Automated Reasoning For The Existence Of Darboux Polynomials

Khalil Ghorbal, Maxime Bridoux

Automated Reasoning For The Existence Of Darboux Polynomials

Khalil Ghorbal
Inria
Rennes, France
khalil.ghorbal@inria.fr

Maxime Bridoux
Inria
Rennes, France
maxime.bridoux@inria.fr

ABSTRACT

Given a polynomial ordinary differential equation (ODE), we devise *generic* polynomial reduction algorithms to automatically investigate the intertwined relationship between the total degree of (nontrivial) Darboux polynomials and the polynomials defining the ODE. By generic we mean that both the coefficients and the multidegree of the involved polynomials are symbolic. We use Newton polytopes as a light-weight abstraction to select optimal weight monomial orders improving the efficiency of the involved computations. The method works by inferring necessary conditions on both the coefficients and the multidegree for the polynomial to be Darboux. These conditions are then used, via constants' propagation, to restrict the shape of the generic candidate, pinpointing which monomials ought to be preserved by removing the superfluous ones. In some relevant cases, we are able to automatically prove the nonexistence of (nontrivial) Darboux polynomials providing a new toolbox to prove and formally certify that some limit cycles are not algebraic.

CCS CONCEPTS

• **Computing methodologies** → **Theorem proving algorithms**; *Algebraic algorithms*; *Representation of polynomials*; • **Mathematics of computing** → **Ordinary differential equations**.

KEYWORDS

Darboux polynomials, Newton polytope, Theorem proving, Automated reasoning, Van der Pol oscillator, Lienard systems

ACM Reference Format:
Khalil Ghorbal and Maxime Bridoux. 2024. Automated Reasoning For The Existence Of Darboux Polynomials. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '24), July 16–19, 2024, Raleigh, NC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3666000.3669705

1 INTRODUCTION

In his seminal work [6, §II, pp 71-73], Gaston Darboux introduced algebraic *particular integrals*, known today as Darboux polynomials, as a mean to construct (rational) *general integrals* (i.e. first integrals or conserved quantities) for polynomial ODEs (equivalently, polynomial vector fields) of the standard form: [1]

$$\dot{x}_i = f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \ .$$

where $f_1, \ldots, f_n$ are multivariate polynomials in $x_1, \ldots, x_n$ over some field and $\dot{x}_i$ denotes the derivative of x_i with respect to an independent variable t. In the sequel, we represent such a system concisely as $\dot{x} = f(x)$. From a differential algebraic perspective, the ODE defines a polynomial *derivation* $D = \sum_{i=1}^{n} f_i \partial_i$, acting on the ring of polynomials where ∂_i denotes the partial derivative with respect to x_i. [2] Darboux polynomials, which we now define, are the main object of interest in this paper.

Definition 1.1 (Darboux polynomial). Let D denote a polynomial derivation. A polynomial p is *Darboux for* D, or simply *Darboux* when D is clear from the context, whenever $D(p) = qp$ for some polynomial q, called the *cofactor* of p. (Equivalently, p is Darboux if and only if the principal ideal $\langle p \rangle$ is a differential ideal.) Polynomials of total degree zero are trivially Darboux.

Computation of Darboux polynomials is a central problem in the *Prelle-Singer procedure* for computing elementary first integrals of planar systems of polynomial ODEs [12], which yields a systematic method for computing elementary closed-form solutions (whenever these exist) to an important class of ordinary differential equations. Owing to this important application, algorithms for generating Darboux polynomials have received considerable attention in computer algebra. More recently, Darboux polynomials have found application in the area of *formal safety verification* of cyber-physical systems, where the problem of their automatic generation is encountered in the broader context of searching for invariant (and positively invariant) sets [7, 9, 13, 15]. Geometrically, the zero set of a Darboux polynomial defines an invariant set (cf. [11, p. 147]).

Theorem 1.2. *Let $\dot{x} = f(x)$ denote a polynomial ODE and let $x(t)$, $t \in I \subseteq \mathbb{R}$, denote its unique solution for a given initial condition $x(0)$. If p is a Darboux polynomial for the given ODE then the zero set of p, $\{x \mid p(x) = 0\}$, is invariant under the flow of the system, i.e. if $p(x(0)) = 0$ then $p(x(t)) = 0$ for all $t \in I$. (In particular, trivial Darboux polynomials correspond to trivial invariant sets, namely the empty set and the entire space.)*

Remark 1.3. The condition $D(p) \in \langle p \rangle$ is only a sufficient condition for the invariance of the zero set of p; over the complex numbers, when p is square-free, the equivalence holds [3]. Over the reals, however, the radical ideal membership does *not* provide a necessary condition for the invariance of the set of real roots of p and it is instead necessary to consider the real radical ideal [7, Theorem 1].

[1] A modern account of Darboux integrability theory can be found in [8, 17].
[2] D is a special case of the Lie derivative with respect to the vector field defined by f.

ISSAC '24, July 16–19, 2024, Raleigh, NC, USA

ACM ISBN 979-8-4007-0696-7/24/07
https://doi.org/10.1145/3666000.3669705

Darboux generation algorithms (e.g. [1, 7, 10]) are semi-decision procedures enumerating all Darboux polynomials up to a certain fixed bound on the total degree. The bound is eventually increased until finding a (not necessarily irreducible) Darboux polynomial or reaching memory and/or time limits. Theoretically, the existence of a bound on the total degree of irreducible Darboux polynomials is, as of today, an open problem when $n \geq 3$ [8, p. 49, Corollary 2.2]. Even when such theoretical bound exists, it is easily seen that it depends non trivially not only on the total degrees of the polynomials f_i but also on their coefficients. For instance, consider the following planar linear (decoupled) ODE, where $\mu \neq 0$:

$$\dot{x}_1 = \mu x_1, \qquad \dot{x}_2 = x_2 \ . \tag{1}$$

When μ is a positive integer, the polynomial $p = x_1 + x_2^{\mu}$ is an irreducible Darboux polynomial of total degree μ. Any generation procedure is unlikely to succeed in finding p (which involves both x_1 and x_2) unless it reaches μ which can be arbitrarily big. In this work, we precisely tackle this problem: we present a procedure that attempts to make explicit the potential dependencies between the total degree of Darboux polynomials and the polynomials defining the ODE.

Contributions. Given a polynomial derivation D, we devise a procedure to infer necessary conditions for a *generic* polynomial ansatz p to be Darboux. By generic we mean that both the coefficients and the multidegree of p are undetermined (sec. 3). We adapt the standard division algorithm to the specific reduction of $D(p)$ w.r.t. p and show how such polynomials can be encoded and manipulated automatically by a computer program (sec. 4). We discuss the sensitivity of the division to the chosen monomial order and propose a light-weight abstraction based on Newton polytopes to select weight monomial orders that minimize the size of the quotients (sec. 5). Finally, we show how to exploit selected coefficients of the remainder to remove superfluous monomials from p and infer necessary conditions on its multidegree d in a principled way (sec. 6). Throughout the paper, we use the Van der Pol dynamics (example 4.1) as a running example to showcase the proposed algorithms and techniques. In particular, we provide an alternative, fully automated, proof that its limiting cycle is not algebraic for any field (theorem 6.7).

2 PRELIMINARIES

Let $x = (x_1, \ldots, x_n)$ denote a set of variables and $\alpha = (\alpha_1, \ldots, \alpha_n)$ be a vector of natural numbers. We use the shorthand notation x^{α} to denote the multivariate monomial $\prod_{i=1}^{n} x_i^{\alpha_i}$. Given a monomial order, we use the symbol $\prec$ to compare monomials and denote by $\mathrm{LT}(p)$, $\mathrm{LM}(p)$, and $\mathrm{LC}(p)$, the leading term, monomial and coefficient of a polynomial $p = \sum_{\alpha} a_{\alpha} x^{\alpha}$, respectively. When $\mathrm{LC}(p) = 1$, we say that p is *monic*. The exponent $d = (d_1, \ldots, d_n)$ of $\mathrm{LM}(p)$ is called the *multidegree* of p. The coefficients a_{α} of p are assumed to range over some fixed base field of characteristic zero (e.g. $\mathbb{R}$ or $\mathbb{C}$). Given a positive weight vector $w \in \mathbb{N}^n$, the *weight* of a monomial x^{α} is $|\alpha|_w = \sum_i w_i \alpha_i$. In particular, when $w = (1, \ldots, 1)$, the weight coincides with the so-called *degree* of x^{α}, that is $|\alpha| = \sum_i \alpha_i$ (we drop the index w in this case).

The set $\{\alpha \in \mathbb{N}^n \mid a_{\alpha} \neq 0\}$ will be called the *support* or *shape* of p. Unless a_{α} is known to be zero, we consider that α belongs to the support of p. The convex hull of the support of a polynomial is known as its *Newton polytope* [16]. The total weight of p is defined as the maximum of $|\alpha|_w$ when α ranges over the support of p. In particular, $\deg(p)$, the total degree of p, is the maximum of $|\alpha|$ over the support of p. Unlike the univariate case, the equality $\deg(\mathrm{LM}(p)) = \deg(p)$ doesn't hold in general for every monomial order.

