y]$ and integers $\alpha_1, \dots, \alpha_n, \lambda_1, \dots, \lambda_n$ such that

$$p(x_1, \dots, x_n) = x_1^{\alpha_1} \cdots x_n^{\alpha_n} P(x_1^{\lambda_1} \cdots x_n^{\lambda_n}). \quad (1.1)$$

Note that the indeterminate q is hidden in the variables x_1, \dots, x_n . Since a common factor of the λ_i can be pulled out and absorbed into P , and a monomial can be merged into $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ if necessary, we assume that the integers $\lambda_1, \dots, \lambda_n$ have no common divisor, that the last nonzero integer is nonnegative, that $\lambda_i = 0$ whenever $\deg_{x_i}(p) = 0$ and that $P(0) \neq 0$. Such a vector $(\lambda_1, \dots, \lambda_n)$, as well as such a polynomial $P(y)$, is unique. We call the vector $(\lambda_1, \dots, \lambda_n)$ the q -integer linear type of p and the polynomial $P(y)$ its corresponding univariate polynomial. Note that the resulting $\alpha_1, \dots, \alpha_n$ all belong to \mathbb{N} since $p \in R[x_1, \dots, x_n]$ and $P(y) \in R[y]$. A polynomial in $R[x_1, \dots, x_n]$ is called q -integer linear (over R) if all its irreducible factors are q -integer linear, possibly with different q -integer linear types. For a polynomial $p \in R[x_1, \dots, x_n]$, we can define its q -integer linear decomposition by factoring into irreducible q -integer linear or non- q -integer linear polynomials and collecting irreducible factors having common types.

As with its ordinary shift counterpart, q -integer linear polynomials find broad applications in the q -analysis of symbolic summation. In particular, it is an important ingredient of the q -analogue of the Ore-Sato theorem for describing the structure of multivariate q -hypergeometric terms (Du and Li, 2019), which in turn, as indicated by (Chen and Koutschan, 2019), serves as a promising indispensable tool for settling a q -analogue of Wilf-Zeilberger's conjecture (Wilf and Zeilberger, 1992). Moreover, the q -integer linearity of polynomials plays a crucial role in detecting the applicability of the q -analogue of Zeilberger's algorithm (also known as the method of creative telescoping) for bivariate q -hypergeometric terms (Chen et al., 2005).

The full q -integer linear decomposition of polynomials is also very useful. On the one hand, it provides a natural way to determine the q -integer linearity of a given polynomial. On the

other hand, it enables one to compute the q -analogue of Ore-Sato decomposition of a given q -hypergeometric term, and can also be employed to develop a fast creative telescoping algorithm for rational functions in the q -shift setting in analogy to (Giesbrecht et al., 2019b). Evidently, the efficiency of the computation of q -integer linear decompositions directly affects the utility of all these algorithms.

In contrast to the ordinary shift case (Abramov and Le, 2002; Giesbrecht et al., 2019a; Li and Zhang, 2013), algorithms for computing the q -integer linear decomposition of a multivariate polynomial are not very well developed. As far as we are aware, there is only one algorithm available to compute such a decomposition of a bivariate polynomial. This algorithm was first developed by Le (2001, §5) with an extended description provided in (Le et al., 2001). Except for using the same pattern as its ordinary shift counterpart (Abramov and Le, 2002), this algorithm takes use of a completely different strategy, especially for finding q -integer linear types. This is mainly because all q -integer linear types appear as the exponent vectors of p , rather than as the coefficients in the ordinary shift case. The main idea used by Le (2001, §5) is to first find candidates for q -integer linear types by computing a resultant and then, for each candidate, extract the corresponding univariate polynomial via bivariate GCD computations. Given the algebraic machinery on which the algorithm is based, it is not clear how one can directly generalize this to handle polynomials in more than two variables.

Our first contribution is a bivariate-based scheme, similar to the one given in (Giesbrecht et al., 2019a, §4), through which the algorithm of Le readily tackles polynomials in any number of variables. For the sake of completeness, we include another algorithm based on full irreducible factorization, which can be viewed as a q -analogue of the algorithm developed by Li and Zhang (2013). This algorithm makes use of the observation that the difference of exponent vectors of any two monomials appearing in an irreducible q -integer linear polynomial, say the polynomial p of the form (1.1), must be a scalar multiple of the q -integer linear type $(\lambda_1, \dots, \lambda_n)$. We also give a complexity analysis for both algorithms, at least in the case of bivariate polynomials over $\mathbb{Z}[q, q^{-1}]$, supporting the superiority of the factorization-based algorithm over the algorithm of Le. Same opinion is suggested by the empirical tests.

The main contribution of this paper is a pair of new fast algorithms for computing the q -integer linear decomposition of a multivariate polynomial. Both algorithms will work for any unique factorization domain containing all integers and for any polynomial with an arbitrary number of variables. The first approach combines the main ideas of the two previous algorithms - in the sense that it follows the pattern of the algorithm of Le and also makes use of exponent vectors as the factorization-based algorithm - but avoids the computation of resultants as well as the need for full irreducible factorization. More precisely, this approach reduces the problem of finding candidates for q -integer linear types to an easy *ad hoc* geometry task of constructing the Newton polytope of the given polynomial, implying computations only using basic arithmetic operations $(+, -, \div, \times)$ of integers, and then computes each corresponding univariate polynomial by a simple content computation as in the ordinary shift case. As such we show that the q -analogue is actually simpler than its ordinary shift counterpart in the sense that, instead of finding rational roots of polynomials, one merely needs to perform basic integer manipulations.

Our second approach relies on the bivariate-based scheme mentioned earlier. This scheme takes any bivariate algorithm, that is, an algorithm for computing the q -integer linear decomposition of a bivariate polynomial, as a base case and iteratively tackles only two variables at a time until all variables are treated. This scheme, together with the bivariate restriction of the first approach, immediately establishes our second approach. Clearly, these two approaches coincide in the bivariate case.

Both approaches appear to be efficient in practice, though the second method shows an advantage for polynomials of a large number of variables. In order to do a theoretical comparison we have analyzed the worst-case running time complexity of both approaches in the case of multivariate polynomials over $\mathbb{Z}[q, q^{-1}]$. The analysis shows that the second approach is superior to the first one when the given polynomial has more than two variables. When restricted to the case of bivariate polynomials over $\mathbb{Z}[q, q^{-1}]$, the two approaches merge into one, which in turn is considerably faster than the algorithm of Le and the algorithm based on factorization. In addition, we also give experimental results which verify our complexity comparisons.

The remainder of the paper proceeds as follows. Background and basic notions required in the paper are provided in the next section. In Section 3 we present a fast algorithm for computing the q -integer linear decomposition in the special case of a bivariate polynomial including its complexity costs. Two approaches for the general multivariate case, along with their respective complexity analyses, are given successively in Sections 4 and 5. The following section provides a complexity comparison of our bivariate algorithm, the algorithm of Le and the factorization-based algorithm. The paper ends with an experimental comparison among all algorithms, along with a conclusion section.

2. Preliminaries

Throughout the paper, we let D be a unique factorization domain (UFD) of characteristic zero with $R = D[q, q^{-1}]$ denoting the transcendental ring extension by an indeterminate q . Note that a domain of characteristic zero always contains the ring of integers \mathbb{Z} as a subdomain. Let $R[x_1, \dots, x_n]$ be the ring of polynomials in x_1, \dots, x_n over R , where x_1, \dots, x_n are variables distinct from q . We reserve the variables x and y as synonyms for x_1 and x_2 , respectively, so as to avoid subscripts in the case when $n \leq 2$.

Let p be a polynomial in $R[x_1, \dots, x_n]$. Throughout this paper we will order monomials in $R[x_1, \dots, x_n]$ using a pure lexicographic order in $x_1 < \dots < x_n$. For this order we let $\text{lc}(p)$ and $\text{deg}(p)$ denote the leading coefficient and the total degree, respectively, of p with respect to x_1, \dots, x_n . We follow the convention that $\text{deg}(0) = -\infty$. We say that p is *monic* (over R) if $\text{lc}(p) = 1$. The *content* of p (over R), denoted by $\text{cont}(p)$, is the greatest common divisor (GCD) over R of the coefficients of p with respect to x_1, \dots, x_n with p being *primitive* if $\text{cont}(p) = 1$. The *primitive part* $\text{prim}(p)$ of p (over R) is defined as $p / \text{cont}(p)$. For brevity, we will omit the domain if it is clear from the context. In certain instances, we also need to consider the above notions with respect to a subset of the n variables. In these cases, we will either specify the relevant domain or indicate the related variables as subscripts of the corresponding notion. For example, $\text{lc}_{x_1, x_2}(p)$, $\text{deg}_{x_1, x_2}(p)$, $\text{cont}_{x_1, x_2}(p)$ and $\text{prim}_{x_1, x_2}(p)$ denote each function but applied to a polynomial p viewing it as a polynomial in x_1, x_2 over the domain $R[x_3, \dots, x_n]$.

In order to obtain a canonical representation, we introduce the notion of q -primitive polynomials in the univariate case. A polynomial $p \in R[x]$ is called *q -primitive* if it is primitive over R and its constant term $p(0)$ is nonzero. Note that this concept is a ring counterpart of q -monic polynomials introduced by Paule and Riese (1997). Clearly, any factor of a q -primitive polynomial in $R[x]$ is again q -primitive.

We work with n -dimensional vectors from the lattice \mathbb{Z}^n . In order to simplify notations, we employ bold letters, say \mathbf{i} , for an n -dimensional vector $(i_1, \dots, i_n) \in \mathbb{Z}^n$, and the *multi-index convention* $\mathbf{x}^{\mathbf{i}}$ for the monomial $x_1^{i_1} \cdots x_n^{i_n}$. The zero n -dimensional vector is denoted by boldface $\mathbf{0}$. As usual, the operations, especially addition/subtraction and scalar multiplication, on vectors in the lattice \mathbb{Z}^n are performed componentwise. Notice that there is a one-to-one correspondence

between monomials in $\mathbb{R}[x_1, \dots, x_n]$ and vectors in \mathbb{N}^n . From the monomial ordering $x_1 < \dots < x_n$, we then establish a natural total ordering “ $<$ ” on \mathbb{N}^n in such a way that for any two vectors $\mathbf{i}, \mathbf{j} \in \mathbb{N}^n$, if $\mathbf{x}^{\mathbf{i}} < \mathbf{x}^{\mathbf{j}}$ then we also say that $\mathbf{i} < \mathbf{j}$.

The support of polynomials plays a crucial role in our algorithms. Recall that the *support* of a polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$, denoted by $\text{supp}(p)$, is defined as the set of indices $\mathbf{i} \in \mathbb{N}^n$ with the property that the coefficient of $\mathbf{x}^{\mathbf{i}}$ in p is nonzero. Roughly speaking, $\text{supp}(p)$ records exponent vectors of all monomials present in p . Clearly, the support of a polynomial in $\mathbb{R}[x_1, \dots, x_n]$ is a finite set in \mathbb{N}^n , and it is the empty set if and only if the given polynomial is zero. As described previously, we sometimes only need to consider supports of polynomials with respect to part of the variables and these instances will be identified by explicitly pointing out the involved variables.

We are interested in finding the following decomposition of a polynomial, something briefly alluded to in the introduction.

Definition 2.1. Let $p \in \mathbb{R}[x_1, \dots, x_n]$ be a polynomial admitting the decomposition

$$p = c \mathbf{x}^{\alpha} P_0 \prod_{i=1}^m P_i(\mathbf{x}^{\lambda_i}), \quad (2.1)$$

where $c \in \mathbb{R}$, $m \in \mathbb{N}$, $\alpha \in \mathbb{N}^n$, $\lambda_i = (\lambda_{i1}, \dots, \lambda_{in}) \in \mathbb{Z}^n$, $P_0 \in \mathbb{R}[x_1, \dots, x_n]$ and $P_i(y) \in \mathbb{R}[y]$. Then (2.1) is called the q -integer linear decomposition of p (over \mathbb{R}) if

- (1) P_0 is primitive and none of its nonconstant irreducible factors is q -integer linear;
- (2) each $P_i(y)$ is q -primitive and of positive degree in y ;
- (3) each λ_i is a nonzero q -integer linear type, in other words, $\gcd(\lambda_{i1}, \dots, \lambda_{in}) = 1$ and its rightmost nonzero entry is positive;
- (4) any two vectors from the λ_i are distinct.

We say that the λ_i are q -integer linear types of p and each $P_i(y)$ is the corresponding univariate polynomial of the type λ_i .

Evidently, p is q -integer linear if and only if P_0 is a unit of \mathbb{R} in the decomposition (2.1). Hence all univariate polynomials are q -integer linear. By full factorization, we see that every polynomial admits a q -integer linear decomposition. Moreover, this decomposition is unique up to the order of factors and multiplication by units of \mathbb{R} , according to the uniqueness of the q -integer linear types and the full factorization.

We will be considering Laurent polynomials in the ring $\mathbb{R}[x_1, x_1^{-1}, \dots, x_{n-1}, x_{n-1}^{-1}, x_n]$, namely polynomials of the form $x_1^{-\alpha_1} \dots x_{n-1}^{-\alpha_{n-1}} \cdot p$ for some $\alpha_1, \dots, \alpha_{n-1} \in \mathbb{N}$ and $p \in \mathbb{R}[x_1, \dots, x_n]$. More generally, let \mathbb{K} throughout denote the quotient field of \mathbb{R} . Then we are interested in polynomials in x_n over the field $\mathbb{K}(x_1, \dots, x_{n-1})$, all of which form the ring $\mathbb{K}(x_1, \dots, x_{n-1})[x_n]$, containing $\mathbb{R}[x_1, x_1^{-1}, \dots, x_{n-1}, x_{n-1}^{-1}, x_n]$ as a subring. It is convenient to extend the definition of content and primitive part to polynomials in this setting. Let $p \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n]$ be of the form $\sum_{i=1}^d (a_i/b) x_n^i$ for $d \in \mathbb{N}$ and $a_i, b \in \mathbb{R}[x_1, \dots, x_{n-1}]$. Then we let the *content* $\text{cont}_{x_n}(p)$ of p with respect to x_n be $\gcd(a_1, \dots, a_d)/b$ and the corresponding *primitive part* $\text{prim}_{x_n}(p) = p/\text{cont}_{x_n}(p)$. Evidently, $\text{prim}_{x_n}(p) \in \mathbb{R}[x_1, \dots, x_n]$. In particular, if $p \in \mathbb{K}[x_1, \dots, x_n]$ then we can further consider its content with respect to all variables x_1, \dots, x_n . To be specific, by writing

$p = \sum_{i \in \mathbb{N}^n} (a_{i_1, \dots, i_n} / b) \mathbf{x}^i$ for $a_{i_1, \dots, i_n}, b \in \mathbb{R}$, we let its *content* $\text{cont}(p)$ (over \mathbb{K}) be a/b with a being the GCD of the a_{i_1, \dots, i_n} over \mathbb{R} and the *primitive part* $\text{prim}(p) = p / \text{cont}(p)$. Note that the definition of leading coefficient and degree extends to polynomials in $\mathbb{K}[x_1, \dots, x_n]$ in a natural manner. Again, as before, all these notions can be used in terms of partial variables.

Two basic lemmas are given below for later use.

