Factoring Differential Operators in $n$ Variables

Mark Giesbrecht
David R. Cheriton School of Computer Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
mwg@uwaterloo.ca

Albert Heinle
David R. Cheriton School of Computer Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
aheinle@uwaterloo.ca

Viktor Levandovskyy
Lehrstuhl D für Mathematik,
RWTH Aachen University
Templergraben 64
Aachen, Germany
viktor.levandovskyy@math.rwth-aachen.de

ABSTRACT
In this paper, we present a new algorithm and an experimental implementation for factoring elements in the polynomial $n$th Weyl algebra, the polynomial $n$th shift algebra, and $\mathbb{Z}^n$-graded polynomials in the $n$th $q$-Weyl algebra.

The most unexpected result is that this noncommutative problem of factoring partial differential operators can be approached effectively by reducing it to the problem of solving systems of polynomial equations over a commutative ring. In the case where a given polynomial is $\mathbb{Z}^n$-graded, we can reduce the problem completely to factoring an element in a commutative multivariate polynomial ring.

The implementation in Singular is effective on a broad range of polynomials and increases the ability of computer algebra systems to address this important problem. We compare the performance and output of our algorithm with other implementations in commodity computer algebra systems on nontrivial examples.

Categories and Subject Descriptors
G.4 [Mathematical Software]: Algorithm design and analysis; I.1.2 [Symbolic and Algebraic Manipulation]: Algorithms—Factorization

General Terms
Algorithms, Design, Theory

Keywords
Factorization, linear partial differential operator, non-commutative algebra, Singular, algebra of operators, Weyl algebra

1. INTRODUCTION
In this paper we present a new method and an implementation for factoring elements in the $n$th polynomial Weyl algebra $A_n$, and the $n$th polynomial shift algebra. An adaptation of these ideas can also be applied to classes of polynomials in the $n$th $q$-Weyl algebra, which is also outlined here.

There are numerous important applications for this method, notably since one can view those rings as operator algebras. For example, given an element $L \in A_n$ and viewing $L$ as a differential operator, one can derive properties of its solution spaces. Especially concerning the problem of finding the solution to the differential equation associated with $L$, the preconditioning step of factoring $L$ can help to reduce the complexity of that problem in advance.

The new technique heavily uses the nontrivial $\mathbb{Z}^n$-grading on $A_n$ and, to the best of our knowledge, has no analogues in the literature on factorizations for $n \geq 2$. However, for $n = 1$ it is the same grading that lies behind the Kashihara-Malgrange V-filtration ([16] and [21]), which is of great importance in the $D$-module theory. Van Hoeij also made use of this technique in [30] to factorize elements in the first Weyl algebra with power series coefficients. Notably, for $n \geq 2$, the $\mathbb{Z}^n$-grading we propose is very different from the mentioned $\mathbb{Z}$-grading. Among others, a recent result from [4] states that the Gel’fand-Kirillov dimension [11] of the 0th graded part of $\mathbb{Z}$-grading of $A_2$ is in fact $2n - 1$. The Gel’fand-Kirillov dimension of the whole ring $A_n$ is, for comparison, $2n$. The 0th graded part of the $\mathbb{Z}^n$-grading we propose has Gel’fand-Kirillov dimension $n$.

Definition 1. Let $A$ be an algebra over a field $\mathbb{K}$ and $f \in A \setminus \mathbb{K}$ be a polynomial. A nontrivial factorization of $f$ is a tuple $(c, f_1, \ldots, f_m)$, where $c \in \mathbb{K} \setminus \{0\}$, $f_1, \ldots, f_m \in A \setminus \{1\}$ are monic polynomials and $f = c \cdot f_1 \cdots f_m$.

In general, we identify two problems in noncommutative factorization for a given polynomial $f$: (i) finding one factorization of $f$, and (ii) finding all possible factorizations of $f$. Item (ii) is interesting since factorizations in noncommutative rings are not unique in the classical sense (i.e., up to multiplication by a unit), and regarding the problem of solving the associated differential equation one factorization might be more useful than another. We show how to approach both problems here.

A number of papers and implementations have been published in the field of factorization in operator algebras over the past few decades. Most of them concentrated on linear differential operators with rational coefficients. Tsarev has studied the form, number and properties of the factors of a differential operator in [26] and [27], which extends the papers [19] and [20]. A very general approach to noncommutative algebras and their properties, including factorization, is also done in the book by Bueso et al. in [8]. The authors provide several algorithms and introduce various points of views when dealing with noncommutative polynomial algebras.

In his dissertation van Hoeij [28] developed an algorithm...
to factorize a univariate differential operator. Several papers following that dissertation extend these techniques \cite{29, 30, 31}, and this algorithm is implemented in the DEnTools package of MAPLE \cite{23} as the standard algorithm for factorization of these operators.

In the REDUCE-base computer algebra system AL-TYPES, Schwarz and Grigoriev \cite{25} have implemented the algorithm for factoring differential operators they introduced in \cite{12}. As far as we know, this implementation is solely accessible as a web service. Beals and Kartashova \cite{6} consider the problem of factoring polynomials in the second Weyl algebra, where they are able to deduce parametric factors.

For special classes of polynomials in operator algebras, Foupouagnigni et al. \cite{10} show some unexpected results about factorizations of fourth-order differential equations satisfied by certain Laguerre-Hahn orthogonal polynomials.

From a more algebraic point of view, and dealing only with strictly polynomial noncommutative algebras, Melenk and Apel \cite{22} developed a package for the computer algebra system REDUCE. That package provides tools to deal with noncommutative polynomial algebras and also contains a factorization algorithm for the supported algebras. It is, moreover, the only tool besides our implementation in SINGULAR \cite{9} that is capable of factoring in operator algebras with more than one variable. Unfortunately, there are no further publications about how the implementation works besides the available code.

The above mentioned algorithms and implementations are very well written and they are able to factorize a large number of polynomials. Nonetheless, as pointed out in \cite{13, 14}, there exists a large class of polynomials, even in the first Weyl algebra, that seem to form the worst case for those algorithms. This class is namely the graded (or homogeneous) polynomials in the sense of the \( \mathbb{Z} \)-graded structure on the nth Weyl algebra. Using our techniques, we are able to obtain a factorization very quickly utilizing commutative factorization and some combinatorics. Those techniques are discussed for the first \( \mathbb{Z} \)-Weyl algebra in detail in \cite{14}.

Factorization of a general non-graded polynomial is much more involved. The main idea lies in inspecting the highest and the lowest graded summands of the polynomial to factorize. Any factorization corresponds respectively to the highest or the lowest graded summands of the factors. Since the graded factorization appears to be easy, we are able to factorize those summands and obtain finitely many candidates for highest and lowest summands of the factors. Obtaining the rest of the graded summands is the subject considered in this paper.

An implementation dealing with the first Weyl algebra, the first shift algebra, and graded polynomials in the first \( \mathbb{Z} \)-Weyl algebra, was created by Heinle and Levandovskyy within the computer algebra system SINGULAR \cite{9}. For the latter algebra, the implementation in SINGULAR is the only one available that deals with \( \mathbb{Z} \)-Weyl algebras, to the knowledge of the authors. The code has been distributed since version 3-1-3 of SINGULAR inside the library ncfactor.lib, and received a major update in version 3-1-6.

The new approach described in this paper will soon be available in an upcoming release of SINGULAR. There are new functions for factoring polynomials in the nth polynomial Weyl algebra, homogeneous polynomials in the nth polynomial \( \mathbb{Z} \)-Weyl algebra and the nth polynomial shift algebra.

The remainder of this paper is organized as follows. The rest of this section is dedicated to providing basic notions, definitions and results that are needed to describe our approach. Most of the results are well known, but have not been used for factorization until now.

Section 2 contains a methodology to deal with the factorization problem for graded polynomials, while in Section 3 we utilize this methodology to factorize arbitrary polynomials in the nth Weyl algebra. In Section 4 we evaluate our experimental implementation on several examples in Section 4 and compare the results to other commodity computer algebra systems.

1.1 Basic Notions and Definitions

By \( K \) we always denote a field of characteristic zero (though some of the statements also hold for some finite fields). For notational convenience we write \( n \) for \( \{1, \ldots, n\} \) and \( \emptyset \) for \( \emptyset, \ldots, \emptyset \) for \( n \in \mathbb{N} \) throughout.