The division algorithm [5, Chapter 2] over multivariate polynomials takes as input a polynomial h and an ordered list of polynomials $p_1, \ldots, p_m$ and produces an ordered list of quotients $q_1, \ldots, q_m$ and a remainder, or *normal form*, r with the property that no monomial in r is divisible by $\mathrm{LM}(p_1), \ldots, \mathrm{LM}(p_m)$. In this work, we are interested in the division by only one polynomial p. The dependence of the division algorithm to the monomial order remains, however, even for this simpler case (since the leading term of p depends itself on such order). For instance, consider the polynomial $h = x_1^2 x_2 + x_1 x_2^2 + x_2^2$, and the divisor $p = x_1^2 + x_2^2 - 1$. The reductions below are w.r.t. the degree lexicographic monomial order with $x_1 \succ x_2$ and $x_2 \succ x_1$, denoted $\mathtt{DLex}_{12}$ and $\mathtt{DLex}_{21}$ respectively:

$$h =_{\mathtt{DLex}_{12}} x_2 p + (x_1 x_2^2 - x_2^3 + x_2^2 + x_2)$$
$$h =_{\mathtt{DLex}_{21}} (1 + x_1) p + (x_2 x_1^2 - x_1^3 - x_1^2 + x_1 + 1) \ .$$

Remarkably, when the remainder is zero, the reduction no longer depends on the chosen monomial order. This observation follows from [5, §6, Corollary 2] and the fact that $\{p\}$ is a Gröbner basis of the principal ideal $\langle p \rangle$ for any monomial ordering.[3] As an immediate consequence, we get the following useful facts for Darboux polynomials.

Proposition 2.1. *Let D denote a polynomial derivation. Then p is a Darboux polynomial for D if and only if the remainder of $D(p)$ w.r.t. p is zero for any monomial order. In particular, if q, q' are the quotients of the reduction w.r.t. two distinct monomial orders, then $q = q'$.*

Thus, if one is able to compute the remainder r of $D(p)$ w.r.t. a generic polynomial p, then the equation $r = 0$ gives a necessary and sufficient condition for p to be Darboux. Moreover, if one obtains two quotients for distinct monomial orders, then equating these two quotients leads to necessary conditions for p to be Darboux. While these observations might seem obvious once stated, the former was so far exclusively exploited to search for Darboux polynomials with a fixed total degree and the latter was completely overlooked (cofactors are for instance not used at all in [10]).

3 APPROACH AND INTUITIONS

Given a polynomial derivation D, we shall see in the upcoming sections how to automate (partially or fully) the following steps:

(1) *Encode* a generic ansatz $p = \sum_{\alpha} a_{\alpha} x^{\alpha}$ where the a_{α} as well as the multidegree d of p are symbolic expressions.
(2) Perform the *division* of $D(p)$ w.r.t. p to get a quotient q and a remainder r. (Sec 4)
(3) Exploit "optimal" monomial orders to simplify both q and r for p to be Darboux. (Sec 5)

[3] One doesn't need Gröbner theory to prove such a simple result. A direct proof is provided in appendix A.

(4) Exploit selected equations from the system $r = 0$ to simplify p and get conditions on d, eventually proving the non-existence of nontrivial Darboux polynomials for D. (Sec 6)

Computer algebra systems do not provide built-in capabilities to manipulate polynomials with symbolic multidegrees since even comparing two monomials becomes undecidable in general. In our case, the multidegree of p and therefore its support are not fixed a priori making otherwise straightforward tasks like the reduction challenging. The algorithms presented in the next section aim precisely to overcome these issues.

Furthermore, it is well known that the complexity of the computation (in both time and space) of the polynomial reduction is sensitive to the selected monomial order even when the final result is independent of such order (cf. the discussion at the end of [5, Chapter 2, §9]). It is thus unclear what monomial order to choose a priori and why. This work also presents light-weight heuristics to select *optimal* orders to minimize the size of the support of the quotient q (without performing the division).

Unlike standard generation algorithms, in our case the system $r = 0$ cannot be made explicit. We shall see how to exploit a partial knowledge of this system to infer valuable information on d via constants' propagation. In some cases, this is enough to prove the non-existence of nontrivial Darboux polynomials. In other cases, the system $r = 0$ infers constraints on the multidegree d and one can generate irreducible Darboux polynomials by solving a mixed optimization problem as briefly illustrated below.

To better appreciate the interest and difficulties of the proposed approach, we consider below the (purposely simple) linear dynamic of (1) using $\texttt{DLex}_{21}$ to order the monomials. A generic (monic) ansatz p has then the form $\sum_{P_d(i,j)} a_{i,j} m_{i,j}$ where

$$P_d(i,j) := i, j \geq 0 \wedge (i + j < |d| \vee (i + j = |d| \wedge i \geq d_1)), \quad (2)$$

$\mathrm{LC}(p) = a_{d_1,d_2} = 1$, $m_{i,j} = x_1^i x_2^j$, and $a_{i,j}$ are undetermined elements of the base field. In this case $\deg(p) = |d| = d_1 + d_2$. The general expression for $D(p)$ is

$$D(p) = \mu x_1 \partial_1 p + x_2 \partial_2 p = \sum_{P_d(i,j)} (\mu i + j) a_{i,j} m_{i,j} \ .$$

There is no need to store $D(p)$ entirely to perform the reduction. Only the terms of $D(p)$ in the ideal $\langle \mathrm{LM}(p) \rangle$ are required. For our example, only the leading term of $D(p)$, namely $(\mu d_1 + d_2)\,\mathrm{LM}(p)$, is needed as it is the only term divisible by $\mathrm{LM}(p)$. Thus, we immediately have $q = \mu d_1 + d_2$. The normal form is then $r = D(p) - qp$:

$$\begin{aligned} r &= \sum_{P_d(i,j)} (\mu i + j) a_{i,j} m_{i,j} - (\mu d_1 + d_2) \sum_{P_d(i,j)} a_{i,j} m_{i,j} \\ &= \sum_{P_d(i,j)} (\mu(i - d_1) + j - d_2) a_{i,j} m_{i,j} \ . \end{aligned}$$

To reason about the system $r = 0$, the only expression one needs to store is the coefficient $c_{i,j}$ of the monomial $m_{i,j}$ of r, namely $c_{i,j} = (\mu(i - d_1) + j - d_2) a_{i,j}$. For p to be Darboux, $c_{i,j}$ has to vanish for all i, j. Thus, the only nonzero coefficients of p are those for which the pair (i, j) satisfies the equation $H_d(i, j)$ defined by $\mu(i - d_1) + j - d_2 = 0$. If follows that p is Darboux if and only if its support satisfies the conjunction $P_d(i, j) \wedge H_d(i, j)$. By solving the optimization problem

$$\begin{aligned} \min \quad & |d| \\ \text{s.t.} \quad & P_d(i,j) \wedge H_d(i,j) \end{aligned} \quad (\bigstar)$$

where $i, j, d_1, d_2 \in \mathbb{N}$ and μ is a parameter (of the base field), one gets *irreducible* Darboux polynomials. The irreducibility is a consequence of the following known property of Darboux polynomials [8, Proposition 2.5]: if p is a Darboux polynomial for D, then all factors of p are themselves Darboux polynomials for D. It follows that if the total degree of p is an optimal vector d^* of ($\bigstar$) and p is reducible, then its factors are themselves Darboux with lower total degrees, contradicting the optimality of d^*.

We observe that *all* reduction-based semi-decision procedures for finding Darboux polynomials *cannot* guarantee the irreducibility of the generated polynomials. This limitation is inherent to the way these algorithms operate and cannot be easily bypassed.

It is interesting to observe that for the considered planar case, the multidegree of *any* irreducible Darboux polynomial is an optimal solution of ($\bigstar$). Indeed, if p is an irreducible Darboux polynomial with a non-optimal multidegree d and d^* denotes an optimal multidegree, then there exists an index k with $d^*_k < d_k$ for some index $k \in \{1, 2\}$. One then checks that

$$x_k^{d^*_k - d_k} p$$

is a polynomial contradicting the irreducibility of p. Thus solving ($\bigstar$) provides a principled way to enumerate *all* irreducible Darboux polynomials (detailed in appendix B for convenience). In general, investigating whether each irreducible Darboux polynomial is a solution of a similar optimization problem (completeness) is an interesting research direction that we leave for future work.

4 GENERIC POLYNOMIAL REDUCTION

A straightforward data structure to encode a standard polynomial is a finite set of variables together with a dictionary of (exponent, coefficient) pairs, where each exponent is a vector of natural numbers and the coefficients are allowed to be symbolic expressions (distinct from the expression 0). In such settings, each pair corresponds to a *term*. A generic polynomial p, however, is more subtle to encode as the set of monomials is not fixed and depends on both the multidegree as well as the total degree of p.

Assuming a fixed list of variables $x = (x_1, \ldots, x_n)$, we extend the (exponent, coefficient)-dictionary to encode a generic polynomial $p = \sum_\alpha a_\alpha x^\alpha$ with a symbolic multidegree $d = (d_1, \ldots, d_n)$ as follows. We allow the exponents to be linear expressions with integer coefficients and add a *default* pair $(*, a_*)$ to encode undetermined coefficients. Such a dictionary can be thought of as an n-ary *uninterpreted function* 'a' which is only partially specified by the pairs provided in the dictionary and where each pair corresponds to what we call a *generalized* term. For instance, a monic generic polynomial p with multidegree d is encoded as $p := \{(d, 1), (*, a_*)\}$. We use $p[\alpha]$ (with square brackets) to access the coefficient of the exponent α. For instance, w.r.t. the dictionary above, $p[d]$ evaluates to 1 and $p[\alpha]$ returns the fresh symbol a_α for any α distinct from d. [4]

[4] $p[\alpha]$ should not be confused with $p(\alpha)$ which denotes the standard evaluation of the polynomial p at $x = \alpha$.

We define the functions exponent$(\cdot)$ and coefficient$(\cdot)$ to respectively extract the exponent and coefficient of a generalized term. In the sequel, 't' will be used to denote a (generalized) term and does no longer refer to the time variable.