Lemma 2.2. *Let $P(y) \in \mathbb{R}[y] \setminus \mathbb{R}$ with $P(0) \neq 0$ and let $\lambda \in \mathbb{Z}^n$ with $\gcd(\lambda_1, \dots, \lambda_n) = 1$, $\lambda_1, \dots, \lambda_{n-1}$ not all zero and $\lambda_n > 0$. Let $f \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n]$ be a factor of $P(\mathbf{x}^\lambda)$ which is monic with respect to x_n . Then*

- (i) *there exists $c \in \mathbb{K}$, $\alpha_1, \dots, \alpha_{n-1} \in \mathbb{Z}$ and a factor $g(y) \in \mathbb{R}[y]$ of $P(y)$ such that $f = c x_1^{\alpha_1} \cdots x_{n-1}^{\alpha_{n-1}} g(\mathbf{x}^\lambda)$. Moreover, if f is of positive degree in x_n then $\deg(g(y)) > 0$.*
- (ii) *$P(y)$ is irreducible over \mathbb{R} if and only if $P(\mathbf{x}^\lambda)$ is irreducible over $\mathbb{K}(x_1, \dots, x_{n-1})$ if and only if $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ is irreducible over \mathbb{R} .*

Proof. (i) Since $P(0) \neq 0$, all its roots in the algebraic closure $\overline{\mathbb{K}}$ of \mathbb{K} are nonzero. In order to prove the assertion, it is sufficient to show that $\mathbf{x}^\lambda - r$ for any root $r \in \overline{\mathbb{K}}$ of $P(y)$ is irreducible over $\overline{\mathbb{K}}(x_1, \dots, x_{n-1})$, because then since $f \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n]$ is a factor of $P(\mathbf{x}^\lambda)$ and it is monic with respect to x_n , it factors completely into irreducibles in $\overline{\mathbb{K}}(x_1, \dots, x_{n-1})[x_n]$ as follows

$$f = \prod_{i=1}^s (x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})(\mathbf{x}^\lambda - r_i) = (x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})^s \prod_{i=1}^s (\mathbf{x}^\lambda - r_i),$$

where $s \in \mathbb{N}$ with $s \leq \deg(P(y))$ and the $r_i \in \overline{\mathbb{K}}$ are roots of $P(y)$, and thus the assertion directly follows by pulling out the content over \mathbb{K} .

Let $r \in \overline{\mathbb{K}}$ be a root of $P(y)$ and suppose that $\mathbf{x}^\lambda - r$ is reducible over $\mathbb{K}(x_1, \dots, x_{n-1})$. Consider the algebraic closure $\overline{\mathbb{K}}(x_1, \dots, x_{n-1})$ of $\mathbb{K}(x_1, \dots, x_{n-1})$ and let $\omega \in \overline{\mathbb{K}}$ be a λ_n -th root of unity such that $\omega^{\lambda_n} = 1$. Since r is nonzero, the complete factorization of $\mathbf{x}^\lambda - r$ over $\overline{\mathbb{K}}(x_1, \dots, x_{n-1})$ is given by

$$\mathbf{x}^\lambda - r = x_1^{\lambda_1} \cdots x_{n-1}^{\lambda_{n-1}} \prod_{i=0}^{\lambda_n-1} \left(x_n - \omega^i r^{1/\lambda_n} x_1^{-\lambda_1/\lambda_n} \cdots x_{n-1}^{-\lambda_{n-1}/\lambda_n} \right).$$

It then follows from the reducibility of $\mathbf{x}^\lambda - r$ over $\mathbb{K}(x_1, \dots, x_{n-1})$ that there exist $i_1, \dots, i_k \in \{0, \dots, \lambda_n - 1\}$ with $0 < k < \lambda_n$ such that

$$\prod_{j=1}^k \left(x_n - \omega^{i_j} r^{1/\lambda_n} x_1^{-\lambda_1/\lambda_n} \cdots x_{n-1}^{-\lambda_{n-1}/\lambda_n} \right) \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n].$$

This implies that $(\lambda_i/\lambda_n)k \in \mathbb{Z}$ for all $i = 1, \dots, n-1$. Thus λ_n divides $k \cdot \gcd(\lambda_1, \dots, \lambda_{n-1})$ in \mathbb{Z} . Since $\lambda_1, \dots, \lambda_{n-1}$ are not all zero and $\gcd(\lambda_1, \dots, \lambda_n) = 1$, we have λ_n divides k in \mathbb{Z} , a contradiction with $0 < k < \lambda_n$.

(ii) For the first equivalence, the sufficiency is evident. In order to show the necessity, suppose that $P(\mathbf{x}^\lambda)$ is reducible over $\mathbb{K}(x_1, \dots, x_{n-1})$. Then there exists a factor $f \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n]$ of $P(\mathbf{x}^\lambda)$ which is of positive degree in x_n . By the assertion (i), we then obtain that there exists a polynomial $g(y) \in \mathbb{R}[y] \setminus \mathbb{R}$ dividing $P(y)$ in $\mathbb{R}[y]$, a contradiction with the assumption that $P(y)$ is irreducible over \mathbb{R} . Therefore, $P(\mathbf{x}^\lambda)$ is irreducible over $\mathbb{K}(x_1, \dots, x_{n-1})$.

For the second equivalence, by Gauß' lemma, one easily sees that $P(\mathbf{x}^\lambda)$ is irreducible over $\mathbb{K}(x_1, \dots, x_{n-1})$ if and only if $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ is irreducible over $\mathbb{R}[x_1, \dots, x_{n-1}]$. It thus amounts to showing the equivalence between the irreducibility of $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ over $\mathbb{R}[x_1, \dots, x_{n-1}]$ and its irreducibility over \mathbb{R} . The direction from \mathbb{R} to $\mathbb{R}[x_1, \dots, x_{n-1}]$ is trivial. In order to see the converse, notice that any nonconstant factor of $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ can only belong to $\mathbb{R}[x_1, \dots, x_{n-1}]$ since $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ is irreducible over $\mathbb{R}[x_1, \dots, x_{n-1}]$. On the other hand, the existence of any such a nonconstant factor would contradict with the primitivity of $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ with respect to x_n . Accordingly, $\text{prim}_{x_n}(P(\mathbf{x}^\lambda))$ must be irreducible over \mathbb{R} . \square

Lemma 2.3. *Let $p \in \mathbb{R}[x_1, \dots, x_n]$ and $\lambda \in \mathbb{Z}^n$ with $\gcd(\lambda_1, \dots, \lambda_n) = 1$, $\lambda_1, \dots, \lambda_{n-1}$ not all zero and $\lambda_n > 0$. Then λ is a q -integer linear type of p if and only if there exists a q -primitive polynomial $P(y) \in \mathbb{K}[y]$ such that $P(\mathbf{x}^\lambda)$ divides p in $\mathbb{K}(x_1, \dots, x_{n-1})[x_n]$.*

Proof. The necessity is readily seen from the definition of q -integer linear types. In order to show the sufficiency, it amounts to proving that there exists an irreducible polynomial in $\mathbb{R}[x_1, \dots, x_n]$ of the form $\mathbf{x}^\alpha f(\mathbf{x}^\lambda)$ for $\alpha \in \mathbb{N}^n$ and $f(y) \in \mathbb{R}[y]$ dividing p in $\mathbb{R}[x_1, \dots, x_n]$. Let $f(y) \in \mathbb{R}[y]$ be a primitive irreducible factor of $P(y)$. Since $P(y)$ is q -primitive, we know that $f(y)$ is q -primitive as well. Hence $\text{prim}_{x_n}(f(\mathbf{x}^\lambda)) = \mathbf{x}^\alpha f(\mathbf{x}^\lambda)$ for some $\alpha \in \mathbb{N}^n$ with $\alpha_n = 0$. By Lemma 2.2 (ii), $\text{prim}_{x_n}(f(\mathbf{x}^\lambda))$ is irreducible over \mathbb{R} . Because $P(\mathbf{x}^\lambda)$ divides p in $\mathbb{K}(x_1, \dots, x_{n-1})[x_n]$, so does $f(\mathbf{x}^\lambda)$. It then follows that $\text{prim}_{x_n}(f(\mathbf{x}^\lambda))$ divides p in $\mathbb{R}[x_1, \dots, x_n]$. In other words, $\mathbf{x}^\alpha f(\mathbf{x}^\lambda)$ is an irreducible factor of p in $\mathbb{R}[x_1, \dots, x_n]$. Therefore, λ is a q -integer linear type of p by definition. This concludes the proof. \square

3. The bivariate case

Before turning to the general multivariate case, we first consider the simpler yet important subcase of bivariate polynomials and present a fast algorithm for computing the q -integer linear decomposition in this context. As we will see shortly, this subcase serves as a nice illustration of the main idea of the first approach developed in the next section. Moreover, this subcase can be used as a base case, yielding a second, iterative approach for handling the general multivariate case. This is discussed in Section 5. In this section, we write (x, y) for the two variables (x_1, x_2) .

Let p be a polynomial in $\mathbb{R}[x, y]$. As univariate polynomials are q -integer linear, we may assume without loss of generality that the given polynomial p is nonconstant and it is primitive with respect to x as well as with respect to y , or equivalently, $\text{cont}_x(p) = \text{cont}_y(p) = 1$. With this set-up, p admits the q -integer linear decomposition of the particular form

$$p = x^\alpha P_0 \prod_{i=1}^m P_i(x^{\lambda_i} y^{\mu_i}), \quad (3.1)$$

where $\alpha, m, \mu_i \in \mathbb{N}$, $\lambda_i \in \mathbb{Z}$, $P_0 \in \mathbb{R}[x, y]$ and $P_i(y) \in \mathbb{R}[y]$ with

- (1) P_0 being primitive and having only non- q -integer linear factors except for constants;
- (2) the $P_i(y)$ being nonconstant and q -primitive;
- (3) the (λ_i, μ_i) being distinct q -integer linear types and all $\lambda_i \mu_i \neq 0$.

In order to compute (3.1), following the strategy in the ordinary shift case (Abramov and Le, 2002; Giesbrecht et al., 2019a), we first try to find candidates for all q -integer linear types of p . One way to proceed, according to (Le, 2001), is to compute the resultant of p and its q -shift $p(qx, q^r y)$ with respect to x for an indeterminate r , and then obtain the candidates (λ, μ) by finding possible rationals $r = -\lambda/\mu$ that make the resultant zero. We show below how the problem can be reduced to an interesting and easy geometry task, avoiding computing any resultant.

Observe that, unlike the ordinary shift case, all q -integer linear types (λ_i, μ_i) in (3.1) appear as the exponent vectors, instead of as the coefficients, of p . This observation implies that the homogeneous polynomial technique from (Giesbrecht et al., 2019a) will not work in the q -case. On the other hand, it also suggests that supports of polynomials may play a role similar to homogeneous polynomials in the ordinary shift case. This is exactly the key idea we are describing now.

Lemma 3.1. *Let $p \in \mathbb{R}[x, y] \setminus \mathbb{R}$ with $\text{cont}_x(p) = \text{cont}_y(p) = 1$ and admitting (3.1). Then for any $\ell \in \mathbb{N}$ with $1 \leq \ell \leq m$ and for any pair $(i, j) \in \text{supp}(p)$, there exists another pair $(\tilde{i}, \tilde{j}) \in \text{supp}(p)$ such that $(i, j) = (\tilde{i} + \lambda_\ell k, \tilde{j} + \mu_\ell k)$ for $k \in \mathbb{Z} \setminus \{0\}$, that is, $(i - \tilde{i})/(j - \tilde{j}) = \lambda_\ell/\mu_\ell$.*

Proof. There is nothing to show when $m = 0$. Assume that $m > 0$. It suffices to show the assertion for $\ell = m$ and then the lemma follows by the symmetry.

Let $p^* = x^\alpha P_0 \prod_{i=1}^{m-1} P_i(x^{\lambda_i} y^{\mu_i})$. Then $p^* \in \mathbb{R}[x, y] \setminus \{0\}$ as $p \neq 0$. By (3.1),

$$p = p^* \cdot P_m(x^{\lambda_m} y^{\mu_m}). \quad (3.2)$$

By assumption, $\text{supp}(p) \neq \{\}$. Let $(i, j) \in \text{supp}(p)$. It follows from (3.2) that there is $(i^*, j^*) \in \text{supp}(p^*)$ and $k^* \in \text{supp}(P_m(y))$ such that $(i, j) = (i^* + \lambda_m k^*, j^* + \mu_m k^*)$. Now consider the set

$$S_{(i^*, j^*)} = \{(i', j') \in \text{supp}(p^*) \mid (i', j') = (i^* + \lambda_m k', j^* + \mu_m k') \text{ for some } k' \in \mathbb{Z}\}.$$

Then there exist $p_1^*, p_2^* \in \mathbb{R}[x, y]$ with $\text{supp}(p_1^*) = S_{(i^*, j^*)}$ and $\text{supp}(p_2^*) = \text{supp}(p^*) \setminus S_{(i^*, j^*)}$ such that $p^* = p_1^* + p_2^*$. It is evident that $(i^*, j^*) \in S_{(i^*, j^*)}$. Thus $S_{(i^*, j^*)}$ is a nonempty set in \mathbb{N}^2 and then p_1^* is nonzero. Let (α^*, β^*) be the minimal element in $S_{(i^*, j^*)}$. One then sees from the definition of $S_{(i^*, j^*)}$ that any its element can be written as $(\alpha^* + \lambda_m k', \beta^* + \mu_m k')$ for $k' \in \mathbb{N}$, or equivalently, every monomial appearing in p_1^* has the form $x^{\alpha^* + \lambda_m k'} y^{\beta^* + \mu_m k'}$ for $k' \in \mathbb{N}$. It then follows that there exists a nonzero univariate polynomial $P^*(y) \in \mathbb{R}[y]$ such that $p_1^* = x^{\alpha^*} y^{\beta^*} P^*(x^{\lambda_m} y^{\mu_m})$.

On the other hand, by noticing that for any $(i', j') \in \text{supp}(p_2^*) = \text{supp}(p^*) \setminus S_{(i^*, j^*)}$, we have $(i', j') \neq (i^* + \lambda_m k', j^* + \mu_m k')$ for all $k' \in \mathbb{Z}$. Hence, p can be decomposed as $p = f + g$, where $f = p_1^* P_m(x^{\lambda_m} y^{\mu_m})$ and $g = p_2^* P_m(x^{\lambda_m} y^{\mu_m})$ with $\text{supp}(f) \cap \text{supp}(g) = \{\}$. As a consequence, $\text{supp}(p) = \text{supp}(f) \uplus \text{supp}(g)$. Since $(i^*, j^*) \in S_{(i^*, j^*)}$, we have $(i, j) \in \text{supp}(f)$. Notice that $p_1^* = x^{\alpha^*} y^{\beta^*} P^*(x^{\lambda_m} y^{\mu_m})$. So $f = x^{\alpha^*} y^{\beta^*} \tilde{P}(x^{\lambda_m} y^{\mu_m})$ with $\tilde{P}(y) = P^*(y) P_m(y) \in \mathbb{R}[y] \setminus \{0\}$. Then there exists $k \in \text{supp}(\tilde{P}(y))$ such that $(i, j) = (\alpha^* + \lambda_m k, \beta^* + \mu_m k)$. Since $P_m(y)$ is nonconstant and q -primitive, it has more than one monomial, then so does $\tilde{P}(y)$. This implies that there is another element $\tilde{k} \in \text{supp}(\tilde{P}(y))$ with $\tilde{k} \neq k$. Let $(\tilde{i}, \tilde{j}) = (\alpha^* + \lambda_m \tilde{k}, \beta^* + \mu_m \tilde{k})$. Then the assertion for $\ell = m$ follows by the observation that $(\tilde{i}, \tilde{j}) \in \text{supp}(f) \subset \text{supp}(p)$. \square

The above lemma suggests a simple geometric way to find candidates for all q -integer linear types of the given polynomial p . For each monomial $x^i y^j$ appearing in p plot a point at the coordinate (j, i) on the (y, x) -plane with y -axis the horizontal axis and x -axis the vertical axis, and then determine the lower convex hull C of all these points (namely the convex hull of all vertical rays starting from these points and continuing upwards). We call C the *Newton polygon* of p .

One can refer to (Walker, 1950, Chapter IV) for more information about the Newton polygon. The following basic properties are then geometrically evident.