**Definition 2.** The nth \( \mathbb{Z} \)-Weyl algebra \( Q_n \) is defined as
\[
Q_n := K\left\{ x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \right\} 
\]
for \( (i, j) \in \mathbb{N} \times \mathbb{N} \):
\[
\partial_i x_j = \begin{cases} x_j \partial_i, & \text{if } i \neq j \\ q_i x_j \partial_i + 1, & \text{if } i = j \end{cases}
\]
where \( q_1, \ldots, q_n \) are units in \( K \). For the special case where \( q_1 = \cdots = q_n = 1 \) we have the nth Weyl algebra, which is denoted by \( A_n \). For notational convenience, we write \( \Delta^D_w := x_1^{e_1} \cdots x_n^{e_n} \partial_1^{w_1} \cdots \partial_n^{w_n} \) for every monomial, where \( e, w \in \mathbb{N}_0^n \).

**Definition 3.** The nth shift algebra \( S_n \) is defined as
\[
S_n := K\left\{ x_1, \ldots, x_n, s_1, \ldots, s_n \right\} 
\]
for \( (i, j) \in \mathbb{N} \times \mathbb{N} \):
\[
s_i x_j = \begin{cases} x_j s_i, & \text{if } i \neq j \\ (x_j + 1) s_i, & \text{if } i = j \end{cases}
\]
For notational convenience, we write as above \( \Delta^S_w := x_1^{e_1} \cdots x_n^{e_n} s_1^{w_1} \cdots s_n^{w_n} \), where \( e, w \in \mathbb{N}_0^n \).

**Remark 1.** Throughout this paper we view \( \mathbb{Z}^n \), equipped with the coordinate-wise addition, as an ordered monoid with respect to a total ordering \( \prec \), compatible with addition and satisfying the following property: for any \( z_1, z_2 \in \mathbb{Z}^n \), such that \( z_2 < z_1 \), the set \( \{ z \in \mathbb{Z}^n : z_2 < z < z_1 \} \) is finite.

The nth \( \mathbb{Z} \)-Weyl algebra possesses a nontrivial \( \mathbb{Z}^n \)-grading using the weight vector \( [w, w] \) for a \( \emptyset \neq w \in \mathbb{Z}^n \) on the elements \( x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \). For simplicity, we choose \( w := [1, \ldots, 1] \). In what follows, \( \deg \) denotes the degree induced by this weight vector, that is \( \deg(\Delta^D_w) := [b_1 - a_1, \ldots, b_n - a_n] \) for \( a, b \in \mathbb{N}_0^n \). Note, that a \( \mathbb{Z} \)-grading, arising from \( V \)-filtration \cite{16, 21} prescribes to \( \Delta^D_w \) the grade \( \sum_{i=1}^n (b_i - a_i) \in \mathbb{Z} \).

We call a polynomial homogeneous or graded, if every summand is weighted homogeneous with respect to the weight vector \([w, w]\) as above.

**Example 1.** In the second Weyl algebra \( A_2 \) one has
\[
\deg(x_1 x_2 \partial_1 \partial_2) = \deg((\partial_1 x_1 + 1)x_2 \partial_2) = [0, 0].
\]
The polynomial \( x_1 \partial_1^2 x_2 + x_2 \partial_2 x_1 + \partial_1 x_2 \) is homogeneous of degree \([1, -1]\). The monomials \( x_1 \partial_2, \text{ resp. } x_2 \partial_1 \), have degrees \([-1, 1]\), resp. \([1, -1]\), hence their sum is not homogeneous.
Note that the so-called Euler operators $\theta_i := x_i \partial_i, i \in \mathbb{N}$ have degree 0 for all $i$, and thus play an important role, as we shall soon see.

First, we study some commutation rules the Euler operator $\theta_i$ has with $x_i$ and $\partial_i$. For $Q_n$, in order abbreviate the size of our formulae, we introduce the so-called q-bracket.

**Definition 4.** For $n \in \mathbb{N}$ and $q \in K \setminus \{0\}$, the $q$-bracket of $n$ is defined to be $[n]_q := \frac{q^n - 1}{q - 1} = \sum_{i=0}^{n-1} q^i$.

**Lemma 1** (Compare with [24]). In $Q_n$, the following commutation rules hold for $m \in \mathbb{N}$ and $i \in \mathbb{N}$:

$$\theta_i x^m_i = x^m_i (\theta_i + m), \quad \theta_i \partial^m_i = \partial^m_i (\theta_i - m).$$

More generally, in $Q_n$, the following commutation rules hold for $m \in \mathbb{N}$ and $i \in \mathbb{N}$:

$$\theta_i x^m_i = x^m_i (\theta_i + [m]_q), \quad \theta_i \partial^m_i = \frac{\partial^m_i}{q_i} \left( \frac{\theta_i - 1}{q_i^{m+1}} - \frac{q_i^{m+2} - q_i}{1 - q_i} \right).$$

The commutation rules stated in Lemma 1 can, of course, be extended to arbitrary polynomials in the $\theta_i$, $i \in \mathbb{N}$.

**Corollary 1.** Consider $f(\theta) \in K[\theta]$. Then, in $Q_n$, we have $f(\theta) \sum k = \sum f(\theta_i + \theta_{i+1} + \cdots + \theta_n) + c_n$, and $f(\theta) \partial^m = \partial^m f(\theta_i - \theta_1 - \cdots - \theta_m)$. Analogous identities with the respective commutation rules as given in Lemma 1 hold for $Q_n$.

**2. FACTORING GRADED POLYNOMIALS**

For graded polynomials, the main idea of our factorization technique lies in the reduction to a commutative univariate polynomial subring of $A_n$, respectively $Q_n$, namely $K[\theta]$. Actually, it appears that this subring is quite large in the sense of reducibility of its elements in $A_n$ (resp. $Q_n$).

Due to the commutativity of $x_i$ with $\partial_i$, for $i \neq j$, we can write $\sum k = x_1 \cdots x_n \cdot \partial_1 \cdots \partial_n = x_1 \cdots x_n \cdot \partial_n$, for any $a, b \in N_0^n$. By definition, a monomial $\sum k$ has degree $[l]_q := [0, \ldots, 0]$ if and only if $a = b$. The following lemma shows how we can rewrite every homogeneous polynomial of degree $[l]_q$ in $A_n$ (resp. $Q_n$) as a polynomial in $K[\theta]$.

**Lemma 2** (Compare with [24], Lemma 3.1.3). In $A_n$, we have the identity $x^m \partial^m = \prod_{i=0}^{m-1} (\theta_i - j_1)$, for $m \in \mathbb{N}$ and $i \in \mathbb{N}$. In $Q_n$, one can rewrite $x^m \partial^m$ as element in $K[\theta]$ and it is equal to $\frac{1}{q_i^{m-1}} \prod_{i=0}^{m-1} (\theta_i - j_1)_i$, where $T_j$ denotes the $j$th triangular number, i.e., $T_j := j(j+1)/2$ for all $j \in \mathbb{N}$.

**Corollary 2.** The $0$th graded part of $Q_n$, respectively $A_n$, is $K[\theta_1, \ldots, \theta_n]$.

Recall, that the $r$th graded part for $z \in \mathbb{Z}^n$ of $Q_n$, resp. $A_n$, is defined to be the $K$-vector space:

$$Q^{(r)}_n := K[\sum k = z_1 \cdots z_n, n_2 - n_1 = \cdots = n_2 - n_1 = \cdots = z],$$

i.e., the degree of a monomial is determined by the difference of its powers in the $x_i$ and the $\partial_i$. Moreover, since in a grading $Q^{(r)}_n \subseteq Q^{(r+1)}_n$ holds for all $z_1, z_2 \in \mathbb{Z}^n$, $Q^{(r)}_n$ is naturally a $Q^{(r)}_n$-module.

**Proposition 1.** For $z \in \mathbb{Z}^n \setminus \{0\}$, the $z$th graded part $Q^{(z)}_n$, resp. $A^{(z)}_n$, is a cyclic $K[\theta]$-bimodule, generated by the element $\sum k = z$, exponent vectors of which read for $i \in \mathbb{N}$ as follows:

$$e_i(z) := \begin{cases} -z_i, & \text{if } z_i < 0, \\ 0, & \text{otherwise}, \\ w_i(z) := \begin{cases} z_i, & \text{if } z_i > 0, \\ 0, & \text{otherwise}. \end{cases} \end{cases}$$

**Proof.** A polynomial $p \in Q^{(z)}_n$ resp. $p \in A^{(z)}_n$ is homogeneous of degree $z \in \mathbb{Z}^n$ if and only if every monomial of $p$ is of the form $\sum k = z \cdot \partial^{\sum k = w}$, where $k \in \mathbb{N}_0$ and $\mathbb{N}_0 := \{k, \ldots, k\}$. By doing a rewriting, similar to the above, we obtain $\sum k = z \cdot \partial^{\sum k = w} = \sum k = z \cdot \partial^{\sum k = w}$, where $\psi(\theta)$ is computed utilizing Lemma 2. Moreover, by Corollary 1, we conclude that $\sum k = z \cdot \partial^{\sum k = w} = \partial^{\sum k = w}$, or, equivalently, $\sum k = z \cdot \partial^{\sum k = w}$ or $\psi(\theta)$ holds for all $z \in \mathbb{Z}^n$.