To make explicit how the coefficients of $D(p)$ and $f_0 p$, for some polynomial f_0, are related to those of p, we decompose the computation as actions of *elementary operators* which we now introduce. An elementary operator has the form $t\partial_i$, $0 \le i \le n$, where t is a term distinct from zero and ∂_i is either the identity (when $i = 0$) or the partial derivative with respect to x_i when $i \ge 1$. Elementary operators act on generalized terms. We have $t\partial_i : g \mapsto t\partial_i g$ and every operator $\delta = t\partial_i$ has a natural right inverse δ^{-1} defined on a (nonzero) term g as the indefinite integral $\int (t^{-1} g) dx_i$.

Algorithm 1 computes the coefficient of a monomial m in $R(p)$ for some operator R defined as a sum of elementary operators. The loop sums up the contributions of all elementary operators composing R. To compute the coefficient of a monomial m in $D(p)$, it suffices to set R to D. If moreover the quotient q of the division of $D(p)$ by p is known, then by setting R to $-q\partial_0 + D$, algorithm 1 computes the coefficient of any monomial of the remainder.

We stress the fact that the algorithm doesn't compute the support of $R(p)$. It only gives the formal expression of the coefficient of a given monomial. Observe also that the algorithm doesn't explicitly check if α belongs to the support of p. Indeed, for a given exponent α, one cannot compare in general the monomials x^α and x^d. Such comparison is however possible when $\alpha - d$ is a vector of integers, and we shall see that this special case occurs frequently in our settings, allowing to effectively simplify the final expressions of the computed coefficients.

Algorithm 2 presents a convenient rewriting of the (standard) division algorithm [5, Theorem 3, p 64] to reduce $D(p)$ w.r.t. p. It highlights the fact that, at each step, a subset of terms is required, namely those divisible by LM(p) (stored in g in line 3). Assuming g is finite and has all its monomials of the form $x^{d+\beta}$, with $\beta \in \mathbb{N}^n$, we observe that computing LT(g') (line 5) becomes a licit operation that can be performed using standard procedures ordering the monomials since g' is a standard polynomial by construction (its exponents are fixed vectors of natural numbers).

Algorithm 3 implements the subroutine required to compute g in line 3 of algorithm 2. We show below that the aforementioned assumptions on g do hold. Line 4 stores in S the monomials $x^\alpha \preceq x^d = \text{LM}(p)$ having their images, via δ, in the monomial ideal $\langle \text{LM}(p) \rangle$ (encoded as exponent$(\delta(x^\alpha)) \ge d$). The system to solve is amenable to an equivalent system of linear inequalities. Using the matrix associated with the selected monomial order [14], the condition $x^\alpha \preceq x^d$ is equivalent to a linear system of inequalities in $d - \alpha$ which we denote by $L(d - \alpha)$. The condition exponent$(\delta(x^\alpha)) \ge d$ is equivalent to $\alpha + \gamma \ge d$ where $\gamma \in \mathbb{Z}^n$ encodes the shift of x^α by the elementary operator δ. By the change of variables $\alpha' = d - \alpha$, the system to solve (line 4) becomes equivalent to $\{\alpha' \mid L(\alpha') \wedge \gamma \ge \alpha'\}$. Any $\alpha \in S$ has thus the form $d - \alpha'$, with $\alpha' \in \mathbb{Z}^n$, and exponent$(\delta(p[\alpha]x^\alpha))$ = exponent$(\delta(x^\alpha))$ = $(d - \alpha') + \gamma = d + (\gamma - \alpha')$ with $\gamma - \alpha' \ge 0$. Letting $\beta = \gamma - \alpha'$, we just proved that all monomials of g (line 5) have the form $d + \beta$, with $\beta \ge 0$.

Algorithm 1: Computation of coeff$_p(R, m)$.

Data: a set of variables x, an operator $R := \sum_{i=0}^n f_i \partial_i$, a generic polynomial $p := \{\ldots, (*, a_*)\}$, a monomial m.
Result: the coefficient of m in $R(p)$.

1 $\Delta \leftarrow \bigcup_{i=0}^n \{t_{i,j}\partial_i \mid f_i = \sum_j t_{i,j}\}$
2 $c \leftarrow 0$
3 **for** $\delta \in \Delta$ **do**
4 $\quad t \leftarrow \delta^{-1}(m)$
5 $\quad \alpha \leftarrow$ exponent(t)
6 $\quad t' \leftarrow \delta(p[\alpha]x^\alpha)$
7 $\quad c \leftarrow c +$ coefficient(t')
8 **return** c

Algorithm 2: Quotient computation.

Data: a set of variables x, a weight monomial order, a polynomial derivation D, a monic generic polynomial $p := \{\ldots, (d, 1), (*, a_*)\}$ with multidegree d.
Result: the quotient q of $D(p)$ w.r.t. p.

1 $q \leftarrow 0$
2 **repeat**
3 $\quad g \leftarrow$ Terms of $(-qp + D(p))$ in $\langle \text{LM}(p) \rangle$
4 $\quad g' \leftarrow \frac{g}{\text{LT}(p)}$
5 $\quad q \leftarrow q + \text{LT}(g')$
6 **until** $g = 0$
7 **return** q

Algorithm 3: Terms of $R(p)$ in $\langle \text{LM}(p) \rangle$.

Data: a set of variables x, a weight monomial order, an operator $R := \sum_{i=0}^n f_i \partial_i$, a generic polynomial $p := \{\ldots, (*, a_*)\}$ with multidegree d.
Result: terms of $R(p)$ in $\langle \text{LM}(p) \rangle$.

1 $\Delta \leftarrow \bigcup_{i=0}^n \{t_{i,j}\partial_i \mid f_i = \sum_j t_{i,j}\}$
2 $g \leftarrow 0$
3 **for** $\delta \in \Delta$ **do**
4 $\quad S \leftarrow \{\alpha \mid x^\alpha \preceq x^d \wedge$ exponent$(\delta(x^\alpha)) \ge d\}$
5 $\quad g \leftarrow g + \sum_{\alpha \in S} \delta(p[\alpha]x^\alpha)$
6 **return** g

Depending on the selected monomial order, the set of solutions S might be infinite. For instance, for $n = 2$, $\delta = x_1\partial_0$, and w.r.t. Lex_{12}, $\alpha = (d_1 - 1, d_2')$ is in S for any $d_2' \ge d_2$. This problem occurs because the stated conditions are not enough to enforce the compactness of the support of p which is assumed only implicitly. To ensure the compactness of the support of p (and therefore S) without adding extra conditions on its total degree, we restrict the computation to weight monomial orders as specified in the input of the algorithm.

Note that the presentation of the different algorithms favors clarity over efficiency. In practice, the computation of g (line 3) is performed incrementally to avoid recomputing the monomials of $D(p)$ in $\langle \text{LM}(p) \rangle$ at each iteration. Likewise, S (line 4) is computed faster using the aforementioned change of variables.

We end this section by performing a generic polynomial reduction for the Van der Pol dynamic with respect to $\mathtt{DLex}_{21}$ (the choice of this particular monomial order will become clearer in section 5).

Example 4.1 (Van der Pol oscillator). The dynamic of the Van der Pol oscillator is defined by the following ODE

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= \mu(1-x_1^2)x_2 - x_1\end{aligned} \tag{3}$$

For simplicity, we fix μ to 1. The corresponding derivation D is then $x_2\partial_1 + ((1-x_1^2)x_2 - x_1)\partial_2$.

Proposition 4.2. *Let $p = \sum_{P_d(i,j)} a_{i,j}x_1^i x_2^j$ be a generic monic polynomial with $\mathrm{LM}(p) = x^d$, $d = (d_1, d_2)$. For the monomial order $\mathtt{DLex}_{21}$, the polynomial reduction of $D(p)$ w.r.t. p is given by $r = -qp + D(p)$ where:*

$$q = -d_2x_1^2 - a_{d_1-2,d_2+1}x_2 + a_{d_1-2,d_2+1}a_{d_1+1,d_2-1}x_1 + q_0, \tag{4}$$

and

$$\begin{aligned}q_0 = a_{d_1+1,d_2-1}&\left(-a_{d_1-2,d_2+1}a_{d_1-1,d_2} + d_1 + 1\right)\\ &+ a_{d_1-2,d_2+1}a_{d_1,d_2-1} + d_2\ .\end{aligned} \tag{5}$$

Moreover, the coefficient $c_{i,j}$ of $x_1^ix_2^j$ in r is given by:

$$\begin{aligned}c_{i,j} = &-(j+1)a_{i-1,j+1} + (j-q_0)a_{i,j} + (i+1)a_{i+1,j-1}\\ &+ (-a_{d_1-2,d_2+1}a_{d_1+1,d_2-1})a_{i-1,j}\\ &+ (a_{d_1-2,d_2+1})a_{i,j-1} + (d_2-j)a_{i-2,j}\ .\end{aligned} \tag{6}$$

Proof. The quotient of the reduction is given by algorithm 2. We compute the coefficient of any term of r using algorithm 1. For convenience, we detail below the several contributions of the involved elementary operators (the first 4 from the operator D and the last 4 from the multiplication by $-q$). With respect to the notations of algorithm 1, we detail coefficient(t'), δ and α:

- $(i+1)a_{i+1,j-1}$ from $x_2\partial_1$ and $(i+1, j-1)$,
- $ja_{i,j}$ from $x_2\partial_2$ and (i, j),
- $-ja_{i-2,j}$ from $-x_1^2x_2\partial_2$ and $(i-2, j)$,
- $-(j+1)a_{i-1,j+1}$ from $-x_1\partial_2$ and $(i-1, j+1)$,
- $d_2a_{i-2,j}$ from $d_2x_1^2\partial_0$ and $(i-2, j)$,
- $a_{d_1-2,d_2+1}a_{i,j-1}$ from $a_{d_1-2,d_2+1}x_2\partial_0$ and $(i, j-1)$,
- $-a_{d_1-2,d_2+1}a_{d_1+1,d_2-1}a_{i-1,j}$ from $-a_{d_1-2,d_2+1}a_{d_1+1,d_2-1}x_1\partial_0$ and $(i-1, j)$,
- $-q_0a_{i,j}$ from $-q_0\partial_0$ and (i, j) .