Lemma 3.2. *With the assumptions of Lemma 3.1, let C be the Newton polygon of p . Then*

- (i) *for each pair $(i, j) \in \text{supp}(p)$, the point (j, i) is contained in C .*
- (ii) *for every vertex (j, i) of C , the pair (i, j) belongs to $\text{supp}(p)$.*
- (iii) *with all points (j, i) with $(i, j) \in \text{supp}(p)$ plotted on the plane, the leftmost (resp. rightmost) vertex of C is one of the leftmost (resp. rightmost) points.*
- (iv) *slopes of the non-vertical edges of C increase from left to right.*
- (v) *each non-vertical edge of C has the lowest possible slope, compared with those line segments inside C which start at the left vertex of the edge and end at any point on the right.*

The key feature of the Newton polygon of p is that the slopes of its non-vertical edges provide all possible choices for the rational numbers λ_i/μ_i , and thus give all candidates for the q -integer linear types (λ_i, μ_i) . Observe that all $\lambda_i/\mu_i \neq 0$. Hence we consider only non-horizontal and non-vertical edges of C , namely those which have nonzero and finite slopes.

Proposition 3.3. *With the assumptions of Lemma 3.1, let C be the Newton polygon of p . Then the slopes of non-horizontal and non-vertical edges of C constitute a superset of all q -integer linear types of p . Moreover, with $s \in \mathbb{N}$ denoting the cardinality of $\text{supp}(p)$, this superset has no more than $s - 1$ elements in total.*

Proof. There is nothing to show when $m = 0$ in (3.1). Assume that $m > 0$. It suffices to show that there exists one edge of C whose slope is equal to λ_m/μ_m , with the rest following from the symmetry and the observation that all λ_i/μ_i are nonzero and finite.

Suppose that none of the edges of C has the slope λ_m/μ_m . By assumption, one sees that p has at least two monomials of distinct powers in y . Thus C has at least one non-vertical edge. Let A_1, \dots, A_{t+1} with $t \in \mathbb{N} \setminus \{0\}$ be all the vertices of C from left to right, and for each $1 \leq \ell \leq t$ let α_ℓ denote the slope of the edge connecting A_ℓ and $A_{\ell+1}$ (as visualized in Figure 1). Then $\alpha_1 < \dots < \alpha_t < \infty$ by Lemma 3.2 (iv) and we have the following case distinction.

Case 1. $\lambda_m/\mu_m < \alpha_1$. Let (j, i) be the coordinate of the leftmost vertex A_1 . Then $(i, j) \in \text{supp}(p)$ by Lemma 3.2 (ii). It follows from Lemma 3.1 that there is $(\tilde{i}, \tilde{j}) \in \text{supp}(p)$ such that $\tilde{j} \neq j$ and $(i - \tilde{i})/(j - \tilde{j}) = \lambda_m/\mu_m < \alpha_1$. Notice that C is convex with A_1 the leftmost vertex. By Lemma 3.2 (iii), the point (\tilde{j}, \tilde{i}) lies outside of C , a contradiction with Lemma 3.2 (i).

Case 2. $\alpha_\ell < \lambda_m/\mu_m < \alpha_{\ell+1}$ for some ℓ with $1 \leq \ell < t$. Let (j, i) be the coordinate of the vertex A_ℓ . A similar argument as Case 1 leads to the contradiction that there exists a point (\tilde{j}, \tilde{i}) with $(\tilde{i}, \tilde{j}) \in \text{supp}(p)$ lying outside of C .

Case 3. $\alpha_t < \lambda_m/\mu_m$. Let (j, i) be the coordinate of the rightmost vertex A_{t+1} . A similar argument as Case 1 leads to the contradiction that there exists a point (\tilde{j}, \tilde{i}) with $(\tilde{i}, \tilde{j}) \in \text{supp}(p)$ lying outside of C . \square

The above discussions provide the following necessary condition for p to be a q -integer linear polynomial.

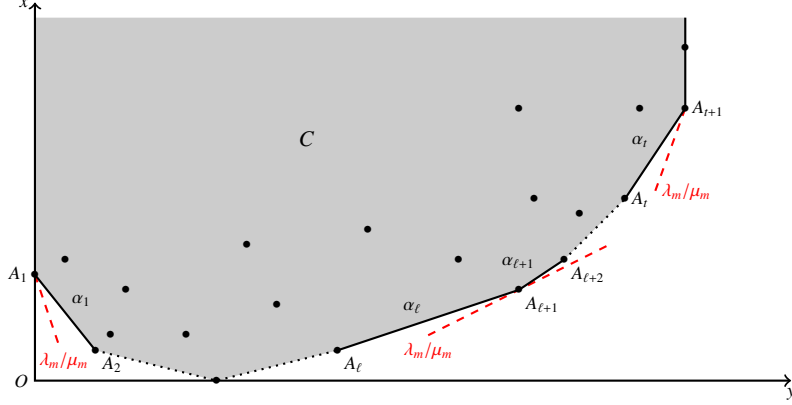


Figure 1: An illustration of the Newton polygon C of p in Proposition 3.3.

Proposition 3.4. *With the assumptions of Lemma 3.1, further assume that p is q -integer linear. Then there is only one monomial appearing in p of lowest or highest degree in y ; a symmetry argument holds in terms of x . Geometrically speaking, this is equivalent to say that, with all points (j, i) corresponding to monomials $x^i y^j$ appearing in p plotted on the (y, x) -plane, there is only one leftmost point, one rightmost point, one lowest point and one highest point. Consequently, the Newton polygon of p does not have horizontal edges.*

Proof. We only show the assertion for monomials of lowest degree in y , corresponding to leftmost points on the (y, x) -plane, with the other three assertions following by similar arguments.

In this respect, it is sufficient to show that if p contains two distinct monomials of lowest degree in y then so does its non- q -integer linear part, namely P_0 in (3.1), because then we know that P_0 is not a unit in \mathbb{R} , and thus obtain the contradiction that p is not q -integer linear.

We proceed by induction on the number m in (3.1). The assertion is evident when $m = 0$, since $p = x^\alpha P_0$ by (3.1). Assume that $m > 0$ and the assertion holds for $m - 1$, namely for any polynomial in $\mathbb{R}[x, y]$ of form (3.1) with m replaced by $m - 1$, if it has two different monomials of lowest degree in y then so does its non- q -integer linear part. Let $j_0 \in \mathbb{N}$ be the lowest degree of p in y and suppose that $x^{i_0} y^{j_0}$ and $x^{i_1} y^{j_0}$ with $i_0 \neq i_1$ in \mathbb{N} are two monomials appearing in p . Let $p^* = x^\alpha P_0 \prod_{i=1}^{m-1} P_i(x^{\lambda_i} y^{\mu_i})$ so that (3.2) holds. It amounts to showing that $x^{i_0} y^{j_0}$ and $x^{i_1} y^{j_0}$ both appear in p^* , which, along with a subsequent application of the induction hypothesis to p^* , concludes the proof. We see from (3.2) that there exist $(i_0^*, j_0^*), (i_1^*, j_1^*) \in \text{supp}(p^*)$ and $k_0, k_1 \in \text{supp}(P_m(y))$ such that

$$j_0 = j_0^* + \mu_m k_0 = j_1^* + \mu_m k_1, \quad i_0 = i_0^* + \lambda_m k_0, \quad i_1 = i_1^* + \lambda_m k_1.$$

Since $\mu_m > 0$ and $P_m(y)$ is q -primitive, we derive from the minimality of j_0 that $k_0 = k_1 = 0$ and then $j_0^* = j_1^* = j_0$. Therefore, $i_0^* = i_0$ and $i_1^* = i_1$. This implies that $(i_0, j_0), (i_1, j_0) \in \text{supp}(p^*)$, that is, $x^{i_0} y^{j_0}$ and $x^{i_1} y^{j_0}$ indeed both appear in p^* . \square

With candidates for the q -integer linear types (λ_i, μ_i) at hand, we are now able to find the corresponding univariate polynomials $P_i(y)$ based on a q -counterpart of (Giesbrecht et al., 2019a, Proposition 3.2).

Proposition 3.5. *With the assumptions of Lemma 3.1, let $(\lambda, \mu) \in \mathbb{Z}^2$ with $\gcd(\lambda, \mu) = 1$, $\lambda \neq 0$ and $\mu > 0$. Let $P^*(y) \in \mathbb{R}[y]$ be the content of the numerator of $p(x^\mu, yx^{-\lambda})$ with respect to x . Then (λ, μ) is a q -integer linear type of p if and only if $P^*(y) \notin \mathbb{R}$. Moreover, if (λ, μ) is a q -integer linear type of p then $P^*(y^{1/\mu}) \in \mathbb{R}[y]$ and it is the corresponding univariate polynomial of the type (λ, μ) .*

Proof. For the necessity of the first assertion, assume that (λ, μ) is a q -integer linear type of p . Then by (3.1), there exists an integer i with $1 \leq i \leq m$ such that $(\lambda_i, \mu_i) = (\lambda, \mu)$, and thus

$$p(x^\mu, yx^{-\lambda}) = \left(x^{\mu\alpha} P_0(x^\mu, yx^{-\lambda}) \prod_{j=1, j \neq i}^m P_j(x^{\lambda_j \mu - \lambda \mu_j} y^{\mu_j}) \right) \cdot P_i(y^\mu) \in \mathbb{R}[x, x^{-1}, y].$$

Notice that $\mu > 0$ and $P_i(y) \in \mathbb{R}[y] \setminus \mathbb{R}$. Therefore one sees from the above equation that $P_i(y^\mu)$ divides all coefficients of $p(x^\mu, yx^{-\lambda})$ with respect to x , giving rise to a desired nontrivial common divisor of $P^*(y)$ in $\mathbb{R}[y]$.

To show the sufficiency, let $f(y) \in \mathbb{K}[y] \setminus \mathbb{K}$ be a monic irreducible factor of $P^*(y)$. Then $f(y)$ divides $p(x^\mu, yx^{-\lambda})$ in $\mathbb{K}(x)[y]$. Substituting y by $yx^{1/\mu}$ and x by $x^{1/\mu}$ yields that $f(yx^{1/\mu})$ divides p in $\mathbb{K}(x)[y]$ with $\mathbb{K}(x)$ the algebraic closure of the field $\mathbb{K}(x)$. This implies that $f(y) \neq y$, for, otherwise, we would have y divides p in $\mathbb{K}(x)[y]$ and then $p(x, 0) = 0$, a contradiction with the primitivity of p with respect to y . Observe that $f(y)$ is a monic irreducible polynomial in $\mathbb{K}[y]$ of positive degree. Let $r \in \mathbb{K}$ be a root of $f(y)$. Then $r \neq 0$ and $f(y)$ is its minimal polynomial in $\mathbb{K}[y]$. Meanwhile, it follows from the divisibility of p by $f(yx^{1/\mu})$ in $\mathbb{K}(x)[y]$ that $y = rx^{-\lambda/\mu}$ is a root of p in $\mathbb{K}(x)$, that is, $p(x, rx^{-\lambda/\mu}) = 0$. Let $P(y) \in \mathbb{K}[y]$ be the minimal polynomial of r^μ . Then $\deg(P(y)) > 0$ and $P(0) \neq 0$ as $r \neq 0$. Also, by Lemma 2.2 (ii), $P(x^\lambda y^\mu)$ is irreducible over $\mathbb{K}(x)$. A simple calculation verifies that $P(x^\lambda y^\mu)$ also vanishes at $y = rx^{-\lambda/\mu}$. Therefore, $P(x^\lambda y^\mu)$, upon making it monic with respect to y , gives rise to the minimal polynomial of $y = rx^{-\lambda/\mu}$ in $\mathbb{K}(x)[y]$. One then derives from $p(x, rx^{-\lambda/\mu}) = 0$ that $P(x^\lambda y^\mu)$ divides p in $\mathbb{K}(x)[y]$. By Lemma 2.3, (λ, μ) is a q -integer linear type of p .

From the preceding paragraph, one can additionally conclude that $f(y)$ divides $P(y^\mu)$ in $\mathbb{K}[y]$, and then by Gauß' lemma, $\text{prim}_y(f(y))$ divides $\text{prim}_y(P(y^\mu))$ in $\mathbb{R}[y]$. Indeed, by the definition of $P(y)$, one sees that $P(r^\mu) = 0$. This in turn means that r is a root of $P(y^\mu)$ in \mathbb{K} . Since $P(y^\mu) \in \mathbb{K}[y]$ and $f(y)$ is the minimal polynomial of r in $\mathbb{K}[y]$, one confirms that $f(y)$ divides $P(y^\mu)$ in $\mathbb{K}[y]$.

It remains to show the last assertion. Notice that p is primitive with respect to y , in particular $p(x, 0) \neq 0$. Then $P^*(y)$ is q -primitive. Since (λ, μ) is a q -integer linear type of p , then $(\lambda, \mu) = (\lambda_i, \mu_i)$ for some integer i with $1 \leq i \leq m$. Thus it amounts to showing that $P^*(y) = P_i(y^\mu)$, which in turn suffices to prove that $P_i(y^\mu)$ divides $P^*(y)$ and conversely in $\mathbb{R}[y]$, because then the assertion directly follows from the fact that both $P^*(y)$ and $P_i(y)$ are q -primitive.

As in the proof of the necessity, one readily sees from (3.1) that $P_i(y^\mu)$ divides $P^*(y)$ in $\mathbb{R}[y]$. In order to show the converse, by noticing that $P^*(y)$ is nonconstant as $\deg(P_i(y)) > 0$, let $f(y) \in \mathbb{R}[y]$ be a primitive irreducible factor of $P^*(y)$ of positive degree in y . From the proof of the sufficiency, one concludes that there exists $P(y) \in \mathbb{K}[y]$ with $P(0) \neq 0$ such that $P(x^\lambda y^\mu)$ divides p in $\mathbb{K}(x)[y]$ and $f(y)$ divides $\text{prim}_y(P(y^\mu))$ in $\mathbb{R}[y]$. By (3.1), $P(x^\lambda y^\mu)$ divides $P_i(x^\lambda y^\mu)$ in $\mathbb{K}(x)[y]$. According to Lemma 2.2 (i), there exists $c \in \mathbb{K}$, $\beta \in \mathbb{Z}$ and a factor $g(y) \in \mathbb{R}[y]$ of $P_i(y)$ such that $P(x^\lambda y^\mu) = cx^\beta g(x^\lambda y^\mu)$. Since $P(0) \neq 0$, we know that $\beta = 0$, that is, $P(y) = cg(y)$. This implies that $\text{prim}_y(P(y))$ divides $P_i(y)$ in $\mathbb{R}[y]$. Therefore, $f(y)$ divides $P_i(y^\mu)$ in $\mathbb{R}[y]$. By the arbitrariness of $f(y)$, one obtains that $P^*(y)$ divides $P_i(y^\mu)$ in $\mathbb{R}[y]$. This completes the proof. \square

Note that the previous proposition also guarantees that any fake candidate for q -integer linear types can be easily recognized by a content computation. Also note that fake candidates only come from the polynomial P_0 in (3.1). Therefore, any occurrence of a trivial content immediately indicates that the given polynomial is not q -integer linear.

Putting things together, we obtain a new algorithm for computing the q -integer linear decomposition of a bivariate polynomial. For compatibility with later algorithms, we take a nonconstant and primitive polynomial as input.

BivariateQILD. Given a nonconstant and primitive polynomial $p \in \mathbb{R}[x, y]$, compute its q -integer linear decomposition.