Therefore, the factorization of a homogeneous polynomial of degree zero can be done by rewriting the polynomial as an element in $K[\theta]$ and applying a commutative factorization on the polynomial, a much-better-understood problem which is also well implemented in every computer algebra system.

Of course, this would not be a complete factorization, as there are still elements irreducible in $K[\theta]$ which are reducible in $Q_n$, resp. $A_n$. An obvious example are the $\theta_i$ themselves. Fortunately, there are only $2n$ monic polynomials irreducible in $K[\theta]$ that are reducible in $A_n$, resp. $Q_n$, and these are of quite a special form. This extends the proof for $A_n$ and $Q_n$ presented in [14].

**Lemma 3.** Let $i \in \mathbb{N}$. The polynomials $\theta_i$ and $\theta_i + \frac{1}{q_i}$ are the only irreducible monic elements in $K[\theta]$ that are reducible in $Q_n$. Respectively, $\theta_i$ and $\theta_i + 1$ are the only irreducible monic polynomials in $K[\theta]$ that are reducible in $A_n$.

**Proof.** We only consider the proof for $A_n$, as the proof for $Q_n$ is done in an analogous way. Let $f(\theta) \in K[\theta]$ be a monic polynomial. Assume that it is irreducible in $K[\theta]$, but reducible in $A_n$. Let $\varphi, \psi$ be elements in $A_n$ with $\varphi \cdot \psi = f$. Then $\varphi$ and $\psi$ are homogeneous and $\varphi, \psi \in A^{(z)}_n$, $\psi \in A^{(z)}_n$, for a $z \in \mathbb{Z}^n$. Let $[e, w] := [e, w] \cdot [z, w] \neq 0$ be as in Proposition 1. Note, that then $w(-z) = e(z) = e$ and $e(-z) = w(z) = w$ holds.

That is, $A^{(z)}_n = K[\theta] \cdot \sum k = w$ whereas $A^{(z)}_n = K[\theta] \cdot \sum k = w$. Then for $\varphi, \psi \in K[\theta]$, we have $\varphi = \varphi(\theta) \cdot \sum k = w$ and $\psi = \psi(\theta) \cdot \sum k = w$. Using Corollary 1, we obtain $f(\theta) = \sum k = w$, $\sum k = w$, $\sum k = w$, $\sum k = w$, $\sum k = w$, $\sum k = w$, $\sum k = w$. Therefore, the case that $z_i < 0$, then $e(z) = w(z) = 0$ on all but 1th place. By the irreducibility assumption on $f(\theta) \in K[\theta]$ we must have $\varphi, \psi \neq K[\theta]$; since $\varphi$ is monic, we must also have $\varphi = \varphi(\theta) \cdot \sum k = w$. By Lemma 2 we obtain $\theta_i = 1$. As a result, the only possible $f$ in this case is $f = \theta_i + 1$. For analogous reasons for the case when $z_i < 0$, we conclude, that the only possible $f$ in that case is $f = \theta_i$.

The result in Lemma 3 provides us with an easy way to factor a homogeneous polynomial $p \in A_n$, resp. $p \in Q_n$, of degree 0. Obtaining one possible factorization into irreducible polynomials can be done using the following steps:

1. Rewrite $p$ as an element in $K[\theta]$.
2. Factorize this resulting element in $K[\theta]$ with commutative methods.
3. If there is $\theta_i$ or $\theta_i + 1$ for $i \in \mathbb{N}$ among the factors, rewrite it as $x_i \cdot \partial_i$ resp. $\partial_i \cdot x_i$. 

As mentioned earlier, the factorization of a polynomial in a noncommutative ring is unique up to a weak similarity [8]. This notion is much more involved than the similarity up to multiplication by units or up to interchanging factors, as in the commutative case. Indeed, several different nontrivial factorizations can occur. Fortunately, in the case of the polynomial first $(g)$-Weyl algebra, there are only finitely many different nontrivial factorizations possible due to [27]. In order to obtain all these different factorizations, one can apply the commutation rules for $x_i$ and $\partial_i$ with $\theta_i$ for $i \in \mathbb{N}$. That these are all possible factorizations up to multiplication by units can be seen using an analogue approach as in the proof of Lemma 3. Consider the following example.

Example 2. Let $p := x_1^2 x_2 \partial_2^2 + 2x_1 x_2 \partial_1 \partial_2 + x_1 \partial_1 + 1 \in A_2$. The polynomial $p$ is homogeneous of degree $1$, and hence belongs to $K[\partial]$ as $\theta_1 (\theta_1 - 1) \partial_2 + 2 \theta_1 \partial_2 + \theta_1 + 1$. This polynomial factorizes in $K[\partial]$ into $(\theta_1 \partial_2 + 1)(\theta_1 + 1)$. Since $\theta_1 + 1$ factorizes as $1 \cdot x_1$, we obtain the following possible different nontrivial factorizations:

$(\theta_1 \partial_2 + 1) \cdot 1 = \partial_1 \cdot x_1$.

Note that $x_1 \partial_1 + 1$ is not irreducible, since it factorizes nontrivially as $1 \cdot x_1$.

Now we consider the factorization of homogeneous polynomials of arbitrary degree $z \in \mathbb{Z}^n$. Fortunately, the hard work is already done in Proposition 1. Indeed, one factorization of a homogeneous polynomial $p \in Q_\mathbb{N}$, resp. $p \in A_\mathbb{N}$, of degree $z \in \mathbb{Z}^n$ can be obtained using the following steps.

1. Write $p(X, D) = \bar{p}(\theta) X^e D^w$, where $\bar{p} \in A_{\mathbb{N}}[\theta] = K[\theta]$ and $e, w$ are constructed according to Proposition 1.

2. Factorize $\bar{p}$ using the technique described for polynomials of degree $\bar{p}$. Append to such a factorization the natural expansion of the monomial $X^e D^w$ into the product of occurring single variables.

This leads to one nontrivial factorization. A characterization of how to obtain all factorizations is given provided by the following lemma.

Lemma 4. Let $z \in \mathbb{Z}^n$ and $p \in A_{\mathbb{N}}$ (resp. $p \in Q_{\mathbb{N}}$), is monic. Suppose, that one factorization has been constructed as above and has the form $p = T(\theta) X^e D^w$, where $T(\theta) = \prod \{x_i \theta_i + 1 \mid i \in \mathbb{N}\}$ is a product of irreducible factors in $K[\theta]$, which are reducible in $A_{\mathbb{N}}$, resp. $Q_{\mathbb{N}}$. Then $p$ can be factored into $T(\theta) X^e D^w$ by using two operations, namely (i) “swapping”, that is interchanging two adjacent factors according to the commutation rules and (ii) “rewriting” of occurring $\theta_i$ resp. $\theta_i + 1 (\theta_i + \frac{1}{2} \in \mathbb{Q}$-Weyl case) by $x_i$, $\partial_i$, resp. $\partial_i \cdot x_i$. 

Proof. Since $p$ is homogeneous, all $p_i$ for $i \in m$ are homogeneous, thus each of them can be written in the form $p_i = p_i(\theta) \cdot X^{e_i} D^{w_i}$, where $e_i, w_i \in N_{\mathbb{N}}$. With respect to the commutation rules as stated in Corollary 1, we can swap the $p_i(\theta)$ to the left for any $2 \leq i \leq m$. Note that it is possible for them to be transformed to the form $\theta_i + 1 (\theta_i + \frac{1}{2} \in \mathbb{Q}$-Weyl case), $i \in \mathbb{N}$, after performing these swapping steps. I.e., we have commuting factors, both belonging to $Q[\theta]$, as well as to $T(\theta)$ at the left. Our resulting product is thus $\bar{Q}(\theta) T(\theta) \prod_{i=1}^m X^{e_i} D^{w_i}$, where the factors in $\bar{Q}(\theta)$, resp. $T(\theta)$, contain a subset of the factors of $\bar{Q}(\theta)$ resp. $T(\theta)$.

By our assumption of $p$ having degree $z$, we are able to swap $X^e D^w$ to the right in $F := \prod_{i=1}^m X^{e_i} D^{w_i}$, i.e., $F = \bar{F} X^e D^w$ for $\bar{F} \in A_{\mathbb{N}}$. This step may involve combining some $x_i$ and $\partial_i$ to $\theta_i$ resp. $\theta_i + 1$, $j \in \mathbb{N}$. Afterwards, this is also done to the remaining factors that belong to $\bar{Q}(\theta)$, resp. $T(\theta)$, and can be swapped commutatively to their respective positions. Since reverse engineering of those steps is possible, we can derive the factorization $p_1 \cdots p_m$ from $\bar{Q}(\theta) \cdot T(\theta) \cdot X^e D^w$ as claimed. $

Summarizing, we are now able to effectively factor graded polynomials in the $n$th Weyl and $\mathbb{Q}$-Weyl algebra. All different factorizations are obtainable using our technique.