It suffices to sum up these contributions to get the stated $c_{i,j}$. □

5 OPTIMAL WEIGHT ORDERS

For non-zero polynomials p and q, one has $\mathrm{LM}(pq) = \mathrm{LM}(p)\,\mathrm{LM}(q)$ and $\mathrm{LM}(p+q) \preceq \max\{\mathrm{LM}(p), \mathrm{LM}(q)\}$ (for any monomial order) where the equality holds whenever $\mathrm{LM}(p) = \mathrm{LM}(q) \implies \mathrm{LC}(p) + \mathrm{LC}(q) \neq 0$. In particular, when r is the normal form of h w.r.t. p, one has $h = qp + r$ and $\mathrm{LM}(h) = \max\{\mathrm{LM}(q)\,\mathrm{LM}(p), \mathrm{LM}(r)\}$ (since $\mathrm{LM}(r) \neq \mathrm{LM}(q)\,\mathrm{LM}(p)$ by definition of r). Finally, if the monomial m divides the monomial m' then $m \preceq m'$ for any monomial order. [5]

The quotient of the polynomial reduction of $D(p)$ w.r.t. p depends on the chosen monomial order. However, proposition 2.1 tells us that for p to be Darboux, all quotients must be equal. Thus computing the quotients for distinct monomial orders and comparing their supports may yield interesting necessary conditions for p to be Darboux. In addition, selecting monomial orders that minimize the size of the support of q is a reasonable parameter to control: monomial orders which lead to larger supports would necessarily have superfluous monomials that would only complicate the expressions of q and the coefficients of r. Indeed, this latter fact can be appreciated in algorithm 2 where q is constructed term by term.

In this section, we present a light-weight abstraction that selects weight orders that minimize the size of q *without computing* q. The idea is to find an upper bound on $\mathrm{LM}(q)$ that depends only on the derivation D and which is valid for all monomial orders.

For a derivation $D = \sum_{i=1}^n f_i\partial_i$, and a monomial x^α, $\alpha > 0$, one has

$$D(x^\alpha) = \sum_{i=1}^n \alpha_i f_i \tfrac{x^\alpha}{x_i} = x^{\alpha-1}\sum_{i=1}^n \alpha_i f_i \tfrac{x}{x_i} = x^{\alpha-1}s_\alpha\ . \tag{7}$$

Thus the action of D on x^α (when $\alpha > 0$) is the same as the multiplication of $x^{\alpha-1}$ by the polynomial s_α. When there exists an index j such that $\alpha_j = 0$ then $\partial_j x^\alpha = 0$. We let I denote the set of indices such that $\alpha_i > 0$ for all $i \in I$ and let ${x^\alpha}_{|I}$ denote the restriction of the monomial x^α to the indices $i \in I$. We then have

$$D({x^\alpha}_{|I}) = \sum_{i\in I} \alpha_i f_i \tfrac{x^\alpha}{x_i} = {x^{\alpha-1}}_{|I}\sum_{i\in I} \alpha_i f_i \tfrac{x_{|I}}{x_i} = {x^{\alpha-1}}_{|I}\, s_{\alpha|I}, \tag{8}$$

and therefore the action of D is also the same as the multiplication of the monomial ${x^{\alpha-1}}_{|I}$ by the polynomial $s_{\alpha|I}$.

To remove the dependency of the polynomials $s_{\alpha|I}$ to α, we abstract away the actual coefficients and keep only the support of the involved terms. To do so, we introduce the formal polynomial $s_{|I}$ having as its support the Newton polytope of all terms in $f_i\frac{x_{|I}}{x_i}$ for all $i \in I$. (There is no harm in regarding $s_{|I}$ as both a formal polynomial or its corresponding Newton polytope as long as the type is clear from the context.) Thus, for any α, and any I, the support of $s_{\alpha|I}$ is a subset of the support of $s_{|I}$. In particular $\mathrm{LM}(s_{\alpha|I}) \preceq \mathrm{LM}(s_{|I})$ for any monomial order. The polynomials $s_{|I}$ depend therefore only on the derivation D and the set I. The following lemma alleviates further the need to enumerate the 2^n-1 polynomials $s_{|I}$ as the leader of the polynomial $s = s_{|\{1,\ldots,n\}}$ provides a tight upper bound. We term s, when seen as a formal polynomial, the *Newton polynomial* of D.

Lemma 5.1. *Let D be a polynomial derivation and let s denote its Newton polynomial. Let $s_{\alpha|I}$ be defined as in equation (8). Then for any monomial order, any non-empty subset $I \subseteq \{1,\ldots,n\}$, and any exponent α, $\mathrm{LM}(s_{\alpha|I}) \preceq \mathrm{LM}(s)$.*

Proof. Let $I \subseteq \{1,\ldots,n\}$ denote a non-empty subset. By definition of the Newton polynomial $s_{|I}$, for any α, and any monomial order, $\mathrm{LM}(s_{\alpha|I}) \preceq \mathrm{LM}(s_{|I})$. Moreover, any monomial m in $s_{|I}$ has the form $t\frac{x_{|I}}{x_i}$ where t denotes some monomial in f_i, $i \in I$. But $m' = t\frac{x}{x_i}$ is also a monomial of s which is divisible by m. By lemma A.2, $m \preceq m'$. When $m = \mathrm{LM}(s_{|I})$, one gets $\mathrm{LM}(s_{|I}) \preceq m' \preceq \mathrm{LM}(s)$ as stated. □

[5] This fact isn't immediate from the definition of monomial orders. Cf. lemma A.2 in appendix A for a discussion and a direct proof.

Proposition 5.2. *Let D be a polynomial derivation and let s denote its Newton polynomial. Let q denote the quotient of the division of $D(p)$ by p for some monomial order. Then $x\,\mathrm{LM}(q) \preceq \mathrm{LM}(s)$.*

Proof. We decompose p and $D(p)$ over non-empty subsets I:

$$p = a_0 + \sum_I \sum_{\substack{\forall j\notin I\\ \alpha_j=0}} a_\alpha x^\alpha = a_0 + \sum_I \sum_{\substack{\forall j\notin I\\ \alpha_j=0}} a_\alpha x^\alpha{}_{|I} \ .$$

$$D(p) = \sum_I \sum_{\substack{\forall j\notin I\\ \alpha_j=0}} a_\alpha D(x^\alpha{}_{|I}) = \sum_I \sum_{\substack{\forall j\notin I\\ \alpha_j=0}} a_\alpha x^{\alpha-1}{}_{|I}\, s_{\alpha|I} \ .$$

For any $\alpha \in \mathrm{support}(p)$, we have by lemma 5.1

$$x_{|I}\,\mathrm{LM}(x^{\alpha-1}{}_{|I} s_{\alpha|I}) = x^\alpha{}_{|I}\,\mathrm{LM}(s_{\alpha|I}) \preceq \mathrm{LM}(p)\,\mathrm{LM}(s)\,.$$

Moreover

$$\mathrm{LM}(D(p)) \preceq \max_I\{x^{\alpha(I)-1}{}_{|I}\,\mathrm{LM}(s_{|I})\},$$

where $x^{\alpha(I)-1}{}_{|I}$ denotes the leading monomial of $\sum_{\substack{\forall j\notin I\\ \alpha_j=0}} a_\alpha x^{\alpha-1}{}_{|I}$.

Thus

$$x_{|I}\,\mathrm{LM}(D(p)) \preceq x_{|I} \max_I\{x^{\alpha(I)-1}{}_{|I}\,\mathrm{LM}(s_{|I})\} \preceq \mathrm{LM}(p)\,\mathrm{LM}(s) \ . \tag{9}$$

By lemma A.2, $x_{|I} \preceq x$ for any non-empty subset I. Thus $x = \max_I\{x_{|I}\}$ and

$$x\,\mathrm{LM}(D(p)) = \max_I\{x_{|I}\}\,\mathrm{LM}(D(p)) = \max_I\{x_{|I}\,\mathrm{LM}(D(p))\} \ .$$

By eq. (9), $x\,\mathrm{LM}(D(p)) \preceq \mathrm{LM}(p)\,\mathrm{LM}(s)$. However $\mathrm{LM}(p)\,\mathrm{LM}(q) \preceq \mathrm{LM}(D(p))$, hence $x\,\mathrm{LM}(p)\,\mathrm{LM}(q) \preceq \mathrm{LM}(p)\,\mathrm{LM}(s)$, and $x\,\mathrm{LM}(q) \preceq \mathrm{LM}(s)$ as desired. □

Remark 5.3. A simple degree-based analysis of the derivation $D = \sum_{i=1}^n f_i\partial_i$, shows that $\deg(q) \le -1 + \max_i \deg(f_i)$. This inequality is also an immediate corollary of proposition 5.2 since $\deg(s) \le n - 1 + \max_i \deg(f_i)$ uniformly for all monomial orders. While uniform upper bounds tend to be appreciated, in our case, the dependency to the monomial order is instrumental as it would potentially lead to a mismatch on the supports of the quotients q giving relevant necessary conditions for p to be Darboux. Moreover, the upper bound $\mathrm{LM}(s)$ is in general tighter to estimate the monomials in q. For instance, for the Van der Pol dynamics, the degree-based analysis would infer that $\deg(q) \le 2$, giving an estimate of 6 monomials for q for all monomial orders. For $\mathtt{DLex}_{21}$, $x\,\mathrm{LM}(q) \preceq \mathrm{LM}(s)$ gives only 4 monomials and the upper bound decreases even to 3 for other weight orders.