1. Set $f_1(y) = \text{cont}_x(p)$, $P_0 = \text{prim}_x(p)$, $m = 0$, and set β to be the lowest degree of $f_1(y)$ with respect to y . Update $f_1(y) = f_1(y)/y^\beta$. If $f_1(y) \neq 1$ then update $m = m + 1$, and set $(\lambda_m, \mu_m) = (0, 1)$ and $P_m(y) = f_1(y)$.
2. Set $f_2(x) = \text{cont}_y(P_0)$ and set α to be the lowest degree of $f_2(x)$ with respect to x . Update $P_0 = \text{prim}_y(P_0)$ and $f_2(x) = f_2(x)/x^\alpha$. If $f_2(x) \neq 1$ then update $m = m + 1$, and set $(\lambda_m, \mu_m) = (1, 0)$ and $P_m(x) = f_2(x)$.
3. If $\deg(P_0) = 0$ then return $cx^\alpha y^\beta P_0 \prod_{i=1}^m P_i(x^{\lambda_i} y^{\mu_i})$.
4. Find the support $\text{supp}(P_0) = \{(i_1, j_1), \dots, (i_s, j_s)\} \subset \mathbb{N}^2$ with $(i_1, j_1) < \dots < (i_s, j_s)$.
5. Set $\Lambda = \{\}$ and $k = 1$.
 While $j_k \neq j_s$ do
 - 5.1 Find the maximum $\ell \in \{k + 1, \dots, s\}$ such that $j_\ell \neq j_k$ and $r = \frac{i_\ell - i_k}{j_\ell - j_k}$ is minimal.
 - 5.2 Update $k = \ell$.
 - 5.3 If $r \neq 0$ then update $\Lambda = \Lambda \cup \{(\lambda, \mu) \in \mathbb{Z}^2 \mid \lambda/\mu = r, \gcd(\lambda, \mu) = 1 \text{ and } \mu > 0\}$.
6. For (λ, μ) in Λ do
 - 6.1 Set $P^*(y)$ to be the content of the numerator of $P_0(x^\mu, yx^{-\lambda})$ with respect to x .
 - 6.2 If $\deg(P^*(y)) > 0$ then
 - Update $m = m + 1$, $(\lambda_m, \mu_m) = (\lambda, \mu)$, $P_m(y) = P^*(y^{1/\mu})$.
 - Set $f, g \in \mathbb{R}[x, y]$ to be the numerator and denominator of $P_m(x^\lambda y^\mu)$, and update $P_0 = P_0/f$ and $\alpha = \alpha + \deg_x(g)$.
7. Return $cx^\alpha y^\beta P_0 \prod_{i=1}^m P_i(x^{\lambda_i} y^{\mu_i})$.

Theorem 3.6. Let $p \in \mathbb{R}[x, y]$ be a valid input of the algorithm **BivariateQILD**. Then the algorithm terminates and correctly computes the q -integer linear decomposition of p .

Proof. Notice that the integer k increases every time the algorithm passes through Steps 5.1-5.3. Thus there are only finitely many iterations happening in Step 5, and then the cardinality of the set Λ is finite. Therefore, the algorithm terminates.

In order to prove the correctness, it suffices to show that the set Λ constructed in Step 6 is in one-to-one correspondence with the set comprised of all slopes of non-horizontal and non-vertical edges of the Newton polygon of p , with the rest following from Propositions 3.3 and 3.5. This in turn is evident by Lemma 3.2 (v). \square

Remark 3.7. Step 5 of the above algorithm presents a naive way to construct the Newton polygon of a bivariate polynomial. Fast algorithms are available (cf. (Graham, 1972; Anderson, 1978)). Nevertheless, the cost of this step will not dominate the total cost. Rather than making effort to gain a slight improvement, we prefer to present the computation in such a way that it extends to a higher dimensional space in a straightforward manner (see Section 4).

Remark 3.8. If one is merely interested in the q -integer linearity of the input polynomial $p \in \mathbb{R}[x, y]$, rather than the full q -integer linear decomposition, then our algorithm can be easily modified: any of the following conditions will trigger the adapted algorithm to terminate early, returning that p is not q -integer linear.

- (Proposition 3.4) In Step 4, the support $\text{supp}(p)$ has more than one element whose either entry has the extremum value. Note that this includes the case when $r = 0$ in Step 5.3.
- (Proposition 3.5) In Step 6.2, the case of $\deg(P^*(y)) = 0$ happens, that is, the candidate (λ, μ) currently under investigation is fake.
- (Definition 2.1) In Step 7, we have $\deg(P_0) > 0$.

3.1. Complexity analysis of the bivariate algorithm

Although our algorithms work in more general UFDs, we confine our complexity analysis to the case of integer (Laurent) polynomials, that is, when D is the ring of integers \mathbb{Z} and then R is equal to $\mathbb{Z}[q, q^{-1}]$. Here q can be viewed as a variable in addition to x_1, \dots, x_n . Note that operations in $\mathbb{Z}[q, q^{-1}]$ can be easily transferred to those in $\mathbb{Z}[q]$ with a negligible cost. The cost is given in terms of number of word operations used so that growth of coefficients comes into play. Recall that the *word length* of a nonzero integer $a \in \mathbb{Z}$ is defined as $O(\log |a|)$. In this paper, all complexity is analyzed in terms of O -estimates (or \tilde{O} -estimates) for classical or fast arithmetic, where the *soft-Oh notation* “ \tilde{O} ” is basically “ O ” but suppressing logarithmic factors (see (von zur Gathen and Gerhard, 2013, Definition 25.8) for a precise definition).

Throughout this paper, we define the *max-norm* $\|p\|_\infty$ of a Laurent polynomial $p \in \mathbb{Z}[q, q^{-1}]$ as the maximum absolute value of its coefficients with respect to q , and the *max-norm* $\|p\|_\infty$ of a polynomial $p = \sum_{i \in \mathbb{N}} p_{i_1, \dots, i_n} \mathbf{x}^i \in \mathbb{Z}[q, q^{-1}][x_1, \dots, x_n]$ as $\max_{i \in \mathbb{N}} \{\|p_{i_1, \dots, i_n}\|_\infty\}$. The GCD computation is fundamental for our algorithms. Before analyzing the algorithm, let us recall some useful complexity results on GCD computation.

Lemma 3.9 ((Gel'fond, 1960, Page 135-139)). Let $p_1, \dots, p_m \in \mathbb{Z}[x_1, \dots, x_n]$. Let $p = p_1 \cdots p_m$ and let $d_i = \deg_{x_i}(p)$ for all $i = 1, \dots, n$. Then

$$\|p_1\|_\infty \cdots \|p_m\|_\infty \leq e^{d_1 + \dots + d_n} \|p\|_\infty,$$

where e is the Euler constant. Therefore, $\log \|p_i\|_\infty \in O(d + \log \|p\|_\infty)$ for all $i = 1, \dots, n$ with $d = \max\{d_1, \dots, d_n\}$.

Note that when $n = 1$ the above bound is actually worse than Mignotte's factor bound for large d , which, however, leads to the same order of magnitude for word lengths of the max-norms.

Lemma 3.10. Let $f, g \in \mathbb{Z}[x_1, \dots, x_n]$ with $\deg_{x_i}(f), \deg_{x_i}(g) \leq d_i$ for all $i = 1, \dots, n$ and $\|f\|_\infty, \|g\|_\infty \leq \beta$. Let $d = \max\{d_1, \dots, d_n\}$ and $D_n = d_1 \cdots d_n$. Then computing $\gcd(f, g)$ over \mathbb{Z} takes $\tilde{O}(d^2 D_n + D_n \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d D_n + D_n \log \beta)$ with fast arithmetic.

Proof. We proceed to compute $\gcd(f, g)$ by a small prime modular algorithm. By Lemma 3.9, $\|\gcd(f, g)\|_\infty \leq e^{d_1 + \dots + d_n} \beta \leq e^{nd} \beta = B$ with e the Euler constant. Then $\log B \in O(nd + \log \beta)$. It is sufficient to choose $\lceil \log_2(2B + 1) \rceil$ primes, each of size $O(\log B)$. For every chosen prime h , we then reduce all coefficients of f and g modulo h , using $O(D_n \log \beta \log h)$ word operations with classical arithmetic, and compute $\gcd(f_h, g_h)$ with $f_h = f \bmod h$ and $g_h = g \bmod h$. The desired $\gcd(f, g)$ can be recovered by a final application of the Chinese remainder theorem, which takes $O(d^2 D_n + D_n \log^2 \beta)$ word operations with classical arithmetic and $O(d D_n + D_n \log \beta)$ with fast arithmetic as each $\log h \in O(\log \log B)$ and $n \in O(\log D_n)$. Now it remains to count the number of arithmetic operations, denoted by $G_h(n, d, D_n)$, used by the gcd computation in the field \mathbb{Z}_h for each prime h , with the rest following by the fact that each operation of these takes $O(\log^2 h)$ word operations with classical arithmetic and $O(\log h)$ with fast arithmetic.

For each prime h , we compute $\gcd(f_h, g_h)$ with $f_h = f \bmod h$ and $g_h = g \bmod h$ by an evaluation-interpolation scheme: evaluate coefficients of f_h, g_h with respect to x_1, \dots, x_{n-1} at d_n points from \mathbb{Z}_h for x_n ; compute d_n GCDs over \mathbb{Z}_h of two $(n-1)$ -variate polynomials of degrees at most d_1, \dots, d_{n-1} in x_1, \dots, x_{n-1} , respectively; recover the final GCD by interpolation. Notice that there are at most $d_1 \cdots d_{n-1}$ monomials in x_1, \dots, x_{n-1} appearing in each of f_h and g_h . The evaluation and interpolation steps take $O(d_n D_n)$ arithmetic operations in \mathbb{Z}_h with classical arithmetic and $O(D_n)$ with fast arithmetic. The second step uses $O(d_n G_h(n-1, d, D_{n-1}))$ arithmetic operations in \mathbb{Z}_h , where $D_{n-1} = d_1 \cdots d_{n-1}$. Thus we obtain the recurrence relation

$$O(G_h(n, d, D_n)) \subset O(d_n D_n) + O(d_n G_h(n-1, d, D_{n-1}))$$

with classical arithmetic and

$$O(G_h(n, d, D_n)) \subset O(D_n) + O(d_n G_h(n-1, d, D_{n-1}))$$

with fast arithmetic. From the initial condition that $G_h(1, d_1, d_1)$ is in $O(d_1^2)$ with classical arithmetic and in $O(d_1)$ with fast arithmetic, one concludes that $G_h(n, d, D_n)$ is in $O(nd D_n)$ with classical arithmetic and in $O(D_n)$ with fast arithmetic. \square

We are now ready to present the complexity of our bivariate algorithm. In order to make it ready to use in the analysis of our second approach in Section 5, we analyze the cost in the case of $\mathbf{R} = \mathbb{Z}[q, q^{-1}, z_1, \dots, z_v]$, where $v \in \mathbb{N}$ is arbitrary but fixed and the z_i are additional parameters independent of q, x, y .

Theorem 3.11. *Let $p \in \mathbb{Z}[q, q^{-1}, z_1, \dots, z_v][x, y]$. Assume that the numerator and denominator of p have maximum degree d in each variable from $\{q, z_1, \dots, z_v, x, y\}$ separately and let $\|p\|_\infty = \beta$. Then the algorithm **BivariateQILD** computes the q -integer linear decomposition of p over $\mathbb{Z}[q, q^{-1}, z_1, \dots, z_v]$ using $O(d^{v+8} + d^{v+6} \log^2 \beta)$ word operations with classical arithmetic and $O(d^{v+7} + d^{v+6} \log \beta)$ with fast arithmetic. In particular, when $v = 0$ the algorithm uses $O(d^8 + d^6 \log^2 \beta)$ word operations with classical arithmetic and $O(d^7 + d^6 \log \beta)$ with fast arithmetic.*

Proof. In Step 1, finding the content $f_1(y)$ amounts to computing a GCD of at most $(d+1)$ polynomials in $\mathbb{Z}[q, z_1, \dots, z_v, y]$ of maximum degrees at most d in each variable separately and max-norms at most β . Thus by Lemma 3.10, this step takes $O(d^{v+5} + d^{v+3} \log^2 \beta)$ word operations with classical arithmetic and $O(d^{v+4} + d^{v+3} \log \beta)$ with fast arithmetic. The same cost applies to Step 2. Step 3 is the trivial case and takes no word operations. Step 4 takes linear time in the cardinality s of the support, namely $O(d^2)$ word operations. In Step 5, it is readily seen that each iteration of the loop takes $O((s-k) \log(s-k) \log^2 d)$ word operations with classical arithmetic,

yielding the total cost $O(d^4 \log^3 d)$ word operations with classical arithmetic as there are at most $s \leq (d+1)^2$ iterations. In Step 6, for each $(\lambda, \mu) \in \Lambda$, a direct calculation shows that $P_0(x^\mu, yx^{-\lambda})$ and P_0 have the same degree in y which is at most d , the same max-norm which is of word length $O((v+3)d + \log \beta)$ and the same number of nonzero monomials in x, y appearing which is at most $(d+1)^2$. Thus by Lemma 3.10, Step 6.1 takes $O^\sim(d^{v+6} + d^{v+4} \log^2 \beta)$ word operations with classical arithmetic and $O^\sim(d^{v+5} + d^{v+4} \log \beta)$ with fast arithmetic, which dominates the cost for Step 6.2. Note that we can easily expand $P_0(x^\mu, yx^{-\lambda})$ within the allowed costs. Since there are at most $s-1 \leq (d+1)^2 - 1$ elements in the set Λ , the claimed cost then follows. \square

Remark 3.12. *If one finds a multivariate version of the algorithm of Conflitti (2003), then the complexity can be further improved.*

4. The first approach for the multivariate case

In this and the next section, we will deal with multivariate polynomials having more than two variables. We propose two approaches to compute the q -integer linear decomposition of such a polynomial. The first one extends the bivariate algorithm introduced in the preceding section to the multivariate case in a straightforward manner, which is discussed in the present section. The second approach takes the bivariate algorithm as a base case and proceeds in an iterative fashion by tackling two variables at each iteration. This is explored in the next section. Unlike the ordinary shift case in (Giesbrecht et al., 2019a), both approaches appear to perform well in practice as we shall see in Section 7.

Let us now start with the first approach. This method follows exactly the same pattern as the bivariate algorithm given in Section 3: first find all possible candidates for q -integer linear types of the given polynomial by means of geometry; then extract the corresponding univariate polynomials for the types one by one via content computations. All results presented below can be shown along almost the same lines as in the bivariate case but from a multivariate point of view, so we will omit the proofs.

Let $p \in \mathbb{R}[x_1, \dots, x_n]$ and assume that it admits the q -integer linear decomposition (2.1). By iteratively removing the content of p with respect to each variable from $\{x_1, \dots, x_n\}$ and recursively computing the q -integer linear decomposition of every removed content which is an $(n-1)$ -variate polynomial over \mathbb{R} , we may further assume without loss of generality that p is nonconstant and primitive with respect to each variable from $\{x_1, \dots, x_n\}$. This implies that in the equation (2.1), we have $c = 1$, $\alpha_n = 0$ and none of the types λ_i has zero entries.

In order to find the candidates, one shows the following multivariate version of Lemma 3.1.

Lemma 4.1. *Let $p \in \mathbb{R}[x_1, \dots, x_n] \setminus \mathbb{R}$ with $\text{cont}_{x_1}(p) = \dots = \text{cont}_{x_n}(p) = 1$. Then for any $j \in \mathbb{N}$ with $1 \leq j \leq m$ and for any $\mathbf{i} \in \text{supp}(p)$, there exists another n -vector $\tilde{\mathbf{i}} \in \text{supp}(p)$ such that $\mathbf{i} = \tilde{\mathbf{i}} + k \cdot \lambda_j$ for some $k \in \mathbb{Z} \setminus \{0\}$.*

By the same reasoning as in the bivariate case, the problem is reduced to computing “slopes” of edges of the *Newton polytope* associated to the given polynomial p . By the Newton polytope associated to p , we mean the lower convex hull (in view of the x_1 -axis) in the (x_1, \dots, x_n) -space of all points corresponding to vectors in $\text{supp}(p)$, whose edges are comprised of line segments connecting the minimal vector to the maximal one in $\text{supp}(p)$ in terms of the total ordering “ $<$ ” on \mathbb{N}^n . And the “slope” of the line connecting two points $\mathbf{i}, \mathbf{j} \in \mathbb{N}^n$ in the (x_1, \dots, x_n) -space is referred to as the rational vector $(\frac{i_1-j_1}{i_n-j_n}, \dots, \frac{i_{n-1}-j_{n-1}}{i_n-j_n}, 1) \in \mathbb{Q}^n$.