Remark 2. A reader might ask what are the merits of our “graded-driven” approach as opposed to a somewhat more direct approach to factorization using leading monomials. Since, for monomials $m, m' \in A_n$, one has $\text{lexp}(m \cdot m') = \text{lexp}(m) + \text{lexp}(m')$, indeed $h = p \cdot q$ implies $\text{lexp}(p) + \text{lexp}(q) = \text{lexp}(h)$. Thus by considering, say, degree lexicographic ordering on $A_n$, the set $C_h := \{a, b \in \mathbb{N}^n \times \mathbb{N}^n : a + b = \text{lexp}(h)\}$ contains all possible pairs of leading monomials of $p$ and $q$. Then, since with respect to the chosen ordering, for any monomials there are only finitely many smaller monomials, one can make an ansatz with unknown coefficients for $p$ and $q$. Each $(a, b) \in C_h$ leads to a system of nonlinear polynomial equations in finitely many variables.

We compare this “leading monomial” approach with our “graded-driven” one. At first, the factorization of a $\mathbb{Z}^n$-graded polynomial, which is very easy to accomplish with our approach, requires solving of several systems within the leading monomial approach. Second, the number of all elements in the set $C_h$ above is significantly bigger than the number of factorizations of the highest graded part of a polynomial, say $\bar{Q}(\theta) X^e D^w$: suppose that $\bar{p}(\theta)$ is irreducible over $K[\theta]$. Then factorization with the “graded-driven” approach are obtained via moving $x$, resp. $\partial$, past $\bar{p}(\theta)$ to the left. Thus the number of such factorizations is much smaller than the number of ways of writing the exponent vector of $\text{lm}(\bar{p}(\theta) X^e D^w) = \bar{p} X^e D^w$ as a sum of two exponent vectors.

In the next section, we show how the developed technique helps us to tackle the factorization problem for arbitrary polynomials in $A_n$.

3. FACTORING ARBITRARY POLYNOMIALS

3.1 Preliminaries

The techniques described in this section solve the factorization problem in $A_n$. A generalization for $Q_n$ is more involved and the subject of ongoing research.

We begin by fixing some notation used throughout this section. From now on, let “$<$” be an ordering on $\mathbb{Z}^n$ satisfying the conditions of Remark 1. Let $h \in A_n$ be the polynomial we want to factorize. As we are deducing information from the graded su mmands of $h$, let furthermore $M := \{z^{(1)}, \ldots, z^{(m)}\}$, where $m \in \mathbb{N}$ and $z^{(1)} > \ldots > z^{(m)}$, be a finite subset of $\mathbb{Z}^n$ containing the degrees of those graded su mmands. Hence, $h$ can be written in the form $h = \sum_{z \in M} h_z \in A_n$, where $h_z \in A_{\mathbb{N}}$ for $z \in M$. Let us assume that $h$ possesses a nontrivial factorization of at least two factors, which are not necessary irreducible. Moreover,
we assume that $m > 1$, which means that $h$ is not graded, since we have dealt with graded polynomials in $A_n$ already. Let us denote the factors by

$$
  h = \sum_{z \in M} h_z := (p_{n_1} + \ldots + p_{n_\ell})(q_{n_1} + \ldots + q_{n_\ell}),
$$

where $n_1 > n_2 > \ldots > n_\ell$ and $\mu_1 > \mu_2 > \ldots > \mu_\ell \in \mathbb{Z}^n$. $p_{n_i} \in A(n_i)$ for all $i \in \mathbb{I}$, $q_{n_i} \in A(n_i)$ for all $j \in \mathbb{I}$. We assume that $p$ and $q$ are not graded, since we could easily obtain those factors by simply comparing all factorizations of the graded summands in $h$. In general, while trying to find a factorization of $h$, we assume that the values of $k$ and $l$ are not known to us beforehand. We will soon see how we can obtain them. One can easily see that $h_{z(1)} = p_{n_1}q_{n_1}$ and $h_{z(m)} = p_{n_\ell}q_{n_\ell}$, as the degree-wise biggest summand of $h$ can only be combined by multiplication of the highest summands of $p$ and $q$: analogously this holds for the degree-wise lowest summand.

A finite set of candidates for $p_{n_1}, q_{n_1}, p_{n_\ell}$ and $q_{n_\ell}$ can be obtained by factoring $h_{z(1)}$ and $h_{z(m)}$ using the technique described in the previous section. Since the set of candidates is finite, we can assume that the correct representatives for $p_{n_1}, q_{n_1}, p_{n_\ell}$, and $q_{n_\ell}$ are known to us. In practice, we would apply our method to all candidates and would succeed in at least one case to factorize the polynomial due to our assumption of $h$ being reducible.

One may ask now how many valid degrees could occur in summands of such factors $p$ and $q$, i.e., what are the values of $\ell$ and $k$. Theoretically, there exist choices for $\eta_1$ and $\eta_k$ (resp. $\mu_1$ and $\mu_\ell$) where there are infinitely many $z \in \mathbb{Z}^n$, such that $n_1 > z > n_k$ (resp. $\mu_1 > z > \mu_\ell$). Fortunately, only finitely many are valid degrees that can appear in a factorization, as the next lemma shows.

**Lemma 5.** For fixed $h, p_{n_1}, q_{n_1}, p_{n_\ell}$, and $q_{n_\ell} \in A_n$ fulfilling the assumptions stated above, there are only finitely many possible $\eta_i$, resp. $\mu_i, i \in \mathbb{I}$, that can appear as degrees for graded summands in $p$ and $q$.

**Proof.** For every variable $x_i \in \{x_1, \ldots, x_n, t_1, \ldots, t_n\} \subset A_n$, there exists a $x_i \in A_0$ that represents the maximal degree of $v$ that occurs among the monomials in $h$. The number $j$ can be seen as a lower bound of the associated position of $v$ in $\eta_i$, resp. $\mu_i$, if $v$ is one of the $x_i$, or as an upper bound if $v$ is one of the $t_i$. If the degree of one of the homogeneous summands of $p$ or $q$ would grow higher, resp. lower, than this degree-bound, $v$ would appear in $h$ in a higher degree than $j$, which contradicts our choice of $j$. \hfill $\square$

**Example 3.** Let us consider

$$
  h = x_3 \partial_3 \partial_2 + \partial_1 + x_1 x_2 \partial_3^2 + 4 \partial_2 + 4 x_1 \partial_1 \in A_2.
$$

One possible factorization of $x_3 \partial_3 \partial_2 + \partial_1$ is $\partial_2 \cdot x_2 \partial_1 =: p_{n_1} \cdot q_{n_1}$, and, on the other end, one possible factorization of $4 x_1 \partial_1$ is $x_1 \partial_1 \cdot 4 =: p_{n_\ell} \cdot q_{n_\ell}$. Concerning $p$, there are no elements in $\mathbb{Z}^n$ that can occur between deg($p_{n_1}$) = $[0, 1]$ and deg($p_{n_\ell}$) = $[0, 0]$; therefore we can set $k = 2$. For $q$, the only degree that can occur between deg($q_{n_1}$) = $[1, -1]$ and deg($q_{n_\ell}$) = $[0, 0]$ is $[0, 1]$, as every variable except $\theta_1$ appears with maximal degree 1 in $h$. We have $l = 3$ in this case.

### 3.2 Reduction to a Commutative System

In the previous subsection we saw that, given $h \in A_n$, that possesses a factorization as in (1), we are able to obtain the elements $p_{n_1}, q_{n_1}, p_{n_\ell}$ and $q_{n_\ell}$. Moreover, we can compute the numbers $k$ and $l$ of homogeneous summands in the factors. Now our goal is to find values for the unknown homogeneous summands, i.e. $p_{n_2}, \ldots, p_{n_{k-1}}, q_{n_2}, \ldots, q_{n_{l-1}}$. Our goal is to reduce this to a commutative problem to the greatest extent we can.

For this, we use Proposition 1 and define for all $i \in \mathbb{I}$ the polynomial $\tilde{p}_{n_i} \in A(n_i)$ by $\tilde{p}_{n_i} \sum_{t \in \mathbb{I}} \partial_t = p_{n_i}$. In the same way we define $\tilde{q}_{n_i}$ for all $j \in \mathbb{I}$ and $\tilde{h}_{z(m)}$ for $z \in M$. We are now aware of the $\tilde{h}_{z(m)}$ can easily be obtained from the input polynomial $h$. We can refer to the $\tilde{h}_{z(m)}, \tilde{p}_{n_i}, \tilde{q}_{n_i}$ as elements in the commutative ring $\mathbb{K}[\theta]$ using Lemma 2.