Let's consider example 4.1 and recall its derivation $D = x_2\partial_1 + ((1-x_1^2)x_2 - x_1)\partial_2$. In this case I can be either $\{1\}$, $\{2\}$ or $\{1,2\}$:

$$\begin{aligned} s_{\alpha|\{1\}} &= \alpha_1 f_1 \tfrac{x_1}{x_1} = \alpha_1 x_2 \\ s_{\alpha|\{2\}} &= \alpha_2 f_2 \tfrac{x_2}{x_2} = \alpha_2(-x_1^2x_2 + x_2 - x_1) \\ s = s_{\alpha|\{1,2\}} &= \alpha_1 f_1 \tfrac{x}{x_1} + \alpha_2 f_2 \tfrac{x}{x_2} \\ &= -\alpha_2 x_1^3 x_2 + \alpha_1 x_2^2 + \alpha_2 x_1 x_2 - \alpha_2 x_1^2 \ . \end{aligned}$$

Their Newton polytopes are depicted in fig. 1. Each dot corresponds to a monomial $t\frac{x_{|I}}{x_i}$ in $s_{\alpha|I}$. One can appreciate the fact that Newton polytopes are over-approximations of the actual supports (for instance $x_1^2x_2$ is in s but not in s_α for any α).

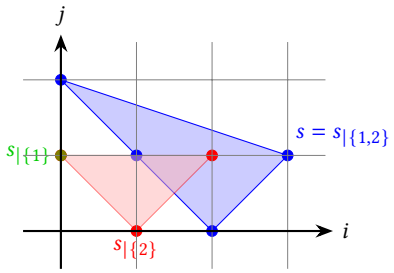


Figure 1: Newton polytopes (Van der Pol).

Proposition 5.2 provides a key abstraction to reason about q without computing it. We exploit such abstraction below to select weight monomial orders that minimize the size of q. Let $|m|_{\mathtt{mord}}$ denote the number of monomials less than m for a given monomial order $\mathtt{mord}$. We drop the index $\mathtt{mord}$ when it's clear from the context. Proposition 5.2 implies that $\mathrm{LM}(q) \prec \mathrm{LM}(s)$ and therefore $|\mathrm{LM}(q)| < |\mathrm{LM}(s)|$ for any monomial order. To minimize $|\mathrm{LM}(q)|$ over monomial orders, it then suffices to minimize $|\mathrm{LM}(s)|$ (which depends only on D).

Let $\mathrm{LM}(s) = x^\beta$ for some $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$. We restrict ourselves to weight orders $w\mathtt{Lex}$ where $w \in \mathbb{N}^n$ is a positive weight vector, and seek to minimize $|\mathrm{LM}(s)|$ over $w\mathtt{Lex}_\sigma$ where σ ranges over the $n!$ permutations of the variables. ($\mathtt{DLex}$ are particular cases with $w = (1, \ldots, 1)$.)

Proposition 5.4. *Let s denote the Newton polynomial for a polynomial derivation D. Fix a positive weight vector $w \in \mathbb{N}^n$ and let $w\mathtt{Lex}_\sigma$ denote the weight monomial order with $x_{\sigma(1)} \succ \cdots \succ x_{\sigma(n)}$. Assume that the leading term x^β of s is the same for all σ. Let σ^* denote a permutation such that $w_{\sigma^*(1)}\beta_{\sigma^*(1)} \le \cdots \le w_{\sigma^*(n)}\beta_{\sigma^*(n)}$. Then the monomial order $w\mathtt{Lex}_{\sigma^*}$ minimizes $|\mathrm{LM}(s)|$ over $w\mathtt{Lex}_\sigma$.*

Proof. For $w\mathtt{Lex}_\sigma$, if $x^\alpha \preceq x^\beta$ then $|\alpha|_w \le |\beta|_w$ (the weight of a monomial is defined at beginning of section 2). The size of the set $\{x^\alpha \mid |\alpha|_w < |\beta|_w\}$ is independent from the permutation σ. It thus suffices to minimize the size of $\{x^\alpha \mid x^\alpha \preceq x^\beta \wedge |\alpha|_w = |\beta|_w\}$ when σ ranges over the permutations. The stated result is then an immediate corollary of proposition C.3 (cf. appendix C). □

For the running example and $\mathtt{DLex}_\sigma$ orders, applying proposition 5.2 to s (shown in fig. 1), one gets $x_1x_2\,\mathrm{LM}(q) \preceq \mathrm{LM}(s) = x_1^3x_2 = x^\beta$ (for all $\mathtt{DLex}_\sigma$ orders). By proposition 5.4, $\mathtt{DLex}_{21}$ minimizes the size of q over all $\mathtt{DLex}_\sigma$ orders since $\beta_2 = 1 \le 3 = \beta_1$. The reason why we computed q and $c_{i,j}$ w.r.t. $\mathtt{DLex}_{21}$ at the end of section 4 is now justified.

Remark 5.5. For a fixed weight vector w, the hypothesis in proposition 5.4 requiring that the leading monomial of s should be the same for $w\mathtt{Lex}_\sigma$ for all σ is not really a limitation. It suffices to perturb slightly the slope defined by w to make the requirement holds for some other w'. For instance, the weight vector $w = (1,3)$ doesn't satisfy the requirement for the Newton polytope s in fig. 1. It suffices then to consider $w' = (1,4)$ or $w' = (1,1)$.

Our strategy is to consider optimal weight monomial orders suggested by the vertices of the Newton polytope s and to compute

the quotients for such orders using algorithm 2. By equating the quotients, one thus gets necessary conditions on the coefficients of p for p to be Darboux.

We further observe that proposition 5.2 can be used to show that, for p to be Darboux, the support of q is tightly over-approximated by the Newton polytope s shifted by $(-1, \ldots, -1)$. Remarkably, a similar result was shown in [4, Proposition 6] where it was used to provide sufficient criteria for the existence of rational first integrals.

We conclude by applying our strategy for the running example 4.1. Vertices $(0, 2)$ and $(3, 1)$ of the Newton polytope s correspond to leading monomials for the respective weight orders $w = (1, 4)$ and $w = (1, 1)$. Proposition 5.4 applies in both cases and suggests the monomial orders $w\mathtt{Lex}_{12}$ and $\mathtt{DLex}_{21}$. We already computed the quotient for $\mathtt{DLex}_{21}$. For $w\mathtt{Lex}_{12}$, algorithm 2 gives the following expression where the coefficients g_i denote some expressions that we don't explicit for conciseness:

$$q_w = g_2 x_1^2 + g_1 x_1 + g_0 \ . \tag{10}$$

The supports mismatch between the quotients obtained for $\mathtt{DLex}_{21}$ and $w\mathtt{Lex}_{12}$ provides the following necessary condition.

Proposition 5.6. *For the Van der Pol dynamic, for p to be Darboux, it is necessary that $a_{d_1-2,d_2+1} = 0$. Therefore*

$$q = -d_2 x_1^2 + d_2 + (1 + d_1) a_{d_1+1,d_2-1} = -d_2 x_1^2 + q_0' \ .$$

Proof. For p to be Darboux, the expression of q in eq. (4) and q_w in eq. (10) must be equal. The supports of q and q_w differ by x_2 which is present in q but not in q_w. Thus the coefficient of x_2 in q has to vanish, that is $a_{d_1-2,d_2+1} = 0$. This in turn simplifies the constant q_0 in eq. (5) to $q_0' = d_2 + (1 + d_1) a_{d_1+1,d_2-1}$. □

As an immediate consequence, the expression of $c_{i,j}$ of eq. (6) simplifies to

$$\begin{aligned} c_{i,j} = &-(j+1) a_{i-1,j+1} + (j - q_0') a_{i,j} \\ &+ (i+1) a_{i+1,j-1} + (d_2 - j) a_{i-2,j} \ . \end{aligned} \tag{11}$$

Fig. 2 is convenient to appreciate which coefficients of p contribute to $c_{i,j}$ where an arrow $(i', j') \to (i, j)$ intuitively means that $a_{i',j'}$ contributes to $c_{i,j}$. The dependency to a_{d_1+1,d_2-1} (which appears in q_0') is omitted as this particular coefficient behaves like a constant with respect to the varying coefficients which depend on the selected (i, j).

6 CONSTANTS' PROPAGATION

For the generic polynomial p to be Darboux, the remainder r has to identically vanish. In our settings, r cannot be made explicit and is only accessible via querying its coefficients. The idea is select

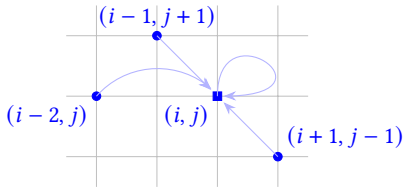


Figure 2: Contributions to $c_{i,j}$.

simple coefficients of r to infer additional necessary conditions on the support of p and its multidegree. One could for instance select coefficients of r that involve a unique undetermined coefficient a_α of p. Once a_α is set, the information is propagated to the entire system. The process is re-iterated until no such simplifications are possible. If one detects a contradiction along the way then a proof of non-existence (of nontrivial Darboux polynomials) is provided.

