Computing critical points for invariant algebraic systems

Jean-Charles Faugère

Inria, Sorbonne Université, CNRS, LIP6, Équipe PolSys, CryptoNext Security, 4 place Jussieu, F-75252, Paris Cedex 05, France

George Labahn

Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

Mohab Safey El Din

Sorbonne Université, CNRS, LIP6, PolSys, Paris, France

Éric Schost

Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada

Thi Xuan Vu

Sorbonne Université, CNRS, LIP6, PolSys, Paris, France and Cheriton School of Computer Science, University of Waterloo, ON, Canada

Abstract

Let $\mathbb{K}$ be a field and $(f_1, \ldots, f_s, \phi)$ be multivariate polynomials in $\mathbb{K}[x_1, \ldots, x_n]$ (with $s < n$) each invariant under the action of $\mathcal{S}_n$, the group of permutations of $\{1, \ldots, n\}$. We consider the problem of computing the critical points of ϕ restricted to the algebraic set $V(\boldsymbol{f})$, where $\boldsymbol{f} = (f_1, \ldots, f_s)$. This is the same as computing the points at which f vanishes and the Jacobian matrix associated to $(f_1, \ldots, f_s, \phi)$ is rank deficient, provided that this set is finite.

We exploit the invariance properties of the input to split the solution space according to the orbits of $\mathcal{S}_n$. This allows us to design an algorithm which gives a triangular description of the solution space and which runs in time polynomial in d^s, $\binom{n+d}{d}$ and $\binom{n}{s+1}$ where d is the maximum degree of the input polynomials. When d, s are fixed, this is polynomial in n while when s is fixed and $d \simeq n$ this yields an exponential speed-up with respect to the usual polynomial system solving algorithms.

Preprint submitted to Journal of Symbolic Computation *4 November 2022*

Key words: Polynomial systems solving, Invariant algebraic systema, Determinantal systems, Critical points, Computational complexity

1. Introduction

1.1. Motivation and Problem Statement

In this paper we consider the problem of finding the critical points of a polynomial map ϕ restricted to the variety $V(\boldsymbol{f})$. Here $\boldsymbol{f} = (f_1, \dots, f_s)$ and ϕ are symmetric polynomials in the ring $\mathbb{K}[x_1, \dots, x_n]$, with $\mathbb{K}$ a field of characteristic zero. Symmetric polynomials are those invariant under the action of $\mathcal{S}_n$, the group of permutations of $\{1, \dots, n\}$.

More precisely we will look at the closely related problem of computing a description of the set $W(\phi, \boldsymbol{f})$ defined by the following equations:

$$\langle f_1, \dots, f_s \rangle + \langle M_{s+1}(\mathrm{Jac}(\boldsymbol{f}, \phi)) \rangle, \tag{1}$$

where $\mathrm{Jac}(\boldsymbol{f}, \phi)$ is the Jacobian matrix of $(f_1, \dots, f_s, \phi)$ with respect to $(x_1, \dots, x_n)$ and $M_r(\mathbf{G})$ denotes the set of all r-minors of a matrix $\mathbf{G}$. If we assume that the Jacobian matrix $\mathrm{Jac}(\boldsymbol{f})$ has full rank s at any point of $V(\boldsymbol{f})$, then the Jacobian criterion [19, Theorem 16.19] implies that the algebraic set $V(\boldsymbol{f})$ is smooth and $(n-s)$-equidimensional, and that $W(\phi, \boldsymbol{f})$ is indeed the set of critical points of ϕ on $V(\boldsymbol{f})$. Our goal is to describe the critical point set with an improved complexity which takes advantage of the added symmetric structure of the input functions.