The next fact about the degree of the remaining unknowns can be easily proven and is useful for our further steps.

**Lemma 6.** The degree of the $\tilde{p}_{n_i}$ and the $\tilde{q}_{n_i}, (i, j) \in \mathbb{I} \times \mathbb{I}$ in $\theta_i$, $i \in \mathbb{I}$, is bounded by $\min\{\deg_{\theta_i}(h), \deg_{\theta_j}(h)\}$, where deg$_{\theta_i}(f)$ denotes the degree of $f \in A_n$ in the variable $\theta_i$.

There are certain equations that the $\tilde{p}_{n_i}$ and the $\tilde{q}_{n_i}$ must fulfill in order for $p$ and $q$ to be factors of $h$.

**Definition 5.** For $\alpha, \beta \in \mathbb{Z}^n$ we define $\gamma_{\alpha, \beta} = \prod_{i=1}^{n} z_{a_i}^{b_i}$ in the latter expression we define for $a, b \in \mathbb{K}$ and $\kappa \in \mathbb{N}$

$$
  z_{a, b} := \begin{cases}
  1, & a > 0 \text{ and } b > 0, \\
  \prod_{t \in \mathbb{I}}^{|a|} (\theta_t - \tau), & a < 0 \text{ or } b = 0, |a| \leq |b|, \\
  \prod_{t \in \mathbb{I}}^{|b|} (\theta_t - \tau - |a| + |b|), & a = 0, b > 0, |a| > |b|, \\
  \prod_{t \in \mathbb{I}}^{|a|} (\theta_t + \tau), & a > 0, b = 0, |a| > |b|, \\
  \prod_{t \in \mathbb{I}}^{|b|} (\theta_t - \tau + |a| - |b|), & a = 0, b > 0, |a| > |b|.
  \end{cases}
$$

**Theorem 1.** Suppose that, with the notation as above, we have $h = pq$ and $\tilde{p}_{n_1}, \tilde{q}_{n_1}, \tilde{p}_{n_\ell}, \tilde{q}_{n_\ell}$. $\tilde{h}_{z(m)}$ are known.

Define $\tilde{h}_z := 0$ for $z^{(1)} > z \geq z^{(m)}$ and $z \notin M$. Then the remaining unknown $\tilde{p}_{n_2}, \ldots, \tilde{p}_{n_{k-1}}, \tilde{q}_{n_2}, \ldots, \tilde{q}_{n_{l-1}}$ are solutions of the following finite set of equations:

$$
  \{ \sum_{\kappa \in \mathbb{N}^{n}} \tilde{p}_{\kappa} \mu_{\kappa} = \tilde{h}_z \} \subset \mathbb{Z}^{n}, z^{(1)} \geq z \geq z^{(m)}.
$$

Moreover, a factorization of $h$ in $A_n$ corresponds to $\tilde{q}_n$ and $\tilde{p}_n$ for $(i, j) \in \mathbb{I} \times \mathbb{I}$ being polynomial solutions with bounds as stated in Lemma 6.

**Proof.** We only sketch this technical proof. Inspecting the product in (2), we split it into its graded summands. By repeated application of Lemma 1, we arrive at the described set of equations via coefficient comparison. The degree bound has been established in Lemma 6 above. \hfill $\square$

**Corollary 3.** The problem of factorizing a polynomial in the nth Weyl algebra can be solved via finding polynomial univariate solutions of degree at most $2 \cdot \sum_{t \in \mathbb{I}} |\deg(h)|$ for a system of difference equations with polynomial coefficients, involving linear and quadratically nonlinear inhomogeneous equations.

As this part of the method is rather technical, let us illustrate it via an example.

**Example 4.** Let $p := \theta_1 \partial_3 + (\theta_1 + 3) \partial_2 + x_2$, $q := (\theta_1 + x_1) \partial_3 + x_2 + (\theta_1 + 1) x_1 x_2 \in A_2$ and $h := \theta_1 \partial_3 + (\theta_1 + 3) \partial_2 + x_2$.
\[ h := pq = \theta_1(\theta_1 + 4)x_1\partial_2^2 \]  
\[ + (\theta_1(\theta_1 - 1)x_2 + 8\theta_1\partial_2 + \theta_1 + 12\partial_2)x_1\partial_2 \]  
\[ + (\theta_1(\theta_1 - 1)x_1 + \partial_2^2 - \theta_1 + 4\theta_1 + 2\theta_1 + 7\theta_2)x_1 \]  
\[ + (\theta_1(\theta_1 - 1)x_2 + 5\theta_1\partial_2 + 3\theta_2 + 1)x_1\cdot x_2 \]  
\[ + (\theta_1 + 1)x_1\partial_2^2 \]

We have written every coefficient in terms of the \( \theta_i \) already for better readability.

By assumption, the only information we have about \( p \) and \( q \) are the values of \( \{0, 1\} =: p_{0, 1} \). \( p_{[0, 1]} := p_{[0, 1]} \). \( q_{[-1, 1]} := q_{[-1, 1]} \). Thus we have, using the above notation, \( \tilde{p}_{q_{[-1, 1]}} = \theta_1 \), \( \tilde{p}_{q_{[-1, 1]}} = 1 \), \( \tilde{q}_{q_{[-1, 1]}} = (\theta_1 + 4) \), and \( \tilde{q}_{q_{[-1, 1]}} = (\theta_1 + 1) \). We set \( k := 1 = 3 \) and it remains to solve for \( \tilde{q}_{q_{[-1, 1]}} \) and \( \tilde{p}_{p_{[0, 1]}} \).

In \( h \), every variable appears in degree 2, except from \( x_1 \), which appears in degree 3. That means that the degree bounds for \( \theta_1 \) and \( \theta_2 \) in \( \tilde{q}_h \) can be set to be two.

The product of \( \{p_{[0, 1]} + p_{[0, 2]} + p_{0, 3} \}(q_{[1, 1]} + q_{[0, 2]} + q_{[0, 3]}) \) with known values inserted is

\[ \begin{align*}
\theta_1(\theta_1 + 4)x_1\partial_2^2 \\
+ (\theta_1\tilde{q}_{q_{[2, 1]}}(\theta_1, \theta_2 + 1) + \tilde{p}_{p_{[1, 1]}}, \theta_1 + 4)x_1\partial_2 \\
+ (\theta_1(\theta_1 + 1)x_1 + (\theta_1 + 4)x_2 + \tilde{p}_{p_{[2, 1]}}, x_1\partial_2 \\
+ (\tilde{q}_{q_{[2, 1]}}, \theta_1 - 1) + \tilde{p}_{p_{[1, 1]}}, x_1\partial_2 \\
+ (\theta_1 + 1)x_1\partial_2^2
\end{align*} \]

Using Lemma 7, we can reduce the unknowns we need to solve for to the \( \tilde{q}_{\mu} \). Sorting the equations in the set (2) from highest to lowest, we can rearrange them by putting the \( \tilde{p}_{\mu} \) on the left hand side and backwards substituting the appearing \( \tilde{p}_{\mu} \) on the respective right hand side by the formulae in the former equations. The same can be done for sorting the equations from lowest to highest, which lead to a second – different – set of equations for the \( \tilde{p}_{\mu} \). The remaining step is then to concatenate the two respective descriptions for the \( \tilde{p}_{\mu} \) and then solve the resulting nonlinear system of equations in the coefficients of the \( \tilde{q}_{\mu} \) using e.g. Gröbner bases [7]. We depict this process in the next example.

**Example 5.** Let us consider \( h = pq \) from Example 4, using all notations that were introduced there.

We assume that the given form of \( \tilde{p}_{\mu} \) is \( \tilde{p}_{\mu} = \tilde{p}_{\mu, 0} + \tilde{p}_{\mu, 1}(\theta_1 + \tilde{p}_{\mu, 2}(\theta_2 + \tilde{p}_{\mu, 3}(1 + \tilde{p}_{\mu, 4}))) \). The proof of this statement can be obtained using coefficient comparison, one can form from this equation \( \partial_t \mu = \tilde{\eta} \). Hence, we get

\[ \begin{align*}
\tilde{p}_{\mu} & = \theta_1(\theta_1 - 1)x_2 + 8\partial_1\partial_2 + \theta_1 + 12\partial_2 \cdot \tilde{q}_{\mu, 0}(\theta_1, \theta_2 + 1) \\
& = \theta_1(\theta_1 - 1)x_2 + 5\theta_2 + 3\theta_2 + 1 - \tilde{q}_{\mu, 2}(\theta_1, \theta_2 - 1).
\end{align*} \]

Thus, \( \tilde{q}_{\mu, 2} \) has to fulfills the equation

\[ \begin{align*}
\tilde{q}_{\mu, 2}(\theta_1 - 1)x_2 + 8\theta_1\partial_2 + \theta_1 + 12\partial_2 - \tilde{q}_{\mu, 2}(\theta_1, \theta_2 + 1) \cdot \theta_1 + 1 \\
= \theta_1(\theta_1 - 1)x_2 + 5\theta_2 + 3\theta_2 + 1 - \tilde{q}_{\mu, 2}(\theta_1, \theta_2 - 1).
\end{align*} \]

Note here, that we could consider more equations that \( \tilde{q}_{\mu, 2} \) must fulfill, but we refrained from it in this example for the sake of brevity.