If the propagation stops without arriving at a contradiction, then the user may want to provide additional directives like trying a different monomial order or supplying additional assumptions. In general, the presented method is not guaranteed to arrive at a contradiction if a Darboux polynomial doesn't exist. We observe that the gathered necessary conditions could complement well, at least in principle, standard generation algorithms by reducing upfront the support of p for a fixed multidegree d. Indeed, solving $r = 0$ is arguably the main computational bottleneck for these procedures.

We end this section by showing the successive steps of the constant propagation on our running example. Fig. 3a shows the initial support of p in red w.r.t. $\mathtt{DLex}_{21}$. The exponent $(d_1 - 2, d_2 + 1)$ was removed by proposition 5.6. The positions of the blue patterns illustrate how we target a coefficient of p using the generic expression of $c_{i,j}$ (cf. fig. 2). Notice that when the blue pattern doesn't overlap with p, it simply means that (i, j) is outside the support of the remainder r (this gives a hint about the support of r without computing it). By "sliding" the (upper) blue pattern of $c_{i,j}$ along the upper red diagonal of fig. 3a, all the coefficients of that diagonal are removed. Formally:

Lemma 6.1. *For all $0 \le \ell \le d_1 - 3$,*

$$c_{\ell+2,d_1+d_2-\ell-1} = 0 \implies a_{\ell,d_1+d_2-\ell-1} = 0 \ .$$

Proof. When $\ell \le d_1 - 3$, for $(i, j) = (\ell + 2, d_1 + d_2 - \ell - 1)$, there is a unique contributor to $c_{i,j}$ (cf. eq. (11)), namely $(d_2 - j) a_{i-2,j}$. Substituting i, j, one gets $-(d_1 - \ell - 1) a_{\ell,d_1+d_2-\ell-1} = 0$. The result follows since $\ell \le d_1 - 3$ implies $-2 \ge -(d_1 - \ell - 1)$. □

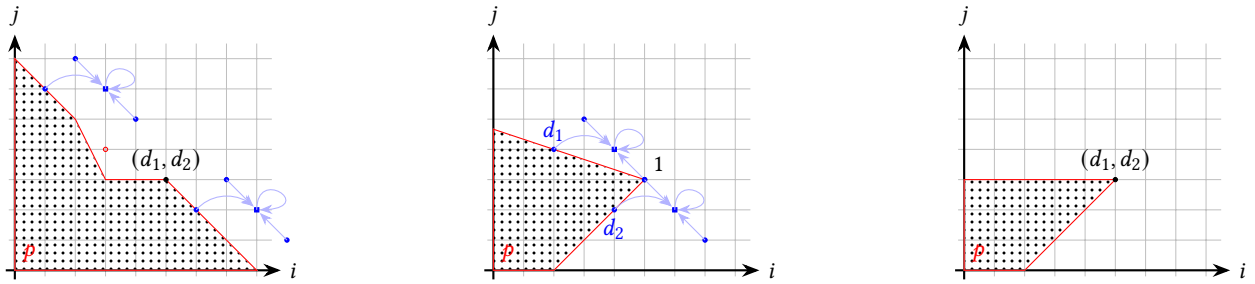
The same simplification holds by repeatedly sliding the blue pattern along the successive upper diagonals of p (for $j > d_2$). The shape of the polynomial p is "trimmed" from its original triangular form to a staircase of slope $-\frac{1}{3}$, as shown in fig. 3b. The same reasoning actually holds for the lower diagonals.

Lemma 6.2. *For $1 \le \ell \le d_2, c_{d_1+\ell+2,d_2-\ell} = 0 \implies a_{d_1+\ell,d_2-\ell} = 0$.*

Proof. Apply eq. (11) with $(i, j) = (d_1 + \ell + 2, d_2 - \ell)$. There is a unique contributor to $c_{i,j}$, namely $(d_2 - j) a_{i-2,j}$. One thus gets $c_{i,j} = -((d_2 - \ell) - d_2) a_{d_1+\ell,d_2-\ell} = \ell a_{d_1+\ell,d_2-\ell}$. (When $\ell = 0$, the equation is trivial and doesn't imply any additional constraint on $a_{d_1,d_2} = \mathrm{LC}(p) = 1$.) □

Likewise, by sliding the pattern along the successive lower diagonals, the support of p gets further simplified to the one shown in fig. 3b. Observe how the final shape of p coincides with the two left slopes of the blue pattern used to trim p. The propagation of zeros is achieved and the predicate $P_d(i, j)$ is updated to

$$P_d'(i, j) := i, j \ge 0 \wedge (i + 3j \le d_1 + 3d_2) \wedge (i - j \le d_1 - d_2) \ . \tag{12}$$



(a) Initial support (DLex). (b) Diagonals trimmed (DLex). (c) Canonical shape.

Figure 3: Successive (truncated) supports of p after constants propagation for example 4.1.

Remark 6.3 (Canonical shape). The shape of p depicted in fig. 3b can be simplified further by considering a weight order with $w = (1, 42)$ say. Theoretically, the weight vector $w = (1, d_1 + 1)$ would remove all the monomials with $j > d_2$ leading to the reduced (asymptotic) shape of p of fig. 3c. This particular shape is *canonical* in the sense that it doesn't depend on any monomial order: all monomials of p divides $\mathrm{LM}(p)$ and are therefore lower than $\mathrm{LM}(p)$ for any monomial order by lemma A.2.

We further propagate one more constant, namely $\mathrm{LC}(p) = 1$. For $\mathtt{DLex}_{21}$, this is achieved by positioning the blue patterns as shown in fig. 3b where the coefficients appear as labels.

Lemma 6.4. *Assume the predicate $P'_d(i, j)$ for p. Then $c_{d_1-1,d_2+1} = 0 \implies a_{d_1-3,d_2+1} = d_1$.*

Proof. Apply eq. (11) with $(i, j) = (d_1 - 1, d_2 + 1)$. We only need to account for 2 monomials. Thus $c_{i,j} = (i+1)a_{i+1,j-1}+(d_2-j)a_{i-2,j}$. Since $\mathrm{LC}(p) = 1$, we get $c_{d_1-1,d_2+1} = d_1 - a_{d_1-3,d_2+1}$. □

Lemma 6.5. *Assume the predicate $P'_d(i, j)$ for p. Then $c_{d_1+1,d_2-1} = 0 \implies a_{d_1-1,d_2-1} = d_2$.*

Proof. Apply eq. (11) with $(i, j) = (d_1 + 1, d_2 - 1)$. We only need to account for 2 monomials. Thus $c_{i,j} = -(j + 1)a_{i-1,j+1} + (d_2 - j)a_{i-2,j}$. Since $\mathrm{LC}(p) = 1$, we get $c_{d_1+1,d_2-1} = -d_2 + a_{d_1-1,d_2-1}$. □

The propagation of d_1, d_2 can now carry on (upwards) along the boundaries of p prescribed by $P'_{i,j}$ until reaching the axes ($i = 0$ and $j = 0$) at which point, some conditions on d_1, d_2 will have to be satisfied. Instead of doing so, we present in the remaining of this section a "shortcut" that leads to the same result. The idea is to explore the coefficients in the vicinity of $\mathrm{LM}(p)$ looking for a *local* contradiction that do not require reaching the axes. Such strategy is appealing as it attempts to minimize the size of the proof.

Assuming $P'_d(i, j)$, the constant q'_0 in proposition 5.6 simplifies to d_2. Thus, when $j = d_2$, eq. (11) simplifies to

$$c_{i,d_2} = -(d_2 + 1)a_{i-1,d_2+1} + (i + 1)a_{i+1,d_2-1} \quad . \tag{13}$$

Pictorially, the corresponding blue pattern gets reduced to its only 2 diagonal dots when it slides along $j = d_2$. In particular, when $(i, j) = (d_1 - 2, d_2)$, on gets the following constraint on d.

Proposition 6.6. *For p to be Darboux, $|d| = 0$ must hold.*

Proof. Fix the monomial order to $\mathtt{DLex}_{21}$. For p to be Darboux, its shape is prescribed by $P'_d(i, j)$. Thus, for $(i, j) = (d_1 - 2, d_2)$, eq. (13) applies and one gets $c_{i,j} = c_{d_1-2,d_2} = -(d_2 + 1)a_{d_1-3,d_2+1} + (d_1 - 1)a_{d_1-1,d_2-1}$. However $a_{d_1-3,d_2+1} = d_1$ by lemma 6.4 and $a_{d_1-1,d_2-1} = d_2$ by lemma 6.5. Thus $c_{i,j} = -(d_2+1)d_1+(d_1-1)d_2 = -d_1 - d_2$. The result follows as $c_{i,j}$ has to vanish. □

As a byproduct, we give a new proof for the following known result [11] where it is stated for complex numbers in terms of invariant algebraic curves (cf. remark 1.3):

Theorem 6.7. *Assuming $\mu \neq 0$, the Van der Pol oscillator (example 4.1) has no nontrivial Darboux polynomials over any field. In particular, its limit cycle is not an algebraic curve.*

Notice that the entire trimming of the shape of p is not necessary to prove theorem 6.7. In fact only finitely many coefficients a_α, all in the vicinity of the leader $\mathrm{LM}(p)$, need to be assigned, namely

$$a_{d_1-2,d_2+1}, a_{d_1+1,d_2-1}, a_{d_1,d_2-1}, a_{d_1-3,d_2+1}, a_{d_1-1,d_2-1}, a_{d_1-2,d_2} \, .$$

This finite set provides a concise *formal certificate* that can be checked independently to verify the claimed result.