The problem of computing critical points appears in many application areas including polynomial optimization [32, 43, 3, 31, 49] and real algebraic geometry [2, 4, 6, 8, 11, 33, 58]. For example, when $\mathbb{K}$ is a real field, then finding critical points provides an effective Morse-theoretic approach to many problems such as real root finding, quantifier elimination or answering connectivity queries (see [5]). In the symmetric case a similar set of applications arise naturally when looking for critical points of functions defined over varieties on an $n-1$ dimensional sphere. For example finding the critical points of $\phi = x_1x_2x_3 - 3x_1 - 3x_2 - 3x_3$ over the sphere defined by $f = x_1^2 + x_2^2 + x_3^2 - 6$ is the same as finding the critical points of the function $\tilde{\phi} : S^2 \mapsto R$ given by

$$\tilde{\phi}(\theta_1, \theta_2) = 2\sin(\theta_1)\cos(\theta_1)\cos(\theta_2) - 2\sin(\theta_1)\cos(\theta_2) - \cos(\theta_1)$$

with θ_1, θ_2 being spherical coordinates.

1.2. Previous work

Prior works encompass three bodies of contributions: *(i)* solving polynomial optimization problems which are invariant under the action of the symmetric group $\mathcal{S}_n$, *(ii)* computing critical points of the restriction of some polynomial map to an algebraic set and *(iii)* solving polynomial systems which are invariant by the action of a finite group.

Email addresses: Jean-Charles.Faugere@inria.fr. (Jean-Charles Faugère), glabahn@uwaterloo.ca. (George Labahn), mohab.safey@lip6.fr (Mohab Safey El Din), eschost@uwaterloo.ca (Éric Schost), txvu@uwaterloo.ca (Thi Xuan Vu).

Polynomial optimization problems under the action of $\mathcal{S}_n$: Polynomial optimization is highly related to deciding non-negativity since computing the infimum of some polynomial map ϕ over a set defined by some polynomial constraints boils down to computing the supremum value $\varphi^\star$ such that $\phi - \varphi^\star$ is non-negative over this set. In [63], Timofte introduces the so-called *degree principle*, which states that a real symmetric polynomial inequality of degree $d \geq 2$ holds in the non-negative orthant if and only if it holds for points with at most $\lfloor \frac{d}{2} \rfloor$ distinct coordinates. This has been further improved and generalized in series of papers by Riener [53, 54] with applications to approaches based on sums of squares decompositions [56]. This series of works leads to algorithms and complexity results which state that, when the maximum degree of ϕ and $\boldsymbol{f} = (f_1, \ldots, f_s)$ is d, then one can decide the non-negativity of ϕ over the real counterpart of $V(\boldsymbol{f})$ in time which is polynomial in the number of variables n. Such results have been extended in various directions considering other group actions such as [55, 64]. All these works rely on the analysis of the orbits of some critical points associated to the map ϕ, identifying the existence of such critical points with a prescribed number of distinct coordinates, hence involving the so-called isotypic decomposition of this set. This analysis is further refined in [48], for the case of sets invariant under symmetric groups $\mathcal{S}_n$, shaping in algebraic terms the degree principle and making explicit the isotypic decomposition of polynomial ideals which are invariant by the action of $\mathcal{S}_n$.

All in all, these works allow one to expect that one should be able to compute $W(\phi, \boldsymbol{f})$ in time which is polynomial in n when the maximum degree of ϕ and entries of $\boldsymbol{f}$ is fixed. Still, taking advantage of the action of $\mathcal{S}_n$ when n and d grows remained an open problem in this context.

Computation of critical points: In the nonsymmetric case, at least when ϕ is linear, there exist algorithms for determining critical points using $d^{O(n)}$ operations in $\mathbb{K}$ [5, Section 14.2]. More precisely, using Gröbner basis techniques, the paper [24, Corollary 3] establishes that, if the polynomials $f_1, \ldots, f_s$ are generic enough of degree d, then this computation can be done using

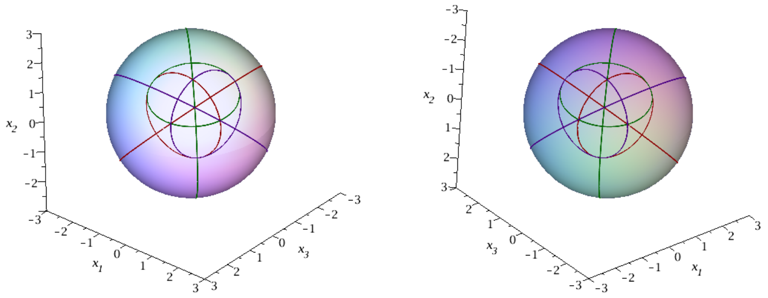
$$O\left(\binom{n + D_{\text{reg}}}{n}^{\omega} + n\left(d^s\,(d-1)^{n-s}\binom{n-1}{s-1}\right)^3\right)$$