Using coefficient comparison, one can form from this equation a nonlinear system of equations with the \( \tilde{q}_{\mu, i} \) in \( \tilde{\mu} \), as indeterminates. The reduced Gröbner basis of this system is \( \tilde{q}_{\mu, 0}(1), \tilde{q}_{\mu, 1}(2), \ldots, \tilde{q}_{\mu, 8}(8) \), which tells us, that \( \tilde{q}_{\mu, 1} = 1 \) and hence, \( \tilde{p}_{\mu} = (\theta_1 + 3)x_2 \). Thus, we have exactly recovered both \( p \) and \( q \) in the factorization of \( h \). The concrete original system is stated in Appendix A.

This approach of course raises the question, if those systems of equations that we construct are over- resp. under-determined. In the latter case, we might end up with some ambiguity regarding the solutions of the systems. The next lemma will show that our construction in fact leads to an overdetermined system.

**Lemma 8.** Let \( v \) denote amount of the vectors in \( \mathbb{N}_0^k \), that are in each component \( t \) smaller or equal to \( \min\{\deg_1(h), \deg_2(h)\} \). After the reduction of the unknowns to \( \tilde{q}_{\mu} \), for \( i \in \{2, \ldots, l - 1\} \), the amount of equations satisfied by the \( \tilde{q}_{\mu} \) will be between \( 2(l - 1)v \) and \( (l - 1)^2v \), and the amount of variables that we have to solve for is \( (l - 2)v \).

**Proof.** The number \( (l - 2) \cdot v \) is obvious for the number of unknowns, as we have for every polynomial \( \tilde{q}_{\mu} \), for \( i \in \{2, \ldots, l - 1\} \) exactly \( v \) unknown coefficients.

In order to obtain expressions for our unknowns, we are considering two times \( l - 1 \) equations of the set in (2), namely \( l - 1 \) equations starting from the bottom and \( l - 1 \) equations starting from the top. Note here, that we also consider the equation for \( \tilde{q}_{\mu} \) when starting from the top, and the equation for \( \tilde{q}_{\mu, 1} \) when starting from the bottom, as we obtain...
more equations fulfilled by the unknown variables in this way, where part of it is known to us. In the backwards substitution phase, we obtain different products of the polynomials \( q_{ji} \). The amount of terms in the \( \theta_j \) for \( j \in \mathbb{N} \) of those products is greater or equal to \( 2 \cdot \nu \) and at most \((I - 1) \cdot \nu \). This leads to the claimed bounds. \( \square \)

### 3.4 Application to Weyl Algebras with Rational Coefficients

In practice, one is often interested in differential equations over the field of rational functions in the indeterminates \( x_i \). We refer to the corresponding operator algebras as the rational Weyl algebras. We have the same commutation rules there, but with extension to the case where \( x_i \) appears in the denominator. These algebras can be recognized as Ore localization of polynomial Weyl algebras with respect to the multiplicatively closed Ore set \( S = \mathbb{K}[x_1, \ldots, x_n] \setminus \{ 0 \} \).

Unlike in the polynomial Weyl algebra, an infinite number of nontrivial factorizations of an element is possible. The easiest example is the polynomial \( \partial_x^2 \in A_1 \), having nontrivial factorizations \( (\partial_x + \frac{1}{x})(\partial_x - \frac{1}{x}) \) for all \( e \in \mathbb{K} \); the only polynomial factorization is \( \theta \cdot \bar{\theta} \). Thus, at first glance, the factorization problem in both the rational and the polynomial Weyl algebras seems to be distinct in general. But there are still many things in common.

Consider the more general case of localization of Ore algebras. In what follows, we denote by \( S \subset R \) the denominator set of an arbitrary localization of a Noetherian integral domain \( R \). For properties that \( S \) has to fulfill and calculation rules of elements in \( S^{-1} R \) please consider [8], Chapter 8. Let us clarify the connection between factorizations in \( S^{-1} R \) and factorizations in \( R \).

**Theorem 2.** Let \( h \) be an element in \( S^{-1} R \setminus \{ 0 \} \). Suppose, that \( h = h_1 \cdots h_m, m \in \mathbb{N}, h_i \in S^{-1} R \) for \( i \in m \). Then there exists \( q \in S \) and \( \tilde{h}_1, \ldots, \tilde{h}_m \in R \), such that \( q \tilde{h} = \tilde{h}_1 \cdots \tilde{h}_m \).

Thus, by clearing denominators in an irreducible element in \( S^{-1} R \) one obtains an irreducible element in \( R \). The other direction does not hold in general. However, one can use our algorithms in a pre-processing step of finding factorization over \( S^{-1} R \). In particular, a reducible element of \( R \) is necessarily reducible over \( S^{-1} R \).

The theorem says that we can lift any factorization from the ring \( S^{-1} R \) to a factorization in \( R \) by a left multiplication with an element of \( S \). This means that in our case, where \( S = \mathbb{K}[x_1, \ldots, x_n] \setminus \{ 0 \} \), it suffices to multiply a polynomial \( h \) by a suitable element in \( \mathbb{K}[x_1, \ldots, x_n] \) in order to obtain a representative of a rational factorization. Finding this element is subject of future research. As we already have shown in [14], a polynomial factorization of an element in \( A_n \) is often more readable than the factorization produced by rational factorization methods. Thus a pre-computation that finds such a premultiplier so that we can just perform polynomial factorization would be a beneficial ansatz in the rational factorization.

**Example 6.** Consider the polynomial \( h := \partial_x^2 - x_1 \partial_x - 2 \in A_1 \), \( h \) is irreducible in \( A_1 \), but in the first rational Weyl algebra, we obtain a factorization given by \((\partial_x + \frac{1}{x})(\partial_x - \frac{1}{x})\). If we multiply \( h \) by \( x_1 \) from the left, our factorization method reveals two different factorizations. The first one is \( x_1 \cdot h \) itself, and the second one is given by \( \partial_x \cdot (x_1 \partial_x^2 - x_1) \), which represents the rational factorization in the sense of Theorem 2.

### 3.5 Application to Shift Algebras (with Rational Coefficients)

With the help of the Lemma 1 one can see that \( S_n \) is a subalgebra of the \( n \)th Weyl algebra \( A_n \) via the following homomorphism of \( \mathbb{K} \)-algebras: \( \iota: S_n \to A_n, \ x_i \mapsto \theta_i, \ s_j \mapsto \partial_j \). One can easily prove that \( \iota \) is, in fact, a monomorphism. This observation leads to the following result, which tells us that we do not have to consider the algebra \( S_n \) separately when dealing with factorization of its elements.

**Corollary 4.** The factorization problem for a polynomial \( p \in S_n \) can be obtained from the solution of a factorization problem of \( \iota(p) \in A_n \) by refining.

Theorem 2 also applies to the rational shift algebra. Thus, the approach to lift factorizations in the shift algebra with rational coefficients can also be applied here. The remaining research is also here to find suitable elements in \( \mathbb{K}[x_1, \ldots, x_n] \) for pre-multiplication.

### 4. IMPLEMENTATION AND TIMINGS

We have implemented the described method for \( A_n \) in the computer algebra system SINGULAR. Our goal was to test the performance of our approach and the versatility of the results in practice and compare it to given implementations. Our implementation is in a complete but experimental stage, and we see potential for optimization in several areas.

The implementation extends the library nc_factor.lib, which contains the functionality to factorize polynomials in the first Weyl algebra, the first shift algebra and graded polynomials in the first q-Weyl algebra. The actual library is distributed with SINGULAR since version 3-1-3.

In the following examples, we consider different polynomials and present the resulting factorizations and timings. Our function to factorize polynomials in the \( n \)th Weyl algebra is written to solve problem (ii) as given in the introduction, i.e. finding all possible factorizations of a given polynomial. All computations were done using SINGULAR version 3-1-6. We compare our performance and our outputs to REDUCE version 3.8. There, we use the function nc_factorize.all in the library RC POLY. The calculations were run on a computer with a 4-core Intel CPU (Intel®Core™ i7-3520M CPU with 2.90GHz, 2 physical cores, 2 hardware threads, 32K L1[i,d], 256K L2, 4MB L3 cache) and 16GB RAM.