We implemented our algorithms as a Wolfram Mathematica package [2] and were able to automatically prove that no nontrivial Darboux polynomial exists for the entire class of Liénard systems as stated in [11] (the Van der Pol oscillator being in particular a typical Liénard system).

7 CONCLUSION

The existence of an upper bound on irreducible Darboux polynomials in higher dimensions is conjectured but remains out of reach. We currently even lack sufficient or necessary algebraic criteria for the existence of Darboux polynomials, except for simple restricted classes of derivations. A key difficulty resides in the highly intricate relationships that the polynomials defining the derivation have to satisfy. We believe that the toolbox provided in this work is an important step forward to approach the problem *experimentally* with the assistance of a computer program, avoiding thereby the cumbersome error-prone pen-and-paper computations and focusing on potentially interesting patterns that could serve to sharpen our intuitions on these challenging problems.

ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers for their thorough reading and very relevant feedback. The first author is in debt to Andrew Sogokon for fruitful and very informative discussions about the state-of-the-art of the generation algorithms for Darboux polynomials as well as his comments and observations on earlier drafts of this work.

REFERENCES

[1] Alin Bostan, Guillaume Chèze, Thomas Cluzeau, and Jacques-Arthur Weil. 2016. Efficient algorithms for computing rational first integrals and Darboux polynomials of planar polynomial vector fields. *Math. Comp.* 85, 299 (2016), 1393–1425. https://doi.org/10.1090/mcom/3007

[2] Maxime Bridoux and Khalil Ghorbal. 2024. A Mathematica Package for Certifying the Nonexistence of Darboux Polynomials. In *ISSAC (Software presentation)*. ACM Press, New York City, NY, USA.

[3] Colin J. Christopher. 1994. Invariant algebraic curves and conditions for a centre. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 124, 6 (1994), 1209–1229. https://doi.org/10.1017/S0308210500030213

[4] Guillaume Chèze. 2014. Darboux theory of integrability in the sparse case. *Journal of Differential Equations* 257, 2 (2014), 601–609. https://doi.org/10.1016/j.jde.2014.04.012

[5] David Cox, John Little, and Donal O'Shea. 2007. *Ideals, Varieties, and Algorithms.* Springer, New York. https://doi.org/10.1007/978-0-387-35651-8

[6] Gaston Darboux. 1878. Mémoire sur les équations différentielles algébriques du premier ordre et du premier degré (in French). *Bulletin des Sciences Mathématiques et Astronomiques* 2, 1 (1878), 60–96. http://eudml.org/doc/85010

[7] Khalil Ghorbal and André Platzer. 2014. Characterizing Algebraic Invariants by Differential Radical Invariants. In *TACAS*. Springer, Berlin, Heidelberg, 279–294. https://doi.org/10.1007/978-3-642-54862-8_19

[8] Alain Goriely. 2001. *Integrability and Nonintegrability of Dynamical Systems.* World Scientific, Singapore. https://doi.org/10.1142/3846

[9] Eric Goubault, Jacques-Henri Jourdan, Sylvie Putot, and Sriram Sankaranarayanan. 2014. Finding non-polynomial positive invariants and Lyapunov functions for polynomial systems through Darboux polynomials. In *ACC*. IEEE, Portland, OR, USA, 3571–3578. https://doi.org/10.1109/ACC.2014.6859330

[10] Yiu-Kwong Man. 1993. Computing Closed Form Solutions of First Order ODEs Using the Prelle-Singer Procedure. *J. Symb. Comput.* 16, 5 (1993), 423–443. https://doi.org/10.1006/jsco.1993.1057

[11] Kenzi Odani. 1995. The Limit Cycle of the van der Pol Equation Is Not Algebraic. *Journal of Differential Equations* 115, 1 (jan 1995), 146–152. https://doi.org/10.1006/jdeq.1995.1008

[12] Myra Jean Prelle and Michael F Singer. 1983. Elementary first integrals of differential equations. *Trans. Amer. Math. Soc.* 279, 1 (1983), 215–229.

[13] Rachid Rebiha, Arnaldo Vieira Moura, and Nadir Matringe. 2015. Generating invariants for non-linear hybrid systems. *Theor. Comput. Sci.* 594 (2015), 180–200. https://doi.org/10.1016/j.tcs.2015.06.018

[14] Lorenzo Robbiano. 1985. Term Orderings on the Polynominal Ring. In *EUROCAL*, Vol. 204. Springer, Berlin, Heidelberg, 513–517. https://doi.org/10.1007/3-540-15984-3_321

[15] Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. 2008. Constructing invariants for hybrid systems. *FMSD* 32, 1 (2008), 25–55. https://doi.org/10.1007/s10703-007-0046-1

[16] Bernd Sturmfels. 1996. *Grobner bases and convex polytopes.* American Mathematical Society, RI, USA.

[17] Xiang Zhang. 2017. *Integrability of Dynamical Systems: Algebra and Analysis.* Developments in Mathematics, Vol. 47. Springer, Singapore. https://doi.org/10.1007/978-981-10-4226-3

A SIMPLE PROOFS

Proposition 2.1 holds for any polynomial h, not just $D(p)$. A simple direct proof is provided below without requiring Gröbner theory.

Lemma A.1. *Let h and p be two polynomials over a field. Then h is in the ideal generated by p if and only if the remainder of the division of h by p is zero with respect to any monomial order.*

Proof. Necessity. Suppose that the ideal membership assumption holds. Then, there exists a polynomial q such that $h = qp$. Suppose there exists a monomial ordering such that $h = q'p + r$ and $r \neq 0$. Then $r = (q - q')p$ forcing $q \neq q'$. But then $\mathrm{LM}(p)$ divides $\mathrm{LM}(r)$ contradicting the definition of r. (Sufficiency is trivial.) □

The following useful fact is not immediate from the definition of monomial orders. It can however be shown using Dickson lemma as stated in [5, Corollary 6, p72]. We provide below a direct proof.

Lemma A.2. *Let m, m' denote two monomials. If m divides m' then $m \leq m'$ for any monomial order. (The converse doesn't hold in general.)*

Proof. The result is immediate when $m' = m$. Suppose that $m' \leq m$. One has $m' = qm$ for some monomial q. Suppose there exists a monomial ordering for which $m \succ m'$ then $m \succ qm$ and one constructs a descending sequence of monomials $m \succ qm \succ q^2m \succ \ldots$ which must terminate (by the well-foundedness of monomial orders). Thus there exists a finite index k such that $q^k m = q^{k+1}m$. But then $m = qm = m'$, a contradiction. To see that the converse doesn't hold in general, consider $\mathtt{DLex}_{21}$. Then, $x_1 \prec x_2$ but x_1 doesn't divide x_2. □

B PLANAR LINEAR DYNAMICS

If μ is irrational then $P_d(i,j) \wedge H_d(i,j)$ reduces to the singleton (d_1, d_2) and the only Darboux polynomial is $\mathrm{LM}(p) = x^d$. This in particular means that x_1 and x_2 are Darboux polynomials since factors of a Darboux polynomial are themselves Darboux polynomials.

Assume next that μ is rational. By substituting j for $d_2 - \mu(i - d_1)$ using $H_d(i,j)$, the conjunction $P_d(i,j) \wedge H_d(i,j)$ reduces to the following predicate on i:

$$P_d(i) := \big((1-\mu)(i-d_1) < 0 \wedge \mu(i-d_1) \leq d_2 \wedge i \geq 0\big) \tag{14}$$

$$\vee \ \big((1-\mu)(i-d_1) = 0 \wedge \mu(i-d_1) \leq d_2 \wedge i \geq d_1\big) \tag{15}$$

which involves the parameter μ as well as the multidegree $d = (d_1, d_2)$ of p.

If $\mu = 1$ then $P_d(i)$ reduces to $d_1 \leq i \leq |d|$, $j = |d| - i$. The optimal value of $|d|$ is d_2 reached for $d_1 = 0$. Thus any homogeneous polynomial is a Darboux polynomial.

If $\mu \neq 1$, eq. (15) gives $i = d_1$, and $(i,j) = (d_1, d_2)$. So $\mathrm{LM}(p)$ is the unique monomial of degree $|d|$ in p.

Using eq. (14), the smallest $|d|$ for which a Darboux polynomial involving both x_1 and x_2 exists satisfies $d_2 = \max\{0, \mu(i - d_1)\}$.

When $\mu(i - d_1) < 0$, $d_2 = 0$, $|d| = d_1$, eq. (14) implies $0 < \mu < 1$ (recall that $i < |d|$). Thus $j = \mu(d_1 - i)$. In this case, the smallest $d_1 - i$ for j to be a positive integer is the denominator of μ. Thus, assuming $\mu = \frac{\mu_1}{\mu_2}$ is the irreducible form of μ, the smallest $|d|$ would be μ_2 reached for $i = 0$. Thus $(i,j) = (0, \mu_1)$ and

$$p = x_1^{\mu_2} + a_{0,\mu_1} x_2^{\mu_1}$$

is an irreducible Darboux polynomial (with a cofactor μ_1) for any constant a_{0,μ_1}. (Equivalently, $x^{\mu_2} y^{-\mu_1}$ is an invariant rational function.)

If $\mu(i - d_1) \geq 0$, $d_2 = \mu(i - d_1)$ and $j = 0$. We observe that $d_1 = |d|$ implies $\mu = 0$ or $i = d_1$. The former is impossible by assumption and the latter is impossible by eq. (14). Therefore $\mu = \frac{d_2}{i - d_1}$ and $\mu > 1$ or $\mu < 0$ (since $0 < \mu < 1$ implies $|d| < i$ and $\mu = 1$ was already discussed). Let $\mu = \frac{\pm\mu_1}{\mu_2}$ be the irreducible form of μ where $\mu_1, \mu_2 > 0$.