Proposition 4.2. *With the assumptions of Lemma 4.1, the finite “slopes” of edges of the Newton polytope associated to p which have no zero entries constitute a superset of all q -integer linear types of p . Moreover, with $s \in \mathbb{N}$ denoting the cardinality of $\text{supp}(p)$, this superset has no more than $s - 1$ elements in total.*

Similar to Proposition 3.4, we obtain a necessary condition for p to be q -integer linear.

Proposition 4.3. *With the assumptions of Lemma 4.1, if p is q -integer linear, then the extremum value of every variable from $\{x_1, \dots, x_n\}$ amongst the vectors of $\text{supp}(p)$ can only be attained once. As a consequence, all “slopes” of edges of the Newton polytope associated to p have no zero entries.*

In order to generalize Step 5 of the bivariate algorithm to the multivariate case, we extend the total ordering “ $<$ ” on \mathbb{N}^n to \mathbb{Q}^n as follows. Let $\mathbf{i}, \mathbf{j} \in \mathbb{Q}^n$, we say that $\mathbf{i} < \mathbf{j}$ if the rightmost nonzero entry of the vector difference $\mathbf{i} - \mathbf{j} \in \mathbb{Q}^n$ is negative. Due to its convexity, the construction of the Newton polytope associated to p can be visualized in the following steps:

- (i) plot a point at the coordinate \mathbf{i} in the (x_1, \dots, x_n) -space for every vector $\mathbf{i} \in \text{supp}(p)$;
- (ii) take the point corresponding to the minimal element in $\text{supp}(p)$ and denote it by A ;
- (iii) draw a line segment from A to every point strictly on its right in view of the x_n -axis;
- (iv) find the line segment of lowest possible “slope”, yielding one edge of the Newton polytope;
- (v) update A to be the other end point of this edge;
- (vi) repeat (iii)-(v) until A reaches the rightmost point in view of the x_n -axis.

The above description, together with Proposition 4.2, provides a natural way to compute candidates for q -integer linear types of p . In analogy to Proposition 3.5, we have the following multivariate criterion for detecting the genuineness of each candidate, along with the computation of the corresponding univariate polynomial when the answer is affirmative.

Proposition 4.4. *With the assumptions of Lemma 4.1, let $\lambda \in \mathbb{Z}^n$ with $\gcd(\lambda_1, \dots, \lambda_n) = 1$, $\lambda_1, \dots, \lambda_{n-1}$ not all zero and $\lambda_n > 0$. Let $P^*(y) \in \mathbb{R}[y]$ be the content of the numerator of $p(x_1^{\lambda_1}, \dots, x_{n-1}^{\lambda_{n-1}}, yx_1^{-\lambda_1} \dots x_{n-1}^{-\lambda_{n-1}})$ with respect to x_1, \dots, x_{n-1} . Then λ is a q -integer linear type of p if and only if $P^*(y) \notin \mathbb{R}$. Moreover, if λ is a q -integer linear type of p then $P^*(y^{1/\lambda_n}) \in \mathbb{R}[y]$ and it is the corresponding univariate polynomial of the type λ .*

Assembling everything together yields our first approach.

MultivariateQILD₁. Given a polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$, compute its q -integer linear decomposition.

1. If $p \in \mathbb{R}$ then set $c = p$; and return c .
2. Set $c = \text{cont}(p)$ and $f = \text{prim}(p)$. If $\text{supp}(f)$ is a singleton then set α to be the only element and update $c = cf/x^\alpha$; and return cx^α .
3. If $n = 1$ then set α_1 to be the lowest degree of f with respect to x_1 , $m = 1$, $\lambda_{m1} = 1$ and $P_m(y) = f(y)/y^{\alpha_1}$; and return $c x_1^{\alpha_1} \prod_{i=1}^m P_i(x_1^{\lambda_{i1}})$.

4. Set $\alpha = \mathbf{0}$, $P_0 = 1$, $m = 0$.

For $i = 1, \dots, n$ do

4.1 Set $g = \text{cont}_{x_i}(f)$, and update $f = \text{prim}_{x_i}(f)$.

4.2 If $g \neq 1$ then call the algorithm recursively with input $g \in \mathbb{R}[x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$, returning

$$g = x_1^{\tilde{\alpha}_1} \cdots x_{i-1}^{\tilde{\alpha}_{i-1}} x_{i+1}^{\tilde{\alpha}_{i+1}} \cdots x_n^{\tilde{\alpha}_n} \tilde{P}_0 \prod_{j=1}^{\tilde{m}} \tilde{P}_j(x_1^{\tilde{\lambda}_{j1}} \cdots x_{i-1}^{\tilde{\lambda}_{ji-1}} x_{i+1}^{\tilde{\lambda}_{ji+1}} \cdots x_n^{\tilde{\lambda}_{jn}}),$$

update $\alpha = \alpha + (\tilde{\alpha}_1, \dots, \tilde{\alpha}_{i-1}, 0, \tilde{\alpha}_{i+1}, \dots, \tilde{\alpha}_n)$, $P_0 = P_0 \tilde{P}_0$, and for $j = 1, \dots, \tilde{m}$ iteratively update $m = m + 1$, $\lambda_m = (\tilde{\lambda}_{j1}, \dots, \tilde{\lambda}_{ji-1}, 0, \tilde{\lambda}_{ji+1}, \dots, \tilde{\lambda}_{jn})$, $P_m(y) = \tilde{P}_j(y)$.

5. If $\deg(f) = 0$ then update $c = cf$; and return $c \mathbf{x}^\alpha P_0 \prod_{i=1}^m P_i(\mathbf{x}^{\lambda_i})$.

6. Find the support $\text{supp}(f) = \{i_1, \dots, i_s\}$ with $i_1 < \dots < i_s$.

7. Set $\Lambda = \{\}$ and $k = 1$.

While $i_{kn} \neq i_{sn}$ do

7.1 Find the maximum $j \in \{k+1, \dots, s\}$ such that $i_{jn} \neq i_{kn}$ and $\mathbf{r} = \left(\frac{i_{j1}-i_{k1}}{i_{jn}-i_{kn}}, \dots, \frac{i_{jn-1}-i_{kn-1}}{i_{jn}-i_{kn}}, 1 \right)$ is minimal.

7.2 Update $k = j$.

7.3 If \mathbf{r} has no zero entries then update

$$\Lambda = \Lambda \cup \{\lambda \in \mathbb{Z}^n \mid (\lambda_1/\lambda_n, \dots, \lambda_{n-1}/\lambda_n, 1) = \mathbf{r}, \gcd(\lambda_1, \dots, \lambda_n) = 1, \lambda_n > 0\}.$$

8. For λ in Λ do

8.1 Set $P^*(y)$ to be the content of the numerator of $f(x_1^{\lambda_n}, \dots, x_{n-1}^{\lambda_n}, yx_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})$ with respect to x_1, \dots, x_{n-1} .

8.2 If $\deg(P^*(y)) > 0$ then

Update $m = m + 1$, $\lambda_m = \lambda$, $P_m(y) = P^*(y^{1/\lambda_n})$.

Set $f^*, g^* \in \mathbb{R}[x_1, \dots, x_n]$ to be the numerator and denominator of $P_m(\mathbf{x}^\lambda)$, and update $f = f/f^*$ and $\alpha_i = \alpha_i + \deg_{x_i}(g^*)$ for $i = 1, \dots, n-1$.

9. If $\deg(f) > 0$ then update $P_0 = P_0 f$ else update $c = cf$.

10. Return $c \mathbf{x}^\alpha P_0 \prod_{i=1}^m P_i(\mathbf{x}^{\lambda_i})$.

Theorem 4.5. Let $p \in \mathbb{R}[x_1, \dots, x_n]$. Then the algorithm **MultivariateQILD**₁ terminates and correctly computes the q -integer linear decomposition of p .

Proof. This is evident by Propositions 4.2, 4.4 and the discussions in between. \square

Remark 4.6. If one is merely interested in the q -integer linearity of the input polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$, rather than the full q -integer linear decomposition, then the above algorithm can be easily modified. Analogous to Remark 3.8, any of the following conditions will trigger the adapted algorithm to terminate early, returning that p is not q -integer linear.

- In Step 4.2, each polynomial g turns out to be non- q -integer linear.

- (Proposition 4.3) In Step 6, the support $\text{supp}(f)$ has more than one element whose either entry has the extremum value. Note that this includes the case when the integer vector \mathbf{r} has a zero entry in Step 7.3.
- (Proposition 4.4) In Step 8.2, the case of $\deg(P^*(y)) = 0$ happens, that is, the candidate λ currently under investigation is fake.
- (Definition 2.1) In Step 10, we have $\deg(P_0) > 0$.

Theorem 4.7. Let $p \in \mathbb{Z}[q, q^{-1}][x_1, \dots, x_n]$. Assume that the numerator and denominator of p have maximum degree d in each variable from $\{q, x_1, \dots, x_n\}$ separately and let $\|p\|_\infty = \beta$. Then the algorithm **MultivariateQILD₁** computes the q -integer linear decomposition of p over \mathbb{Z} using $\tilde{O}(n^2 d^{2n+4} + n^2 d^{2n+2} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(n^2 d^{2n+3} + n^2 d^{2n+2} \log \beta)$ with fast arithmetic.

Proof. Let $T(n, d, \log \beta)$ denote the number of word operations used by the algorithm applied to p . Steps 1 and 5 treat the trivial case, taking no word operations. In Step 2, finding the content c amounts to computing a GCD of at most $(d+1)^n$ polynomials in $\mathbb{Z}[q]$ of degrees in q at most d and max-norms at most β . Thus by Lemma 3.10, this step takes $\tilde{O}(d^{n+3} + d^{n+1} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+2} + d^{n+1} \log \beta)$ with fast arithmetic. Step 3 deals with the univariate case, yielding that the initial cost $T(1, d, \log \beta)$ is in $\tilde{O}(d^4 + d^2 \log^2 \beta)$ with classical arithmetic and $\tilde{O}(d^3 + d^2 \log \beta)$ with fast arithmetic.

In Step 4, at each iteration of the loop, the computation of the content g and its primitive part in Step 4.1 can be done within $\tilde{O}(d^{n+3} + d^{n+1} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+2} + d^{n+1} \log \beta)$ with fast arithmetic; while Step 4.2 takes $\tilde{O}(T(n-1, d, nd + \log \beta))$ word operations as $g \in \mathbb{Z}[q, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$ of maximum degree at most d in each variable and max-norm of word length $\tilde{O}(nd + \log \beta)$ by Lemma 3.9. Since there are n iterations, this step in total takes $\tilde{O}(d^{n+3} + d^{n+1} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+2} + d^{n+1} \log \beta)$ with fast arithmetic, plus $\tilde{O}(nT(n-1, d, nd + \log \beta))$ word operations.

Step 6 takes linear time in the cardinality s of the support, namely $\tilde{O}(d^n)$ word operations. In Step 7, each iteration of the loop requires $\tilde{O}((s-k) \log(s-k)(n-1) \log^2 d)$ word operations with classical arithmetic, yielding the total cost $\tilde{O}(d^{2n} n^2 \log^3 d)$ word operations with classical arithmetic as there are at most $s \leq (d+1)^n$ iterations. In Step 8, for each $\lambda \in \Lambda$, a direct calculation shows that $f(x_1^{\lambda_1}, \dots, x_{n-1}^{\lambda_{n-1}}, y x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})$ and f have the same degree in y which is at most d , the same max-norm which is of word length $\tilde{O}(nd + \log \beta)$ and the same number of nonzero monomials in x_1, \dots, x_{n-1}, y appearing which is at most $(d+1)^n$. Thus by Lemma 3.10, Step 8.1 takes $\tilde{O}(d^{n+4} + d^{n+2} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+3} + d^{n+2} \log \beta)$ with fast arithmetic, which dominates the cost for Step 8.2. Since there are at most $s-1 \leq (d+1)^n - 1$ elements in the set Λ , this step takes $\tilde{O}(d^{2n+4} + d^{2n+2} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{2n+3} + d^{2n+2} \log \beta)$ with fast arithmetic. Steps 9 and 10 both take no word operations without expanding the product.

In summary, we obtain the recurrence relation

$$\tilde{O}(T(n, d, \log \beta)) \subset \tilde{O}(d^{2n+4} + d^{2n+2} \log^2 \beta) + \tilde{O}(nT(n-1, d, nd + \log \beta)),$$

along with $T(1, d, \log \beta) \in \tilde{O}(d^4 + d^2 \log^2 \beta)$ with classical arithmetic or

$$\tilde{O}(T(n, d, \log \beta)) \subset \tilde{O}(d^{2n+3} + d^{2n+2} \log \beta) + \tilde{O}(nT(n-1, d, nd + \log \beta)),$$

along with $T(1, d, \log \beta) \in \tilde{O}(d^3 + d^2 \log \beta)$ with fast arithmetic. The announced cost follows. \square

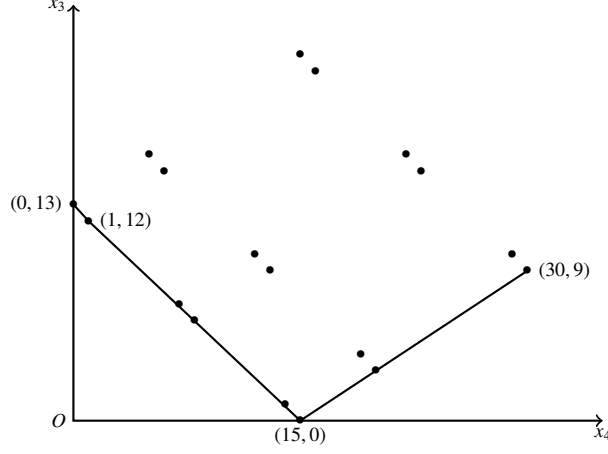


Figure 2: Newton polygon of p in Example 4.8 when viewed as a polynomial in x_3, x_4 .

Example 4.8. Consider the polynomial $p \in \mathbb{Z}[q, q^{-1}][x_1, x_2, x_3, x_4]$ of the form

$$\begin{aligned}
p = & 2q^2 x_1^9 x_2^{12} x_3^{13} + 2qx_1^8 x_2^{14} x_3^{13} + 2qx_1^8 x_2^{14} x_3^{12} x_4 + 18q^2 x_1^{11} x_2^8 x_3^{16} x_4^5 + 18qx_1^{10} x_2^{10} x_3^{16} x_4^5 \\
& + 18qx_1^{10} x_2^{10} x_3^{15} x_4^6 - 2qx_1^5 x_2^{20} x_3^7 x_4^7 - 2x_1^4 x_2^{22} x_3^7 x_4^7 - 2x_1^4 x_2^{22} x_3^6 x_4^8 - 18qx_1^7 x_2^{16} x_3^{10} x_4^{12} \\
& - 18x_1^6 x_2^{18} x_3^{10} x_4^{12} - 18x_1^6 x_2^{18} x_3^9 x_4^{13} + 7q^2 x_1 x_2^{28} x_3 x_4^{14} + 7qx_2^{30} x_3 x_4^{14} + 7qx_2^{30} x_4^{15} \\
& + 6q^4 x_1^{15} x_3^{22} x_4^{15} + 6q^3 x_1^{14} x_2^2 x_3^{22} x_4^{15} + 6q^3 x_1^{14} x_2^2 x_3^{21} x_4^{16} + 63q^2 x_1^3 x_2^{24} x_3^4 x_4^{19} \\
& + 63qx_1^2 x_2^{26} x_3^4 x_4^{19} + 63qx_1^2 x_2^{26} x_3^3 x_4^{20} - 6q^3 x_1^{11} x_2^8 x_3^{16} x_4^{22} - 6q^2 x_1^{10} x_2^{10} x_3^{16} x_4^{22} \\
& - 6q^2 x_1^{10} x_2^{10} x_3^{15} x_4^{23} + 21q^4 x_1^7 x_2^{16} x_3^{10} x_4^{29} + 21q^3 x_1^6 x_2^{18} x_3^{10} x_4^{29} + 21q^3 x_1^6 x_2^{18} x_3^9 x_4^{30}
\end{aligned} \tag{4.1}$$

In order to compute the q -integer linear decomposition of the polynomial p over $\mathbb{Z}[q, q^{-1}]$, the algorithm **MultivariateQILD**₁ first tries to find candidates for all possible q -integer linear types of p . In this respect, it computes the support of p , which can be readily read out from (4.1):

$$\begin{aligned}
& \{ (9, 12, 13, 0) < (8, 14, 13, 0) < (8, 14, 12, 1) < (11, 8, 16, 5) < (10, 10, 16, 5) < (10, 10, 15, 6) \\
& < (5, 20, 7, 7) < (4, 22, 7, 7) < (4, 22, 6, 8) < (7, 16, 10, 12) < (6, 18, 10, 12) < (6, 18, 9, 13) \\
& < (1, 28, 1, 14) < (0, 30, 1, 14) < (0, 30, 0, 15) < (15, 0, 22, 15) < (14, 2, 22, 15) < (14, 2, 21, 16) \\
& < (3, 24, 4, 19) < (2, 26, 4, 19) < (2, 26, 3, 20) < (11, 8, 16, 22) < (10, 10, 16, 22) \\
& < (10, 10, 15, 23) < (7, 16, 10, 29) < (6, 18, 10, 29) < (6, 18, 9, 30) \}.
\end{aligned}$$

From above, we can already tell from Proposition 4.3 that the given polynomial p is not q -integer linear, since there are two elements of $\text{supp}(p)$ (namely the first two elements) attaining the minimum value of x_4 .