operations in $\mathbb{K}$. Here $D_{\text{reg}} = d(s-1) + (d-2)n + 2$, and ω is the exponent of multiplying two $(n \times n)$-matrices with coefficients in $\mathbb{K}$ (see [60] for a generalization to systems with mixed degrees).

Hence, additional special techniques are required when $f_1, \ldots, f_s$ and ϕ are $\mathcal{S}_n$-invariant. One important difficulty to exploit $\mathcal{S}_n$-invariance in this context, is that $f_1, \ldots, f_s$ and ϕ being $\mathcal{S}_n$-invariant does not imply that the individual polynomials in (1) are also invariant. However, we can prove that the set of polynomials in (1) is *globally invariant*. That is, for all σ in $\mathcal{S}_n$, and any g among either $f_1, \ldots, f_s$ or the $(s+1)$-minors of $\text{Jac}(\boldsymbol{f}, \phi)$, either $\sigma(g)$ or $-\sigma(g)$ belongs again to the same set of generators. This implies that $W(\phi, \boldsymbol{f})$ is $\mathcal{S}_n$-invariant.

Example 1. Let $n = 3$ and $s = 1$. In order to determine the critical points of $\phi = x_1x_2x_3 - 3x_1 - 3x_2 - 3x_3$ over the sphere defined by $f = x_1^2 + x_2^2 + x_3^2 - 6$, one has to solve the set of equations defined by

$$\{f\,,\, x_1^2x_3 - x_2^2x_3 - 3x_1 + 3x_2,\, x_1^2x_2 - x_2x_3^2 - 3x_1 + 3x_3,\, x_1x_2^2 - x_1x_3^2 - 3x_2 + 3x_3\,\}.$$



(a) First view

(b) Permuted view

Fig. 1. Critical points of $\phi = x_1x_2x_3 - 3x_1 - 3x_2 - 3x_3$ over the sphere $x_1^2 + x_2^2 + x_3^2 = 6$

One observes that individual generator polynomials are not symmetric but that the algebraic set is globally invariant. Indeed one can see that these critical points are the intersection of three different colored curves in Figure 1. We can see that the set of these critical points is $\mathcal{S}_3$-invariant.

Hence, a key missing ingredient, to leverage the $\mathcal{S}_n$-invariance property for the computation of critical points, is to elaborate computational methods that handle the situation where input systems are globally $\mathcal{S}_n$-invariant but given with polynomials which are *not* individually $\mathcal{S}_n$-invariant.

Solving globally invariant systems: For globally invariant systems, the classical technique which is used is the one of divided differences. Divided differences appear frequently in the context of $\mathcal{S}_n$-equivariant polynomial systems, for example, in [51] and [25] as mentioned above. In addition, given a system of n homogeneous polynomials in n variables, Busé and Karasoulou in [13] prove that its resultant can be decomposed into a product of several resultants. These resultants are easier to compute and can be expressed in terms of the divided differences of the input polynomial system.

The techniques we develop here are more inspired by [25], which, following [22]. There upon input of a polynomial system which is $\mathcal{S}_n$-invariant globally, one uses divided differences to construct a new system where all entries are $\mathcal{S}_n$-invariant. Our work extends this reference, taking into account the specific type of the equations that we solve, that is, those involving minors of a Jacobian matrix, requires us to extend the work from [25]. In our case we will also provide a complexity analysis.

Once, $\mathcal{S}_n$-invariant polynomial systems are transformed to systems where all polynomials are $\mathcal{S}_n$-invariant, there remains the issue of solving them. In Colin [15], the proposed method uses primary and secondary invariants to reformulate the problem (see e.g. [62] for the definition of these invariants). Faugère and Rahmany [23] compute a SAGBI-Gröbner basis in the ring $\mathbb{K}[e_1, \ldots, e_n]$, where e_i is a variable corresponding to i-th elementary symmetric polynomial η_i in $(x_1, \ldots, x_n)$. Steidel [61] designs dedicated Gröbner bases algorithms for this. Note that for all these works, there is no complexity result which stands for general polynomial systems.