In order to make the tests reproducible, we used the SDEVAL [15] framework, created for the SYMBOLIC DATA project [5], for our benchmarking. The functions of SYMBOLIC DATA as well as the data are free to use. In such a way our comparison is easily reproducible by any other person.

Our set of examples is given by

\[
\begin{align*}
\text{h}_1 & := (\partial_1 + 1)^2(\partial_1 + x_1 \partial_2) \in A_2, \\
\text{h}_2 & := (\partial_1 \partial_2 + (\partial_1 + \partial_2 + x_2)) \cdot ((\partial_1 + 2)x_1 \partial_2 + x_1 x_2) \in A_2, \\
\text{h}_3 & := x_1 x_2 \partial_2^2 \partial_1^2 + x_1 x_2 x_1 \partial_3 \in A_3, \\
\text{h}_4 & := (x_1 \partial_x + x_1 x_2)(\partial_1 \partial_2 + x_1 \partial_x + x_1 \partial^2) \in A_2.
\end{align*}
\]

The polynomial \( \text{h}_1 \) can be found in [18], the polynomial \( \text{h}_2 \) is the polynomial from Example 5 and the last two polynomials are graded polynomials.

Our implementation in SINGULAR managed to factor all the polynomials that are listed above. For \( \text{h}_1 \), it took 2.83s to find two distinct factorizations. Besides the given one above, we have \( \text{h}_1 = (x_1 \partial_x + \partial_2^2 + 2x_1 \partial_1 + \partial_1 + 2 \partial_2)(\partial_1 + 1) \). In order to factorize \( \text{h}_2 \), SINGULAR took 23.48s to find three fac-
torizations. For the graded polynomials $h_3$ and $h_4$, our implementation finished its computations as expected quickly (0.46s and 0.32s) and returned $60$ distinct factorizations for each $h_3$ and $h_4$.

REDUCE only terminated for $h_1$ (within two hours). For $h_1$ it returned $3$ different factorizations (within 0.1s), and one of the factorizations contained a reducible factor. For $h_2$, $h_3$ and $h_4$, we cancelled the process after two hours.

Factoring $\mathbb{Z}$-graded polynomials in the first Weyl algebra was already timed and compared with several implementations on various examples in [14]. The comparison there also included the functionality in the computer algebra system MAPLE for factoring polynomials in the first Weyl algebra with rational coefficients.

The next example shows the performance of our implementation for the first Weyl algebra.

Example 7. This example is taken from [17], page 200. We consider $h := (x_1^2 - 1)x_1^2 + (1 + 7x_1^2)\partial_1 + 8x_1^3$. Our implementation takes $0.75$ seconds to find $12$ distinct factorizations in the algebra $A_1$. MAPLE 17, using DFactor from the DETools package, takes the same amount of time and reveals one factorization in the first Weyl algebra with rational coefficients. REDUCE outputs $60$ factorizations in $A_1$ after 3.27s. However, these factorizations contain factorizations with reducible factors. After factoring such cases and removing duplicates from the list, the number of different factorizations reduced to $12$.

5. CONCLUSIONS

An approach to factoring polynomials in the operator algebras $A_n$, $Q_n$ and $S_n$ based on nontrivial $\mathbb{Z}^n$-gradings has been presented, and an experimental implementation has been evaluated. We have shown that the set of polynomials that we can factorize using our technique in a feasible amount of time has been greatly extended. Especially for $\mathbb{Z}^n$-graded polynomials, we have shown that the problem of finding all nontrivial factorizations in $A_n$ resp. $Q_n$ can be reduced to commutative factorization in multivariate rings and some basic combinatorics. Thus, the performance of the factorization algorithm regarding graded polynomials is dominated by the performance of the commutative factorization algorithm that is available.

Our future work consists of implementing the remaining functionalities into nfactor.lib. Furthermore, it would be interesting to extend our technique to deal with the factorization problem in $A_n$ to polynomials in $Q_n$. Additionally, there exist many other operator algebras, and it would be interesting to investigate to what extent we can use the described methodology there.

Applying our techniques for the factorization problem in the case of algebras with coefficients in rational functions is also interesting, albeit more involved. Amongst other problems, in that case infinitely many different factorizations can occur. One has to find representatives of parametrized factorizations, and use these to obtain a factorization in the polynomial sense. This approach could be beneficial, and it has been developed in [14].

6. ACKNOWLEDGMENTS

We would like to thank to Dima Grigoriev for discussions on the subject, and to Mark van Hoeij for his expert opinion. We are grateful to Wolfram Koepf and Martin Lee for providing us with interesting examples and to Michael Singer, Shaoshi Chen and Daniel Rettstadt for sharing with us interesting points of view on our research problems.

We would like to express our gratitude to the German Academic Exchange Service DAAD for funding our project in the context of the German-Canadian PPP program.

7. REFERENCES

[1] S. Abramov. Problems of computer algebra involved in the search for polynomial solutions of linear differential and difference equations. Mosc. Univ. Comput. Math. Cybern., 1989(3):63–68, 1989.

[2] S. Abramov and M. Petkovšek. On polynomial solutions of linear partial differential and ($q$)-difference equations. In Proc. CASC 2012. Berlin: Springer, 2012.

[3] S. A. Abramov, M. Bronstein, and M. Petkovšek. On polynomial solutions of linear operator equations. In Proc. ISSAC 1995. New York, NY: ACM Press, 1995.

[4] W. D. Andres. Noncommutative Computer Algebra with Applications in in Algebraic Analysis. PhD thesis, RWTH Aachen University, 2013.

[5] O. Buchmann and H.-G. Gräbe. The SymbolicData project. In Reports on Computer Algebra, volume 27, 2000.

[6] R. Beals and E. A. Kartashova. Constructively factoring linear partial differential operators in two variables. Theor. Math. Phys., 145(2):1511–1524, 2005.

[7] B. Buchberger. Introduction to Groebner bases. Berlin: Springer, 1997.

[8] J. Bueso, J. Gómez-Torrecillas, and A. Verschoren. Algorithmic methods in non-commutative algebra. Applications to quantum groups. Dordrecht: Kluwer Academic Publishers, 2003.

[9] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönenmann. SINGULAR 3-1-6 — A computer algebra system for polynomial computations. 2012. http://www.singular.uni-kl.de.

[10] M. Foupouangnigni, W. Koepf, and A. Ronveaux. Factorization of fourth-order differential equations for perturbed classical orthogonal polynomials. J. Comp. Appl. Math., 162(2):299–326, 2004.

[11] I. M. Gel’fand and A. A. Kirillov. Sur les corps Liès aux algèbres enveloppantes des algèbres de Lie. Publications Mathématiques de l’IHÉS, 35:5–19, 1966.

[12] D. Grigoriev and F. Schwarz. Factoring and solving linear partial differential equations. Computing, 73(2):179–197, 2004.

[13] A. Heinle and V. Levandovskyy. Factorization of polynomials in $\mathbb{Z}$-graded skew polynomial rings. ACM Commun. Comput. Algebra, 44(3/4):113–114, 2011.

[14] A. Heinle and V. Levandovskyy. Factorization of $\mathbb{Z}$-homogeneous polynomials in the first ($q$)–Weyl algebra. arXiv preprint arXiv:1302.5674, 2013.

[15] A. Heinle, V. Levandovskyy, and A. Nareike. SymbolicData: SDEval — benchmarking for everyone. arXiv preprint arXiv:1310.5551, 2013.

[16] M. Kashiwara. Vanishing cycle sheaves and holonomic systems of differential equations. In Algebraic Geometry. Springer, 1983.

[17] W. Koepf. Hypergeometric summation. An algorithmic approach to summation and special function identities. Wiesbaden: Vieweg, 1998.

[18] E. Landau. Ein Satz über die Zerlegung homogener linearer Differentialausdrücke in irreductible Factoren. Journal für die reine und angewandte Mathematik, 124:115–120, 1902.

[19] A. Loewy. Über reduzible lineare homogene Differentialgleichungen. Math. Ann., 56:549–584, 1903.

[20] A. Loewy. Über vollständig reduzible lineare homogene Differentialgleichungen. Math. Ann., 62:89–117, 1906.

[21] B. Malgrange. Polynômes de Bernstein-Sato et cohomologie évanescente. Astérisque, 101–102:243–267, 1983.