By the definition of the ordering “ $<$ ” on \mathbb{Q}^n , one readily sees that the projection of the Newton polytope associated to p on the (x_4, x_3) -plane coincides with the Newton polygon of p when viewed as a polynomial in x_3, x_4 . The latter is depicted in Figure 2, revealing that the (x_4, x_3) -coordinates of all its vertices are given by $(0, 13)$, $(1, 12)$, $(15, 0)$, $(30, 9)$. Notice that each of the

last three points gives rise to a unique element in $\text{supp}(p)$. It then follows that the following four points:

$$(9, 12, 13, 0), (8, 14, 12, 1), (0, 30, 0, 15), (6, 18, 9, 30)$$

exhaust all vertices of the Newton polytope associated to p in the (x_1, x_2, x_3, x_4) -space. This thus yields three candidates for q -integer linear types of p , namely

$$(-1, 2, -1, 1), (2, -4, 3, 5), (-4, 8, 6, 7).$$

A subsequent content computation for each candidate leads to the following q -integer linear decomposition

$$p = x_1^8 x_2^{12} x_3^{12} \cdot P_0 \cdot P_1(x_1^2 x_2^{-4} x_3^3 x_4^5) \cdot P_2(x_1^{-4} x_2^8 x_3^{-6} x_4^7), \quad (4.2)$$

where $P_0 = qx_1x_3 + x_2^2x_3 + x_2^2x_4$, $P_1(y) = 3q^2y^3 + qy + 1$ and $P_2(y) = 7qy^2 - 2y + 2q$. Note that the candidate $(-1, 2, -1, 1)$ is fake, implying, once again, that the given polynomial p is not q -integer linear.

5. The second approach for the multivariate case

This section presents the second approach for computing the q -integer linear decomposition of a polynomial in an arbitrary number of variables. In order to describe it, we need a q -analogue of (Abramov and Petkovšek, 2002, Proposition 7). To this end, we require two technical lemmas. The first one corresponds to (Abramov and Petkovšek, 2002, Lemma 2) but restricted to the case of Laurent polynomials.

Lemma 5.1. *Let $p \in \mathbb{R}[x, x^{-1}]$ be a nonzero Laurent polynomial. If there exists a nonzero integer a and a nonzero element $c \in \mathbb{R}$ such that $p(q^a x) = cp(x)$, then $c = q^{am}$ for some $m \in \mathbb{Z}$ and $p(x)/x^m \in \mathbb{R}$.*

Proof. Notice that p is nonzero. Write $p(x) = x^m f(x)$ for $m \in \mathbb{Z}$ and $f \in \mathbb{R}[x]$ with $f(0) \neq 0$. It follows from $p(q^a x) = cp(x)$ that $q^{am} f(q^a x) = cf(x)$. Letting $x = 0$ in the equation yields that $c = q^{am}$ as $f(0) \neq 0$. Hence $f(q^a x) = f(x)$. Let x_0 be a nonzero element in \mathbb{R} . Then $f(x_0) \in \mathbb{R}$. By induction on k , we see that $f(q^{ka} x_0) = f(x_0)$ for all $k \in \mathbb{N}$. Let $g(x) = f(x) - f(x_0)$. Then $g \in \mathbb{R}[x]$ and it vanishes on $\{q^{ka} x_0 \mid k \in \mathbb{N}\}$, which is an infinity set in \mathbb{R} since the characteristic of \mathbb{R} is zero and $q \in \mathbb{R}$ is an indeterminate. Therefore g is the zero polynomial and thus $f(x) = f(x_0) \in \mathbb{R}$. The lemma follows. \square

Evidently, in the above lemma, the ring \mathbb{R} can be replaced by any of its ring extensions which is independent of the variable x . The next lemma plays the role of (Abramov and Petkovšek, 2001, Lemma 3), or identically, (Hou, 2004, Lemma 3.3), in the q -shift setting, which describes a nice structure of q -shift invariant bivariate polynomials.

Lemma 5.2. *Let $p \in \mathbb{R}[x, y]$. If there exists $c \in \mathbb{R}$ and $a, b \in \mathbb{Z}$, not both zero, such that $p(q^a x, q^b y) = cp(x, y)$, then there is a univariate polynomial $P(y) \in \mathbb{R}[y]$ and four integers $\alpha, \beta, \lambda, \mu$ with λ, μ not both zero such that $p = x^\alpha y^\beta P(x^\lambda y^\mu)$; and conversely.*

Proof. There is nothing to show if $p = 0$. Now assume that p is nonzero and we adapt the idea from the proof of (Hou, 2004, Lemma 3.3) into the current setting. Let $d = \gcd(a, b)$. Then $d \neq 0$ since a, b are not both zero. Thus letting $\lambda = -b/d$ and $\mu = a/d$ gives us two coprime integers λ, μ . By Bézout's relation, there exist $s, t \in \mathbb{Z}$ such that $s\lambda + t\mu = 1$. Now define

$f(x, y) = p(x^\mu y^s, x^{-\lambda} y^t)$. Then f is a nonzero Laurent polynomial in the ring $\mathbb{R}[x, x^{-1}, y, y^{-1}]$ and $p(x, y) = f(x^t y^{-s}, x^\lambda y^\mu)$ by Bézout's relation. Using $p(q^a x, q^b y) = c p(x, y)$, a direct calculation shows that $f(q^d x, y) = c f(x, y)$. Since $d \neq 0$, applying Lemma 5.1 to f with \mathbb{R} replaced by $\mathbb{R}[y, y^{-1}]$ and extracting a nonpositive power of y yields that $c = q^{dm}$ for $m \in \mathbb{Z}$ and $f(x, y) = x^m y^{-k} P(y)$ for $k \in \mathbb{N}$ and $P(y) \in \mathbb{R}[y]$. Therefore, $p = x^{tm-\lambda k} y^{-sm-\mu k} P(x^\lambda y^\mu)$. The assertion follows by letting $\alpha = tm - \lambda k$ and $\beta = -sm - \mu k$. The converse argument is evident by setting $a = \mu$ and $b = -\lambda$. This completes the proof. \square

From the above lemma, we are then able to establish the fact that the problem of multivariate q -integer linearity is actually made up of a collection of subproblems of bivariate q -integer linearity.

Proposition 5.3. *Let $p \in \mathbb{R}[x_1, \dots, x_n]$. Then there exists a univariate polynomial $P(y) \in \mathbb{R}[y]$ and two vectors $\alpha \in \mathbb{N}^n$, $\lambda \in \mathbb{Z}^n \setminus \{0\}$ such that $p = \mathbf{x}^\alpha P(\mathbf{x}^\lambda)$ if and only if for each pair (i, j) with $1 \leq i < j \leq n$, there is a polynomial $P_{ij}(y) \in \mathbb{R}[x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n][y]$ and four integers $\beta_{ij}, \beta_{ji}, \mu_{ij}, \mu_{ji}$ with μ_{ij}, μ_{ji} not both zero such that $p = x_i^{\beta_{ij}} x_j^{\beta_{ji}} P_{ij}(x_i^{\mu_{ij}} x_j^{\mu_{ji}})$.*

Proof. The necessity is clear. For the sufficiency, we proceed by induction on the number n of variables. There is nothing to show in the base case where $n = 1$. Assume that $n > 1$ and the assertion holds for $n - 1$.

Consider p as a polynomial in x_1, \dots, x_{n-1} over $\mathbb{R}[x_n]$. By the induction hypothesis, there is a polynomial $P^*(y) \in \mathbb{R}[x_n][y]$ and two vectors $(\alpha_1^*, \dots, \alpha_{n-1}^*) \in \mathbb{N}^{n-1}$, $(\lambda_1^*, \dots, \lambda_{n-1}^*) \in \mathbb{Z}^{n-1}$ with the λ_i^* not all zero such that

$$p(x_1, \dots, x_{n-1}) = x_1^{\alpha_1^*} \cdots x_{n-1}^{\alpha_{n-1}^*} P^*(x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}).$$

We may assume without loss of generality that $\lambda_1^* \neq 0$. Regarding $P^*(y)$ as an element of $\mathbb{R}[y, x_n]$, we rewrite the preceding equation as

$$p(x_1, \dots, x_n) = x_1^{\alpha_1^*} \cdots x_{n-1}^{\alpha_{n-1}^*} P^*(x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}, x_n). \quad (5.1)$$

By taking $i = 1$ and $j = n$ in the assumption, we know that $p = x_1^{\beta_{1n}} x_n^{\beta_{n1}} P_{1n}(x_1^{\mu_{1n}} x_n^{\mu_{n1}})$ for $P_{1n}(y) \in \mathbb{R}[x_2, \dots, x_{n-1}][y]$ and $\beta_{1n}, \beta_{n1}, \mu_{1n}, \mu_{n1} \in \mathbb{Z}$ with μ_{1n}, μ_{n1} not both zero. Therefore,

$$p(q^{\mu_{n1}} x_1, x_2, \dots, x_{n-1}, q^{-\mu_{1n}} x_n) = c p(x_1, \dots, x_n) \quad \text{with } c = q^{\beta_{1n}\mu_{n1} - \beta_{n1}\mu_{1n}} \in \mathbb{R}.$$

It then follows from (5.1) that $P^*(q^{\mu_{n1}\lambda_1^*} x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}, q^{-\mu_{1n}} x_n) = c q^{-\mu_{n1}\alpha_1^*} P^*(x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}, x_n)$, that is,

$$P^*(q^{\mu_{n1}\lambda_1^*} y, q^{-\mu_{1n}} x_n) = c q^{-\mu_{n1}\alpha_1^*} P^*(y, x_n).$$

Applying Lemma 5.2 to $P^*(y, x_n)$ yields that there is a univariate polynomial $P(y) \in \mathbb{R}[y]$ and four integers $\alpha_n, \alpha_n^*, \lambda_n, \lambda_n^*$ with λ_n, λ_n^* not both zero such that $P^*(y, x_n) = y^{\alpha_n^*} x_n^{\alpha_n} P(y^{\lambda_n^*} x_n^{\lambda_n})$. Substituting $y = x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}$ into this equation, together with (5.1), implies that $p = \mathbf{x}^\alpha P(\mathbf{x}^\lambda)$ with $\alpha = (\alpha_1^* + \lambda_1^* \alpha_n^*, \dots, \alpha_{n-1}^* + \lambda_{n-1}^* \alpha_n^*, \alpha_n)$ and $\lambda = (\lambda_1^* \lambda_n^*, \dots, \lambda_{n-1}^* \lambda_n^*, \lambda_n)$. The proof is concluded by noticing that $\lambda \neq 0$. \square

Inspired by the above proposition, we propose an algorithm which takes a multivariate polynomial as input and computes its q -integer linear decomposition in an iterative fashion so that at

each iteration step, only two variables enter the game whereas the others are treated as coefficient parameters.

MultivariateQILD₂. Given a polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$, compute its q -integer linear decomposition.

1. If $p \in \mathbb{R}$ then set $c = p$; and return c .
2. Set $c = \text{cont}(p)$ and $f = \text{prim}(p)$. If $\text{supp}(f)$ is a singleton then set α to be the only element and update $c = cf/x^\alpha$; and return cx^α .
3. If $n = 1$ then set α_1 to be the lowest degree of f with respect to x_1 , $m = 1$, $\lambda_{m1} = 1$ and $P_m(y) = f(y)/y^{\alpha_1}$; and return $c x_1^{\alpha_1} \prod_{i=1}^m P_i(x_1^{\lambda_{i1}})$.
4. If $n = 2$ then call the algorithm **BivariateQILD** with input $f \in \mathbb{R}[x_1, x_2]$ to compute its q -integer linear decomposition

$$f = x_1^{\alpha_1} x_2^{\alpha_2} P_0 \prod_{i=1}^m P_i(x_1^{\lambda_{i1}} x_2^{\lambda_{i2}});$$

and then return $c x_1^{\alpha_1} x_2^{\alpha_2} P_0 \prod_{i=1}^m P_i(x_1^{\lambda_{i1}} x_2^{\lambda_{i2}})$.

5. Set $\alpha = \mathbf{0}$, $P_0 = 1$, $m = 0$ and $g = \text{cont}_{x_1, x_2}(f)$, and update $f = \text{prim}_{x_1, x_2}(f)$.
6. If $g \neq 1$ then call the algorithm recursively with input $g \in \mathbb{R}[x_3, \dots, x_n]$, returning

$$g = x_3^{\tilde{\alpha}_3} \cdots x_n^{\tilde{\alpha}_n} \tilde{P}_0 \prod_{i=1}^{\tilde{m}} \tilde{P}_i(x_3^{\tilde{\lambda}_{i3}} \cdots x_n^{\tilde{\lambda}_{in}}),$$

update $\alpha = \alpha + (0, 0, \tilde{\alpha}_3, \dots, \tilde{\alpha}_n)$, $P_0 = P_0 \tilde{P}_0$, and for $i = 1, \dots, \tilde{m}$ iteratively update $m = m + 1$, $\lambda_m = (0, 0, \tilde{\lambda}_{i3}, \dots, \tilde{\lambda}_{in})$, $P_m(y) = \tilde{P}_i(y)$.

7. If $\text{supp}(f)$ is a singleton then set α^* to be the only element and update $\alpha = \alpha + \alpha^*$, $c = cf/x^{\alpha^*}$; and return $c x^\alpha P_0 \prod_{i=1}^m P_i(x^{\lambda_i})$.
8. Set $\Lambda_1 = \{((1), f(y, x_2, \dots, x_n))\}$.
For $k = 1, \dots, n - 1$ do
 - 8.1 Set $\Lambda_{k+1} = \{\}$.
 - 8.2 For $((\mu_1, \dots, \mu_k), h(y, x_{k+1}, \dots, x_n))$ in Λ_k do
Call the algorithm **BivariateQILD** with input $h \in \mathbb{R}[x_{k+2}, \dots, x_n][y, x_{k+1}]$ to compute its q -integer linear decomposition

$$h = y^{\alpha^*} x_{k+1}^{\beta^*} P_0^* \prod_{i=1}^{m^*} P_i^*(y^{\lambda_i^*} x_{k+1}^{\mu_i^*}, x_{k+2}, \dots, x_n), \quad (5.2)$$

where $P_0^* \in \mathbb{R}[y, x_{k+1}, \dots, x_n]$ and $P_i^*(y, x_{k+2}, \dots, x_n) \in \mathbb{R}[y, x_{k+2}, \dots, x_n]$; then update α by adding the vector $(\mu_1 \alpha^*, \dots, \mu_k \alpha^*, \beta^*, 0, \dots, 0)$, update P_0 by multiplying $P_0^*(x_1^{\mu_1} \cdots x_k^{\mu_k}, x_{k+1}, \dots, x_n)$ and update Λ_{k+1} by joining the elements $((\mu_1 \lambda_i^*, \dots, \mu_k \lambda_i^*, \mu_i^*), P_i^*(y, x_{k+2}, \dots, x_n))$ for $i = 1, \dots, m$.