If $\mu > 1$, then the smallest d_2 is μ_1, $i = d_1 + \mu_2$, and the smallest $|d|$ is reached for $d_1 = 0$. Thus $(i, j) = (\mu_2, 0)$ and

$$p = x_2^{\mu_1} + a_{\mu_2,0} x_1^{\mu_2}$$

is an irreducible Darboux polynomial (with a cofactor μ_1) for any constant $a_{\mu_2,0}$. (Equivalently, $x_1^{-\mu_2} x_2^{\mu_1}$ is an invariant rational function.)

Finally, if $\mu < 0$, then the smallest d_2 is μ_1, $i = d_1 - \mu_2$, and the smallest $|d|$ is $\mu_1 + \mu_2$ obtained for $i = 0$. So $(i, j) = (0, 0)$ and

$$p = x_1^{\mu_2} x_2^{\mu_1} + a_{0,0}$$

is an irreducible Darboux polynomial (with a cofactor 0) for any constant $a_{0,0}$. (Equivalently, $x_1^{\mu_2} x_2^{\mu_1}$ is an invariant function.)

C DECREASING CHAINS OF MONOMIALS

Recall that the number of monomials of degree $k \in \mathbb{N}$ in $n \geq 1$ variables is $[k, n] := \binom{k+n-1}{n-1}$. The monotonicity of $[k, n]$ w.r.t. to its first argument is immediate: if $k \leq k'$, then $[k, n] \leq [k', n]$. The same holds w.r.t. its second argument since $[k, n] = \binom{k+n-1}{k}$: if $n \leq n'$, then $[k, n] \leq [k, n']$.

Let $\Gamma(\beta, \sigma)$, $\beta \in \mathbb{N}^n$, σ a permutation of $\{1, \ldots, n\}$, denote the number of monomials of degree $|\beta|$ lower than x^β for the monomial order $\mathtt{DLex}_\sigma$ with $x_{\sigma(1)} \succ \cdots \succ x_{\sigma(n)}$. For each α_1, $0 \leq \alpha_1 < \beta_{\sigma(1)}$, there are $[|\beta| - \alpha_1, n-1]$ monomials of degree $|\beta|$ which are lower than x^β. Similarly, when $\alpha_1 = \beta_{\sigma(1)}$, for each α_2, $0 \leq \alpha_2 < \beta_{\sigma(2)}$, there are $[|\beta| - \beta_{\sigma(1)} - \alpha_2, n-2]$ monomials of degree $|\beta|$ which are lower than x^β, etc. The general formula for $\Gamma(\beta, \sigma)$ is then

$$\sum_{\alpha_1=0}^{\beta_{\sigma(1)}-1} [|\beta| - \alpha_1, n-1] + \sum_{\alpha_2=0}^{\beta_{\sigma(2)}-1} [|\beta| - \beta_{\sigma(1)} - \alpha_2, n-2] + \cdots + \sum_{\alpha_{n-1}=0}^{\beta_{\sigma(n-1)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(n-2)} - \alpha_{n-1}, 1]. \quad (16)$$

Lemma C.1. *Let σ denote a permutation and suppose that there exists i, such that $\beta_{\sigma(i)} > \beta_{\sigma(i+1)}$. Let σ' denote the permutation obtained from σ by swapping $\sigma(i)$ and $\sigma(i+1)$, that is $\sigma'(i) = \sigma(i+1)$, $\sigma'(i+1) = \sigma(i)$, and $\sigma'(j) = \sigma(j)$ for all indices j distinct from i and $i+1$. Then $\Gamma(\beta, \sigma') < \Gamma(\beta, \sigma)$. In words, when $\beta_{\sigma(i)} > \beta_{\sigma(i+1)}$, by swapping $x_{\sigma(i)}$ and $x_{\sigma(i+1)}$ in the variable ordering, the number of monomials of total degree $|\beta|$ lower that x^β decreases.*

Proof. By definition, $\Gamma(\beta, \sigma)$ and $\Gamma(\beta, \sigma')$ differ by the two sums

$$S_i = \sum_{\alpha_i=0}^{\beta_{\sigma(i)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \alpha_i, n-i]$$

$$S_{i+1} = \sum_{\alpha_{i+1}=0}^{\beta_{\sigma(i+1)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i)} - \alpha_{i+1}, n-i-1]$$

in $\Gamma(\beta, \sigma)$ which are respectively replaced in $\Gamma(\beta, \sigma')$ by

$$S'_i = \sum_{\alpha_i=0}^{\beta_{\sigma(i+1)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \alpha_i, n-i]$$

$$S'_{i+1} = \sum_{\alpha_{i+1}=0}^{\beta_{\sigma(i)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \beta_{\sigma(i+1)} - \alpha_{i+1}, n-i-1]$$

One thus gets $S_i - S'_i$

$$\sum_{\alpha_i=\beta_{\sigma(i+1)}}^{\beta_{\sigma(i)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \alpha_i, n-i] = \sum_{k=0}^{\beta_{\sigma(i)}-\beta_{\sigma(i+1)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \beta_{\sigma(i+1)} - k, n-i].$$

By the change of variables $k = \beta_{\sigma(i)} - \beta_{\sigma(i+1)} + \alpha_{i+1}$, for the index α_{i+1}, the sum S_{i+1} can be equivalently rewritten as

$$\sum_{k=\beta_{\sigma(i)}-\beta_{\sigma(i+1)}}^{\beta_{\sigma(i)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \beta_{\sigma(i+1)} - k, n-i-1].$$

Thus one gets for $S'_{i+1} - S_{i+1}$

$$\sum_{k=0}^{\beta_{\sigma(i)}-\beta_{\sigma(i+1)}-1} [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \beta_{\sigma(i+1)} - k, n-i-1]$$

By monotonicity of $[\cdot, \cdot]$ w.r.t. its second argument, for all k, $0 \leq k \leq \beta_{\sigma(i)} - \beta_{\sigma(i+1)} - 1$,

$$[|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \beta_{\sigma(i+1)} - k, n-i-1] < [|\beta| - \beta_{\sigma(1)} - \cdots - \beta_{\sigma(i-1)} - \beta_{\sigma(i+1)} - k, n-i]$$

Thus $S'_{i+1} - S_{i+1} < S_i - S'_i$ or equivalently $S'_{i+1} + S'_i < S_{i+1} + S_i$ making $\Gamma(\beta, \sigma') < \Gamma(\beta, \sigma)$. □

Proposition C.2. *Let $\beta \in \mathbb{N}^n$ and let σ denote a permutation of $\{1, \ldots, n\}$. Then $\Gamma(\beta, \sigma)$ is minimal for σ if and only if $\beta_{\sigma(1)} \leq \cdots \leq \beta_{\sigma(n)}$.*

Proof. Necessity. By contradiction, suppose that $\Gamma(\beta, \sigma)$ is minimal for a permutation σ that doesn't satisfy $\beta_{\sigma(1)} \leq \cdots \leq \beta_{\sigma(n)}$. That is, there exists an index i such that $\beta_{\sigma(i)} > \beta_{\sigma(i+1)}$. Lemma C.1 provides a permutation σ' for which $\Gamma(\beta, \sigma') < \Gamma(\beta, \sigma)$ contradicting the minimality of $\Gamma(\beta, \sigma)$.

Sufficiency. Suppose that $\beta_{\sigma(1)} \leq \cdots \leq \beta_{\sigma(n)}$. Let $\sigma' \neq \sigma$ denote a permutation such that $\Gamma(\beta, \sigma')$ is minimal. Then $\beta_{\sigma'(1)} \leq \cdots \leq \beta_{\sigma'(n)}$. Since the ordering on natural numbers is total, one gets $\beta_{\sigma(i)} = \beta_{\sigma'(i)}$ for all i. Thus, by eq. (16), $\Gamma(\beta, \sigma') = \Gamma(\beta, \sigma)$ and $\Gamma(\beta, \sigma)$ is also minimal. □

Generalizing the degree by a positive weight $w \in \mathbb{N}^n$, one defines $\Gamma_w(\beta, \sigma)$ as the number of monomials of weight $|\beta|_w$ lower than x^β for $w\mathtt{Lex}$ with $x_{\sigma(1)} \succ \cdots \succ x_{\sigma(n)}$. Proposition C.2 generalizes as follows.

Proposition C.3. *Let $\beta \in \mathbb{N}^n$ and let σ denote a permutation of $\{1, \ldots, n\}$. Then $\Gamma_w(\beta, \sigma)$ is minimal for σ if and only if $w_{\sigma(1)}\beta_{\sigma(1)} \leq \cdots \leq w_{\sigma(n)}\beta_{\sigma(n)}$.*

Proof. (Sketch) For a positive weight vector w, the function $\Gamma_w(\beta, \sigma)$ is defined as in eq. (16), except that $\beta_{\sigma(i)}$ and α_i are scaled by $w_{\sigma(i)}$. By setting α'_i to $w_{\sigma(i)}\alpha_i$, each sum S_i defining $\Gamma_w(\beta, \sigma)$ becomes

$$\sum_{\alpha'_i=0}^{w_{\sigma(i)}\beta_{\sigma(i)}-1} [|\beta|_w - w_{\sigma(1)}\beta_{\sigma(1)} - \cdots - w_{\sigma(i-1)}\beta_{\sigma(i-1)} - \alpha'_i, n-i],$$

and the proofs carry on very similarly to the ones seen above. □