In our context, the critical point computations also induce some determinantal structure to take into account in addition to the one coming from the $\mathcal{S}_n$-invariance. Recall also that while we expect complexity results which are polynomial in n when the maximum degree d of the input polynomials is fixed, there was no complexity result that show a discrepancy between the complexity achieved in the general case and the one gotten in the $\mathcal{S}_n$-invariant case when both n and d grow.

1.3. Main results

The global invariance property allows us to split the set $W = W(\phi, \boldsymbol{f})$ into orbits under the action of the symmetric group. It is well known that the size of the orbit of a point in W will depend on the number of pairwise distinct coordinates of that point.

Example 2. Let f and ϕ be as above. The four points $(2, 1, 1), (0, \sqrt{3}, \sqrt{3}), (-2, -1, -1), (0, -\sqrt{3}, -\sqrt{3})$ are solutions with three elements in their respective $\mathcal{S}_3$-orbits, while the two points $(\sqrt{2}, \sqrt{2}, \sqrt{2}), (-\sqrt{2}, -\sqrt{2}, -\sqrt{2})$ are also solutions, with only one point in each of their orbits (this is the complete decomposition of W into orbits).

In order to devise a fast algorithm, the different sizes of orbits needs to be taken into consideration. This phenomenon is to be expected for systems such as (1), but is not discussed for the particular family of equations in [25] (on the other hand, that reference takes into consideration further properties of the family of equations considered therein; we refer the reader to the article [25] for more details of these properties).

The structure of these orbits is determined by the number of pairwise distinct coordinates of the points they contain. To study them, we make use of partitions of n. A sequence $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$, with the ℓ_i and n_i positive integers and $n_1 < \cdots < n_r$, is called a *partition* of n if $n_1\ell_1 + n_2\ell_2 + \cdots + n_r\ell_r = n$. Partitions of n will be used to parametrize orbits, with λ as above parameterizing those points in W having ℓ_1 distinct sets of n_1 equal coordinates, ℓ_2 distinct sets of n_2 equal coordinates and so on. We will write W_λ for the set of such orbits contained in W, so that W is the disjoint union of all W_λ, for all partitions λ of n.

Example 3. For the ϕ and f mentioned previously, our algorithm will determine that the set $W_{(1^3)}$ of orbits parameterized by $\lambda = (1^3)$, which corresponds to the orbits with all distinct coordinates (ξ_1, ξ_2, ξ_3), is equal to the zero set of

$$(f,\ -4,\ -2(x_1 + x_2 + x_3),\ 2(x_1^2 + x_2^2 + x_3^2) + 8(x_1x_2 + x_2x_3 + x_1x_3) - 36)$$

(and so $W_{(1^3)}$ is empty, as we saw above). The set $W_{(1^1\,2^1)}$ of orbits parameterized by $\lambda = (1^1\,2^1)$, that is, orbits of points of the form (ξ_1, ξ_2, ξ_2), with $\xi_1 \neq \xi_2$, is the orbit of the zero set of

$$(x_1^2 + 2x_2^2 - 6,\ x_2^2 + x_1x_2 - 3,\ x_2 - x_3),$$

where the first component is f restricted to the hyperplane $x_2 = x_3$. In particular, $W_{(1^1\,2^1)}$ is the union of the orbits of the points $(2, 1, 1), (0, \sqrt{3}, \sqrt{3}), (-2, -1, -1), (0, -\sqrt{3}, -\sqrt{3})$ seen in Example 2.

Finally, the set $W_{(3^1)}$ of orbits parameterized by $\lambda = (3^1)$, which is orbit of points of the form (ξ_1, ξ_1, ξ_1), is the zero set of $3x_1^2 - 6 = 0$. This polynomial is g restricted to hyperplanes $x_2 = x_1$ and $x_3 = x_1$.