9. Set $g \in \mathbb{R}[x_1, \dots, x_n]$ to be the denominator of P_0 . Update P_0 to be its numerator, $\alpha_i = \alpha_i - \deg_{x_i}(g)$ for $i = 1, \dots, n-1$, and for $(\mu, h(y))$ in Λ_n iteratively update $m = m + 1$, $\lambda_m = \mu$ and $P_m(y) = h(y)$.
10. Return $c \mathbf{x}^\alpha P_0 \prod_{i=1}^m P_i(x^{\lambda_i})$.

Theorem 5.4. *Let $p \in \mathbb{R}[x_1, \dots, x_n]$. Then the algorithm **MultivariateQILD**₂ correctly computes the q -integer linear decomposition of p .*

Proof. The correctness immediately follows from Proposition 5.3. \square

Theorem 5.5. *Let $p \in \mathbb{Z}[q, q^{-1}][x_1, \dots, x_n]$. Assume that the numerator and denominator of p have maximum degree d in each variable from $\{q, x_1, \dots, x_n\}$ separately and let $\|p\|_\infty = \beta$. Then the algorithm **MultivariateQILD**₂ computes the q -integer linear decomposition of p over \mathbb{Z} using $\tilde{O}(d^{n+6} + d^{n+4} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+5} + d^{n+4} \log \beta)$ with fast arithmetic.*

Proof. Let $T(n, d, \log \beta)$ denote the number of word operations used by the algorithm applied to p . The first three steps are exactly the same as the first approach introduced in the preceding section. Thus we know that Step 1 takes no word operations, Step 2 uses $\tilde{O}(d^{n+3} + d^{n+1} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+2} + d^{n+1} \log \beta)$ with fast arithmetic, and Step 3 gives that the initial cost $T(1, d, \log \beta)$ is in $\tilde{O}(d^4 + d^2 \log^2 \beta)$ with classical arithmetic and $\tilde{O}(d^3 + d^2 \log \beta)$ with fast arithmetic. Step 4 deals with the bivariate case. By Theorem 3.11, this step yields that $T(2, d, \log \beta)$ is in $\tilde{O}(d^8 + d^6 \log^2 \beta)$ with classical arithmetic and $\tilde{O}(d^7 + d^6 \log \beta)$ with fast arithmetic.

In Step 5, by Lemma 3.10, the computation of the content and primitive part can be done within $\tilde{O}(d^{n+3} + d^{n+1} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+2} + d^{n+1} \log \beta)$ with fast arithmetic. Notice that $g \in \mathbb{Z}[q, x_3, \dots, x_n]$ has maximum degree at most d in each variable separately and max-norm of word length $\tilde{O}(nd + \log \beta)$ by Lemma 3.9. Then Step 6 takes $\tilde{O}(T(n-2, d, nd + \log \beta))$ word operations. Step 7 takes linear time in the cardinality of $\text{supp}(f)$, which is at most $(d+1)^n$. In Step 8, notice that for the k th iteration, the polynomial $h \in \mathbb{Z}[q, x_{k+2}, \dots, x_n][y, x_{k+1}]$ has maximum degree at most d in each variable separately and max-norm of word length $\tilde{O}(nd + \log \beta)$. Thus by Theorem 3.11 with $v = n - k - 1$, the k th iteration requires $\tilde{O}(d^{n-k+7} + d^{n-k+5} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n-k+6} + d^{n-k+5} \log \beta)$ with fast arithmetic. Since $1 \leq k \leq n-1$, this step in total takes $\tilde{O}(d^{n+6} + d^{n+4} \log^2 \beta)$ word operations with classical arithmetic and $\tilde{O}(d^{n+5} + d^{n+4} \log \beta)$ with fast arithmetic, dominating the costs of Steps 9 and 10.

In summary, we obtain the recurrence relation

$$O(T(n, d, \log \beta)) \subset \tilde{O}(d^{n+6} + d^{n+4} \log^2 \beta) + O(T(n-2, d, nd + \log \beta)),$$

along with $T(1, d, \log \beta) \in \tilde{O}(d^4 + d^2 \log^2 \beta)$ and $T(2, d, \log \beta) \in \tilde{O}(d^8 + d^6 \log^2 \beta)$ with classical arithmetic or

$$O(T(n, d, \log \beta)) \subset \tilde{O}(d^{n+5} + d^{n+4} \log \beta) + O(T(n-2, d, nd + \log \beta)),$$

along with $T(1, d, \log \beta) \in \tilde{O}(d^3 + d^2 \log \beta)$ and $T(2, d, \log \beta) \in \tilde{O}(d^7 + d^6 \log \beta)$ with fast arithmetic. The announced cost follows. \square

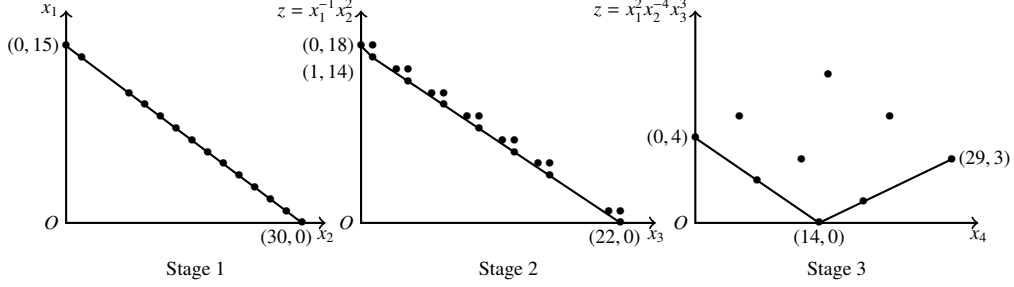


Figure 3: Newton polygons constructed in the three stages in Example 5.6.

Example 5.6. Consider the same polynomial p given by (4.1) as Example 4.8. In order to compute its q -integer linear decomposition over $\mathbb{Z}[q, q^{-1}]$, the algorithm **MultivariateQILD**₂ (mainly Step 8) proceeds in the following three stages with their respective Newton polygons plotted in Figure 3. Firstly, by viewing p as a polynomial in x_1, x_2 over $\mathbb{Z}[q, q^{-1}, x_3, x_4]$, applying the algorithm **BivariateQILD** to p gives

$$p = x_1^{15} P^{(1)}(x_1^{-1} x_2^2, x_3, x_4) \quad (5.3)$$

with

$$\begin{aligned} P^{(1)}(y, x_3, x_4) = & 7qy^{15}x_3x_4^{14} + 7qy^{15}x_4^{15} + 7q^2y^{14}x_3x_4^{14} + 63qy^{13}x_3^4x_4^{19} + 63qy^{13}x_3^3x_4^{20} \\ & + 63q^2y^{12}x_3^4x_4^{19} - 2y^{11}x_3^7x_4^7 - 2y^{11}x_3^6x_4^8 - 2qy^{10}x_3^7x_4^7 + 21q^3y^9x_3^{10}x_4^{29} - 18y^9x_3^{10}x_4^{12} \\ & + 21q^3y^9x_3^9x_4^{30} - 18y^9x_3^9x_4^{13} + 21q^4y^8x_3^{10}x_4^{29} - 18qy^8x_3^{10}x_4^{12} + 2qy^7x_3^{13} + 2qy^7x_3^{12}x_4 \\ & + 2q^2y^6x_3^{13} - 6q^2y^5x_3^{16}x_4^{22} + 18qy^5x_3^{16}x_4^5 - 6q^2y^5x_3^{15}x_4^{23} + 18qy^5x_3^{15}x_4^6 - 6q^3y^4x_3^{16}x_4^{22} \\ & + 18q^2y^4x_3^{16}x_4^5 + 6q^3yx_3^{22}x_4^{15} + 6q^3yx_3^{21}x_4^{16} + 6q^4x_3^{22}x_4^{15}. \end{aligned}$$

There is only one q -integer linear type, namely $(-1, 2)$, of p over $\mathbb{Z}[q, q^{-1}, x_3, x_4]$. Next, with input $P^{(1)}(y, x_3, x_4) \in \mathbb{Z}[q, q^{-1}, x_4][y, x_3]$, calling the algorithm **BivariateQILD** again and substituting $y = x_1^{-1}x_2^2$ yields

$$p = x_2^{28} \cdot P_0 \cdot P^{(2)}(x_1^2x_2^{-4}x_3^3, x_4), \quad (5.4)$$

where $P_0 = qx_1x_3 + x_2^2x_3 + x_2^2x_4$ and $P^{(2)}(y, x_4) = 6q^3y^7x_4^{15} - 6q^2y^5x_4^{22} + 18qy^5x_4^5 + 2qy^4 + 21q^3y^3x_4^{29} - 18y^3x_4^{12} - 2y^2x_4^7 + 63qyx_4^{19} + 7qx_4^{14}$. The vector $(2, -4, 3)$ is then the only q -integer linear type of p over $\mathbb{Z}[q, q^{-1}, x_4]$. Finally, the last call to the algorithm **BivariateQILD** with input $P^{(2)}(y, x_4) \in \mathbb{Z}[q, q^{-1}][y, x_4]$, along with the substitution $y = x_1^2x_2^{-4}x_3^3$, leads to the desired decomposition (4.2). The two q -integer linear types $(2, -4, 3, 5)$ and $(-4, -8, -6, 7)$ of p over $\mathbb{Z}[q, q^{-1}]$ have been correctly recovered.

From (5.3) and (5.4), one sees that p is q -integer linear over $\mathbb{Z}[q, q^{-1}, x_3, x_4]$ but it is not q -integer linear over $\mathbb{Z}[q, q^{-1}, x_4]$. The latter in turn indicates the non- q -integer linearity of p over $\mathbb{Z}[q, q^{-1}]$, even before starting the third stage.

Once more, similar to the bivariate algorithm, the above algorithm can be modified so as to determine the q -integer linearity of a given polynomial only. In other words, the algorithm can halt already and return the negative answer whenever one of the following situations occurs.

- In Step 4 or in any iteration step of Step 8.2, any of the triggers of the bivariate algorithm listed in Remark 3.8 is touched.
- In Step 6, the polynomial g turns out to be not q -integer linear.

6. Complexity comparison

In this section, we discuss two more algorithms for computing the q -integer linear decomposition of polynomials, along with their complexity analyses in the case of bivariate polynomials over $\mathbb{Z}[q, q^{-1}]$, so as to compare with our algorithms presented in Sections 4 and 5.

The first algorithm is based on resultants and was developed by Le (2001), which itself serves as a q -analogue of the algorithm of Abramov and Le (2002) in the ordinary shift case. As already mentioned in the introduction, this algorithm is completely focused on bivariate polynomials. So we will further extend it to also tackle polynomials having more than two variables. The second algorithm is based on full irreducible factorization of polynomials and work for polynomials in any number of variables. This algorithm can be viewed as a q -analogue of the algorithm of Li and Zhang (2013) from the ordinary shift case. In order to give complexity comparison, we need to analyze the costs of these two algorithms. As such we will briefly describe their main ideas.

6.1. Resultant-based algorithm

As we proceed with our bivariate algorithm, the algorithm of Le (2001) first finds candidates for q -integer linear types of a given bivariate polynomial and then obtains the corresponding univariate polynomials by going through these candidates. The difference is that they use resultants to determine candidates and perform bivariate GCD computations iteratively for detecting each candidate.

In order to state its main idea, let $p \in \mathbb{R}[x, y]$ be a nonconstant polynomial which is primitive with respect to its either variable. Then we know that p admits the q -integer linear decomposition of the form (3.1), in which all the λ_i, μ_i are nonzero. By Lemma 5.2, an integer pair (λ, μ) with $\lambda\mu \neq 0$ is one of the q -integer linear types (λ_i, μ_i) if and only if there exists a nonconstant factor $f \in \mathbb{R}[x, y]$ of p with the property that f divides $f(q^\mu x, q^{-\lambda} y)$ in $\mathbb{R}[x, y]$. Note that such an f must satisfy $\deg_x(f) \deg_y(f) > 0$ and $f(x, 0)f(0, y) \neq 0$ because p is assumed to be primitive with respect to its either variable. By a careful study on the structure of the factor f , it is then not hard to see that f divides $f(q^\mu x, q^{-\lambda} y)$ in $\mathbb{R}[x, y]$ if and only if f divides $f(qx, q^{-\lambda/\mu} y)$ in $\mathbb{R}[x, y]$. Observe that any integer pair (λ, μ) with $\lambda\mu \neq 0$ is uniquely determined by the rational $r = -\lambda/\mu$. We have thus shown the following.

Lemma 6.1. *With p given above, a nonzero rational number r gives rise to a q -integer linear type of p if and only if $\gcd(p, p(qx, q^r y))$ is nonconstant.*

This implies that all rationals $r_i = -\lambda_i/\mu_i$ must be roots of the resultant $\text{Res}_y(p, p(qx, q^r y)) \in \mathbb{R}[q^r, x]$ in terms of r , or equivalently, they are eliminated by the content in $\mathbb{R}[q^r]$ of the resultant with respect to x . Note that such rational roots of a polynomial in $\mathbb{R}[q^r]$ can be found by matching powers of q appearing in the given polynomial in pairs along with a subsequent substitution for zero testing. One can find more details in (Le, 2001, §5) or (Le et al., 2001, §6). Accordingly, we derive a way to produce candidates for the r_i (and then for the (λ_i, μ_i)). After generating candidates, the algorithm of Le (2001) continues to find the corresponding univariate polynomials by calculating a factor f of p that stabilizes $\gcd(f, f(qx, q^r y))$, or more efficiently,

$\gcd(f, f(q^\mu x, q^{-\lambda} y))$ for each candidate $r = -\lambda/\mu$. This operation actually induces bivariate polynomial arithmetic over \mathbb{R} and thus may take considerably more time than Step 6.1 of our bivariate algorithm **BivariateQILD**. In order to improve the performance, we instead proceed by using Step 6 of our bivariate algorithm.

Note that Lemma 6.1 cannot be literally carried over to a polynomial in more than two variables. It is not clear how to directly generalize the algorithm of Le (2001) to the multivariate case. Nevertheless, as indicated by Proposition 5.3, this algorithm extends to the case of polynomials in any number of variables in the same fashion as our bivariate algorithm.

The lemma below provides bounds for the resultant of two trivariate integer polynomials, which can be verified by following the proof of (Bistritz and Lifshitz, 2010, Theorem 10) but arguing from the perspective of trivariate polynomials.

Lemma 6.2. *Let $f, g \in \mathbb{Z}[q, x, y]$ with $\deg_q(f) = d_q$, $\deg_x(f) = d_x$, $\deg_y(f) = d_y$, $\deg_q(g) = e_q$, $\deg_x(g) = e_x$ and $\deg_y(g) = e_y$. Let $R = \text{Res}_y(f, g)$. Then $R \in \mathbb{Z}[q, x]$ with $\deg_q(R) \leq d_q e_y + d_y e_q$, $\deg_x(R) \leq d_x e_y + d_y e_x$ and*

$$\|R\|_\infty \leq (d_y + e_y)!(\max\{d_q, e_q\} + 1)^{d_y + e_y - 1} (\max\{d_x, e_x\} + 1)^{d_y + e_y - 1} \|f\|_\infty^{e_y} \|g\|_\infty^{d_y}.$$

The following theorem gives a complexity analysis for the algorithm of Le (2001) when applied to a polynomial in $\mathbb{Z}[q, q^{-1}][x, y]$.