[22] H. Melenk and J. Apel. REDUCE package NCPOLY:
Computation in non-commutative polynomial ideals.
Konrad-Zuse-Zentrum Berlin (ZIB), 1994.
[23] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn,
S. M. Vorkoetter, J. McCarron, and P. DeMarco. Maple
Introductory Tutorial Guide. Maplesoft, 2008.
[24] M. Saito, B. Sturmfels, and N. Takayama. Gröbner
deformations of hypergéometric differential equations.
Berlin: Springer, 2000.
[25] F. Schwarz. Alltypes in the web. ACM Commun.
Comput. Algebra, 42(3):185–187, Feb. 2009.
[26] S. Tsarev. Problems that appear during factorization of
ordinary linear differential operators. Program.
Comput. Softw., 20(1):27–29, 1994.
[27] S. Tsarev. An algorithm for complete enumeration of all
factorizations of a linear ordinary differential operator.
In Proc. ISSAC 1996. New York, NY: ACM Press, 1996.
[28] M. van Hoeij. Factorization of linear differential operators.
Nijmegen, 1996.
[29] M. van Hoeij. Factorization of differential operators with
rational function coefficients. J. Symb. Comput.,
24(5):537–561, 1997.
[30] M. van Hoeij. Formal solutions and factorization of
differential operators with power series coefficients. J.
Symb. Comput., 24(1):1–30, 1997.
[31] M. van Hoeij and Q. Yuan. Finding all Bessel type solutions
for linear differential equations with rational function
coefficients. In Proc. ISSAC 2010, pages 37–44, 2010.

APPENDIX

A. COMMUTATIVE POLYNOMIAL SYSTEM
OF EQUATIONS

The commutative polynomial system of equations that is
formed in Example 5 is given as follows.

\[\begin{align*}
\{ & -q^{(8)}_{12}, -q^{(7)}_{12}, -2q^{(8)}_{12}, -q^{(6)}_{12}, -q^{(7)}_{12}, -q^{(8)}_{12}, -q^{(5)}_{12}, -q^{(4)}_{12}, -2q^{(5)}_{12}, \\
& -4q^{(8)}_{15}, -q^{(8)}_{15}, -q^{(8)}_{15}, -q^{(8)}_{15}, -2q^{(5)}_{15}, -q^{(5)}_{15}, -2q^{(5)}_{15}, \\
& -4q^{(5)}_{15} + 4q^{(7)}_{15} - 8q^{(8)}_{15}, -q^{(6)}_{15}, -q^{(4)}_{15}, -q^{(5)}_{15} - 2q^{(5)}_{15} + 4q^{(6)}_{15}, \\
& -4q^{(7)}_{15} + 4q^{(5)}_{15} + 1, -4q^{(2)}_{15} + 4q^{(4)}_{15} - 8q^{(6)}_{15} - 2q^{(5)}_{15} + 4q^{(6)}_{15}, \\
& -4q^{(4)}_{15} + 4q^{(5)}_{15}, 4q^{(2)}_{15} - 8q^{(2)}_{15} 4q^{(0)}_{15} - 4q^{(1)}_{15} + 4q^{(2)}_{15} - 4, \\
& (q^{(8)}_{15}), 2q^{(7)}_{15} q^{(8)}_{15} + 2q^{(8)}_{12} q^{(8)}_{15} + (q^{(7)}_{15})^2 + 3q^{(8)}_{12} q^{(8)}_{15}, \\
& \{ & q^{(8)}_{12} q^{(7)}_{15}, 2q^{(6)}_{12} q^{(8)}_{15} + (q^{(7)}_{15})^2 + (q^{(6)}_{15})^2, \\
& +q^{(6)}_{15} q^{(7)}_{15} + q^{(6)}_{15} q^{(8)}_{15} + 2q^{(5)} q^{(8)}_{15}, 2q^{(4)} q^{(8)}_{15} + 2q^{(5)} q^{(8)}_{15} + 4q^{(5)} q^{(8)}_{15} \\
& -q^{(8)}_{15} - 2q^{(3)} q^{(8)}_{15} + 2q^{(4)} q^{(8)}_{15} + 2q^{(5)} q^{(8)}_{15} + 2q^{(6)} q^{(8)}_{15} + 3q^{(5)} q^{(8)}_{15} + 2q^{(5)} q^{(8)}_{15} \\
& +2q^{(5)} q^{(8)}_{15} - q^{(7)}_{15} - 2q^{(3)} q^{(8)}_{15} + 2q^{(3)} q^{(8)}_{15} + 2q^{(4)} q^{(8)}_{15} + 2q^{(4)} q^{(8)}_{15} \\
& +q^{(4)} q^{(8)}_{15} + 2q^{(5)} q^{(8)}_{15} + q^{(5)} q^{(8)}_{15} - q^{(6)} + 2q^{(3)} q^{(8)}_{15} - q^{(7)}_{15} + q^{(3)} q^{(8)}_{15} \\
& +q^{(3)} q^{(8)}_{15} + q^{(4)} q^{(8)}_{15} + 2q^{(3)} q^{(8)}_{15} + 2q^{(5)} q^{(8)}_{15} + (q^{(5)}_{15})^2, 2q^{(1)} q^{(8)}_{15} \\
& +2q^{(2)} q^{(7)}_{15} + 4q^{(2)} q^{(8)}_{15} + 2q^{(2)} q^{(8)}_{15} + 2q^{(2)} q^{(8)}_{15} - q^{(5)}_{15} - q^{(8)}_{15}, \\
& 2q^{(0)} q^{(8)}_{12} + 2q^{(1)} q^{(8)}_{15} + 3q^{(1)} q^{(8)}_{15} + 2q^{(2)} q^{(8)}_{15} + 3q^{(2)} q^{(8)}_{15} \\
& -q^{(8)}_{15} q^{(4)} q^{(8)}_{15} + q^{(4)} q^{(8)}_{15} - q^{(6)} + 2q^{(1)} q^{(8)}_{15} + 2q^{(1)} q^{(8)}_{15} + 2q^{(1)} q^{(8)}_{15} \\
& +q^{(1)} q^{(8)}_{15} + 2q^{(2)} q^{(8)}_{15} + q^{(2)} q^{(8)}_{15} + 2q^{(3)} q^{(8)}_{15} + 2q^{(3)} q^{(8)}_{15} - q^{(5)}_{15} - q^{(3)} q^{(8)}_{15} \\
& +q^{(3)} q^{(8)}_{15} + q^{(4)} q^{(8)}_{15} + 2q^{(3)} q^{(8)}_{15} + q^{(5)} q^{(8)}_{15} - q^{(6)} + 2q^{(1)} q^{(8)}_{15} + q^{(1)} q^{(8)}_{15} \\
& +q^{(1)} q^{(8)}_{15} + q^{(2)} q^{(8)}_{15} + (q^{(3)}_{15})^2 + 3q^{(3)} q^{(8)}_{15} + 3q^{(3)} q^{(8)}_{15} - q^{(6)}_{15}, \\
& \{ & 2q^{(3)} q^{(5)} + 2q^{(1)} q^{(5)} + q^{(2)} q^{(8)}_{15} + q^{(2)} q^{(8)}_{15} - q^{(2)} q^{(8)}_{15} - 7q^{(8)}_{15} - 12q^{(8)}_{15}, \\
& 2q^{(0)} q^{(5)} + 2q^{(1)} q^{(5)} + 3q^{(1)} q^{(5)} - q^{(1)} - 2q^{(2)} q^{(3)} + 3q^{(2)} q^{(4)} \\
& +2q^{(2)} q^{(8)}_{15} - 7q^{(4)}_{15} - q^{(8)}_{15} - 12q^{(6)} - 12q^{(8)}_{15}, \\
& 2q^{(1)} q^{(5)} + q^{(1)} q^{(5)} - q^{(1)} - 2q^{(2)} q^{(3)} + 3q^{(2)} q^{(4)} \\
& +2q^{(2)} q^{(8)}_{15} - 7q^{(4)}_{15} - q^{(8)}_{15} - 12q^{(6)} - 12q^{(8)}_{15}, \\
& 2q^{(0)} q^{(2)} q^{(8)}_{15} + (q^{(1)}_{15})^2 + 3q^{(1)} q^{(8)}_{15} - q^{(1)} + (q^{(2)}_{15})^2 - 7q^{(4)}_{15} - 12q^{(4)}_{15}, \\
& 2q^{(0)} q^{(2)} q^{(8)}_{15} + (q^{(1)}_{15})^2 + 3q^{(1)} q^{(8)}_{15} - q^{(1)} + (q^{(2)}_{15})^2 - 7q^{(4)}_{15} - 12q^{(4)}_{15}, \\
& 7, (q^{(0)}_{15})^2 + (q^{(0)}_{15})^2 + (q^{(0)}_{15})^2 - (q^{(0)}_{15})^2 - 12q^{(4)}_{15} + 12q^{(4)}_{15} + 12q^{(4)}_{15}.
\end{align*}\]