Theorem 6.3. *Let p be a polynomial in $\mathbb{Z}[q, q^{-1}][x, y]$ whose numerator and denominator have maximum degree d in each variable from $\{q, x, y\}$ and with $\|p\|_\infty = \beta$. Then the algorithm of Le takes $O^{\sim}(d^{11} + d^{10} \log \beta + d^8 \log^2 \beta)$ word operations with classical arithmetic and $O^{\sim}(d^9 + d^8 \log \beta)$ with fast arithmetic.*

Proof. With a slight abuse of notation, let p be the input polynomial with content with respect to its either variable being removed. Then $p \in \mathbb{Z}[q, x, y]$ and $\log \|p\|_\infty \in O(d + \log \beta)$. The algorithm proceeds to compute the resultant $\text{Res}_y(p, p(qx, q^r y))$ with r undetermined. Observe that every entry in the Sylvester matrix is a monomial in q^r . Thus we have $\|\text{Res}_y(p, p(qx, q^r y))\|_\infty \leq \|\text{Res}_y(p, p(qx, y))\|_\infty$. By Lemma 6.2, $\text{Res}_y(p, p(qx, q^r y))$ has degree in q at most $3d^2$, degree in q^r at most d^2 , degree in x at most $2d^2$ and max-norm at most $B = (2d)!(2d+1)^{2d-1}(d+1)^{2d-1}\|p\|_\infty^{2d}$. Then $\log B \in O(d^2 + d \log \beta)$. Viewing q^r as a new indeterminate u independent of q , we can compute this resultant using a small prime modular algorithm, along with an evaluation-interpolation scheme: (1) choose $\lceil \log_2(2B+1) \rceil$ primes, each of size $O(\log B)$; (2) for every chosen prime h , do the following: reduce all coefficients of $P_0(x, y)$ and $P_0(qx, uy)$ modulo h , evaluate both modular images at $D = 6d^6$ points for (q, u, x) , compute D univariate resultants of two polynomials in $\mathbb{Z}_h[y]$ of degrees in y at most d , and recover the modular resultant by interpolation; (3) reconstruct the desired resultant using the Chinese remainder theorem. Ignoring the cost for choosing primes in Step (1), we analyze the costs used by Steps (2)-(3). In Step (2), the cost per prime h for reducing all coefficients modulo h is $O(d^2 \log \beta \log h)$ word operations with classical arithmetic. The evaluation and interpolation steps are performed in $O(d^3 D) \subset O(d^9)$ arithmetic operations in \mathbb{Z}_h with classical arithmetic and $O^{\sim}(dD) \subset O^{\sim}(d^7)$ with fast arithmetic. Each univariate resultant over $\mathbb{Z}_h[y]$ can be computed in $O(d^2)$ arithmetic operations in \mathbb{Z}_h with classical arithmetic and $O^{\sim}(d)$ with fast arithmetic, yielding $O(d^8)$ arithmetic operations in \mathbb{Z}_h with classical arithmetic and $O^{\sim}(d^7)$ with fast arithmetic in total for this step. Notice that the cost for each arithmetic operations in \mathbb{Z}_h is $O(\log^2 h)$ word operations with classical arithmetic and $O(\log h)$ with fast arithmetic. Also notice that every chosen prime h is of word length $\log h \in O(\log \log B)$. Thus

Step (2) in total takes $O^\sim(d^9 \log B)$ word operations with classical arithmetic and $O^\sim(d^7 \log B)$ with fast arithmetic. In Step (3), the Chinese remainder theorem in total requires $O^\sim(d^6 \log^2 B)$ word operations with classical arithmetic and $O^\sim(d^6 \log B)$ with fast arithmetic. Therefore, computing the resultant $\text{Res}_y(p, p(qx, q^r y))$ takes $O^\sim(d^{11} + d^{10} \log \beta + d^8 \log^2 \beta)$ word operations with classical arithmetic and $O^\sim(d^9 + d^8 \log \beta)$ with fast arithmetic. This will dominate the costs for subsequent steps including finding the rational roots and computing corresponding univariate polynomials. The claimed cost follows. \square

6.2. Factorization-based algorithm

Similar to the ordinary shift case (Li and Zhang, 2013), the q -integer linear decomposition of a multivariate polynomial can also be computed by full irreducible factorization. The key observation is that, for any q -integer linear polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$ of only one type $(\lambda_1, \dots, \lambda_n)$, the difference of any two vectors from $\text{supp}(p)$ can be written into the form $k \cdot (\lambda_1, \dots, \lambda_n)$ for some $k \in \mathbb{Z}$. This allows one to readily determine the q -integer linearity of any irreducible polynomial. That is, given an irreducible polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$, take $\alpha \in \mathbb{N}^n$ to be the minimal vector of $\text{supp}(p)$ and investigate whether the difference between α and any other vector from $\text{supp}(p)$ is equal to a scalar multiple of the same integer vector. One thus immediately establishes a factorization-based algorithm for the computation of the q -integer linear decomposition of a polynomial in $\mathbb{R}[x_1, \dots, x_n]$: first perform the full irreducible factorization of the input polynomial over \mathbb{R} ; then determine the q -integer linearity of each irreducible factor; finally regroup all factors of the same q -integer linear type.

A careful study of the above algorithm leads to the following complexity.

Theorem 6.4. *Let p be a polynomial in $\mathbb{Z}[q, q^{-1}][x, y]$ whose numerator and denominator have maximum degree d in each variable from $\{q, x, y\}$ and with $\|p\|_\infty = \beta$. Then the factorization-based algorithm described above requires $O^\sim(d^9 \log^2 \beta)$ word operations with classical arithmetic and $O^\sim(d^8 \log \beta)$ with fast arithmetic.*

Proof. Computing a complete factorization of p into irreducibles over $\mathbb{Z}[q, q^{-1}]$ dominates the other costs of the algorithm. This is essentially the complexity of factoring in $\mathbb{Z}[q][x, y]$, for polynomials bounded by degree d in all variables (q, x and y). While we do not know of an explicit analysis of this complexity (beyond being in polynomial-time, since (Kaltofen, 1985)), the algorithm of Gao (2003) can be applied and analyzed over the function field $\mathbb{Q}(q)$, and appears to require $O^\sim(d^9 \log^2 \beta)$ word operations using classical arithmetic, and $O^\sim(d^8 \log \beta)$ word operations using fast arithmetic. \square

7. Implementation and timings

We have implemented all algorithms in MAPLE 2018 in the case where the domain \mathbb{R} is the ring of polynomials over $\mathbb{Z}[q, q^{-1}]$. The code is available by email request. In order to get an idea about the efficiency of our algorithms, we have compared their runtime, as well as the memory requirements, to the performance of our Maple implementations of the two algorithms discussed in the preceding section.

The test suite was generated by

$$p = P_0 \prod_{i=1}^m \text{num}(P_i(\mathbf{x}^{\lambda_i})), \quad (7.1)$$

where $n, m \in \mathbb{N}$,

- $P_0 \in \mathbb{Z}[q][x_1, \dots, x_n]$ is a random polynomial with $\deg_{x_1, \dots, x_n}(P_0) = \deg_q(P_0) = d_0$,
- the $\lambda_i \in \mathbb{Z}^n$ are random integer vectors each of which has entries of maximum absolute value no more than 10 (note that they may not be distinct),
- $P_i(z) = f_{i1}(z)f_{i2}(z)$ with $f_{ij}(z) \in \mathbb{Z}[q][z]$ a random polynomial of degree $j \cdot d$ for some $d \in \mathbb{N}$, and $\text{num}(\dots)$ denotes the numerator of the argument.

Note that, in all tests, the algorithms take the expanded forms of examples given above as input. All timings are measured in seconds on a Linux computer with 128GB RAM and fifteen 1.2GHz Dual core processors. The computations for the experiments did not use any parallelism.

For a selection of random polynomials of the form (7.1) for different choices of n, m, d_0, d , Table 1 collects the timings of the algorithm of Le (LQILD), the algorithm based on factorization (FQILD) and our two algorithms (MQILD₁, MQILD₂). The dash in the table indicates that with this choice of (m, n, d_0, d) , the corresponding procedure reached the CPU time limit (which was set to 12 hours) and yet did not return.

(n, m, d_0, d)	LQILD	FQILD	MQILD ₁	MQILD ₂
(2, 1, 1, 1)	5408.48	0.04	0.01	0.01
(2, 1, 5, 1)	8381.99	0.06	0.03	0.03
(2, 1, 10, 1)	—	0.19	0.04	0.04
(2, 1, 20, 1)	—	0.63	0.09	0.09
(2, 1, 30, 1)	—	1.47	0.13	0.10
(2, 1, 40, 1)	—	2.55	0.24	0.21
(2, 1, 50, 1)	—	6.64	0.42	0.39
(2, 2, 10, 1)	—	0.92	0.10	0.08
(2, 3, 10, 1)	—	3.29	0.31	0.26
(2, 4, 10, 1)	—	5.74	0.67	0.54
(2, 5, 10, 1)	—	18.83	2.01	1.54
(2, 2, 10, 2)	—	4.55	0.27	0.20
(2, 4, 10, 2)	—	114.82	4.98	4.53
(2, 5, 10, 2)	—	264.02	25.63	24.29
(2, 3, 10, 2)	—	36.14	1.38	1.21
(2, 3, 10, 3)	—	169.13	4.28	3.80
(2, 3, 10, 4)	—	649.03	12.15	12.86
(2, 3, 10, 5)	—	1554.31	31.54	33.50
(2, 2, 5, 1)	—	0.32	0.05	0.05
(3, 2, 5, 1)	—	1.99	0.14	0.12
(4, 2, 5, 1)	—	11.46	0.35	0.20
(5, 2, 5, 1)	—	183.17	0.99	0.63
(6, 2, 5, 1)	—	1141.32	2.58	0.98
(7, 2, 5, 1)	—	11759.89	6.07	1.74
(8, 2, 5, 1)	—	18153.45	10.60	5.29
(9, 2, 5, 1)	—	—	65.53	38.12
(10, 2, 5, 1)	—	—	176.25	89.87

Table 1: Comparison of all four algorithms for a collection of polynomials p of the form (7.1).

8. Conclusion

In this paper we have presented two new algorithms for computing the q -integer linear decomposition of a multivariate polynomial over any UFD of characteristic zero. When restricted to the bivariate case, both algorithms reduce to the same bivariate algorithm. For the sake of comparison, we included an algorithm based on full irreducible factorization of polynomials. Compared with the known algorithm of Le (2001) and this factorization-based algorithm in the bivariate case, our algorithm is considerably faster. In practice, both our algorithms are also more efficient than these two algorithms. In addition, we have extended and improved the original contribution of Le and provided complexity analysis for the improved version. We remark that both our algorithms have much better performances than the other two algorithms in the case where the coefficient domain contains algebraic numbers.

References

- Abramov, S. A., Le, H. Q., 2002. A criterion for the applicability of Zeilberger's algorithm to rational functions. *Discrete Math.* 259 (1-3), 1–17.
URL [http://dx.doi.org/10.1016/S0012-365X\(02\)00442-9](http://dx.doi.org/10.1016/S0012-365X(02)00442-9)
- Abramov, S. A., Petkovšek, M., 2001. Proof of a conjecture of Wilf and Zeilberger. Preprints Series of the Institute of Mathematics, Physics and Mechanics 39(748)(2001), Ljubljana, March 9, 2001.
- Abramov, S. A., Petkovšek, M., 2002. On the structure of multivariate hypergeometric terms. *Adv. in Appl. Math.* 29 (3), 386–411.
URL [http://dx.doi.org/10.1016/S0196-8858\(02\)00022-2](http://dx.doi.org/10.1016/S0196-8858(02)00022-2)
- Anderson, K. R., 1978. A reevaluation of an efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.* 7 (1), 53–55.
URL [https://doi.org/10.1016/0020-0190\(78\)90041-8](https://doi.org/10.1016/0020-0190(78)90041-8)
- Bistritz, Y., Lifshitz, A., 2010. Bounds for resultants of univariate and bivariate polynomials. *Linear Algebra Appl.* 432 (8), 1995–2005.
URL <https://doi.org/10.1016/j.laa.2009.08.012>
- Chen, S., Koutschan, C., 2019. Proof of the Wilf-Zeilberger conjecture for mixed hypergeometric terms. *J. Symbolic Comput.* 93, 133–147.
URL <http://www.sciencedirect.com/science/article/pii/S0747717118300671>
- Chen, W. Y. C., Hou, Q.-H., Mu, Y.-P., 2005. Applicability of the q -analogue of Zeilberger's algorithm. *J. Symbolic Comput.* 39 (2), 155–170.
URL <http://dx.doi.org/10.1016/j.jsc.2004.09.002>
- Conflitti, A., 2003. On computation of the greatest common divisor of several polynomials over a finite field. *Finite Fields Appl.* 9 (4), 423–431.
URL [https://doi.org/10.1016/S1071-5797\(03\)00022-4](https://doi.org/10.1016/S1071-5797(03)00022-4)
- Du, H., Li, Z., 2019. The Ore-Sato theorem and shift exponents in the q -difference case. *J. Syst. Sci. Complex.* 32 (1), 271–286.
URL <https://doi.org/10.1007/s11424-019-8355-1>
- Gao, S., 2003. Factoring multivariate polynomials via partial differential equations. *Math. Comp.* 72 (242), 801–822.
URL <https://doi.org/10.1090/S0025-5718-02-01428-X>
- von zur Gathen, J., Gerhard, J., 2013. *Modern Computer Algebra*, 3rd Edition. Cambridge University Press, Cambridge.
URL <http://dx.doi.org/10.1017/CBO9781139856065>
- Gel'fond, A. O., 1960. *Transcendental and algebraic numbers*. Translated from the first Russian edition by Leo F. Boron. Dover Publications, Inc., New York.
- Giesbrecht, M., Huang, H., Labahn, G., Zima, E., 2019a. Efficient integer-linear decomposition of multivariate polynomials. In: *Proceedings of ISSAC'19*. ACM, New York, pp. 171–178.
URL <http://doi.acm.org/10.1145/3326229.3326261>
- Giesbrecht, M., Huang, H., Labahn, G., Zima, E., 2019b. Efficient rational creative telescoping. Preprint: arXiv:1909.06898.
- Graham, R. L., 1972. An efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.* 1, 132–133.
URL [https://doi.org/10.1016/0020-0190\(72\)90045-2](https://doi.org/10.1016/0020-0190(72)90045-2)

- Hou, Q.-H., 2004. k -free recurrences of double hypergeometric terms. *Adv. in Appl. Math.* 32 (3), 468–484.
 URL [http://dx.doi.org/10.1016/S0196-8858\(03\)00056-3](http://dx.doi.org/10.1016/S0196-8858(03)00056-3)
- Kaltofen, E., 1985. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.* 14 (2), 469–489.
 URL <https://doi.org/10.1137/0214035>
- Le, H. Q., 2001. On the q -analogue of Zeilberger’s algorithm to rational functions. *Program. Comput. Softw.* 27 (1), 35–42.
 URL <http://dx.doi.org/10.1023/A:1007186703355>
- Le, H. Q., Abramov, S. A., Geddes, K. O., 2001. A direct algorithm to construct the minimal telescopers for rational functions (q -difference case). Technical Report CS-2001-25, Department of Computer Science, University of Waterloo, ON, Canada.
- Li, Z., Zhang, Y., 2013. An algorithm for decomposing multivariate hypergeometric terms. A contributed talk in CM’13.
- Paule, P., Riese, A., 1997. A Mathematica q -analogue of Zeilberger’s algorithm based on an algebraically motivated approach to q -hypergeometric telescoping. In: *Special functions, q -series and related topics* (Toronto, ON, 1995). Vol. 14 of *Fields Inst. Commun. Amer. Math. Soc.*, Providence, RI, pp. 179–210.
- Walker, R. J., 1950. *Algebraic Curves*. Princeton Mathematical Series, vol. 13. Princeton University Press, Princeton, N. J.
- Wilf, H. S., Zeilberger, D., 1992. An algorithmic proof theory for hypergeometric (ordinary and “ q ”) multisum/integral identities. *Invent. Math.* 108 (3), 575–633.
 URL <http://dx.doi.org/10.1007/BF02100618>