In this paper we provide a procedure to find invariant polynomials that describe these $\mathcal{S}_n$-orbits. For an orbit parameterized by the partition $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$, we work with points which have distinct coordinates $(\xi_{1,1}, \dots, \xi_{1,\ell_1}, \xi_{2,1}, \dots, \xi_{2,\ell_2}, \dots, \xi_{r,1}, \dots, \xi_{r,\ell_r})$, so that instead of n coordinates, there are only $\ell = \ell_1 + \cdots + \ell_r$ distinct coordinates for points in this orbit. Then, the invariance of W under permutations implies that single distinct points are permuted, groups of two points are permuted, etc. This will allow us to work with polynomials in $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r] = \mathbb{K}[e_{1,1}, \dots, e_{1,\ell_1}, e_{2,1}, \dots, e_{2,\ell_2}, \dots, e_{r,1}, \dots, e_{r,\ell_r}]$, in order to represent a certain "compressed" image $W'_\lambda \subset \overline{\mathbb{K}}^\ell$ of W_λ. Here, $e_{i,1}, \dots, e_{i,\ell_i}$ are variables standing for the elementary symmetric polynomials in ℓ_i indeterminates and $\overline{\mathbb{K}}$ is an algebraic closure of $\mathbb{K}$. The process of finding such polynomials in $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r]$, which is presented as the Symmetrize algorithm in Section 3, is one of the main contributions in this paper.

Example 4. In our running example, for $\lambda = (1^1\, 2^1)$, we have $\ell = 2$ and $W'_{(1^1\, 2^1)}$ is the set $\{(2, 1), (0, \sqrt{3}), (-2, -1), (0, -\sqrt{3})\}$.

Throughout the paper, we will assume that W, and thus all W_λ and W'_λ, are finite. Then, for λ as above, the cardinality of W'_λ is smaller than that of W_λ by a factor

$$\nu_\lambda = \ell_1! \cdots \ell_r! \cdot \binom{n}{n_1, \dots, n_1, \dots, n_r, \dots, n_r}, \tag{2}$$

where each n_i in the multinomial coefficient is repeated ℓ_i times. The first part in (2) is obtained from the fact that we compress the set W_λ to W'_λ; it is also the order of the group $\mathcal{S}_{\ell_1} \times \cdots \times \mathcal{S}_{\ell_r}$. Meanwhile, the second part in (2) is the total number permutations of a point $\boldsymbol{\xi}$ of type

$$\boldsymbol{\xi} = (\underbrace{\xi_{1,1}, \dots, \xi_{1,1}}_{n_1}, \quad \dots, \quad \underbrace{\xi_{1,\ell_1}, \dots, \xi_{1,\ell_1}}_{n_1}, \quad \dots, \quad \underbrace{\xi_{r,1}, \dots, \xi_{r,1}}_{n_r}, \quad \dots, \quad \underbrace{\xi_{r,\ell_r}, \dots, \xi_{r,\ell_r}}_{n_r}),$$

where $\xi_{i,j}$'s are pairwise distinct.

The idea of using group orbits to describe the critical points of (1) when all polynomials f_i are symmetric is already in [51]. There, the authors also use the partitions of n to give a description of critical points of a homogeneous symmetric polynomial in $(x_1, \dots, x_n)$. However they do not use polynomials in $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r]$ to encode the critical points for each group orbit, something which is vital to the efficiency of our computations.

Altogether, if d is the maximum of the degrees of the input of polynomials, then we will prove bounds, denoted by $\mathfrak{c}_\lambda$, on the cardinality of the finite set W'_λ. We will see that, in practice, each of the $\mathfrak{c}_\lambda$ provides an accurate bound on the cardinality of W'_λ. The sum of the $\mathfrak{c}_\lambda$'s then gives us an upper bound on the size of the output of our main algorithm. There is no closed formula for this sum, but we prove it is bounded above by

$$\mathfrak{c} = d^s \binom{n+d-1}{n}. \tag{3}$$

We will see that, in practice, this is a rather rough upper bound but in several cases, it compares well to the upper bound

$$\tilde{\mathfrak{c}} = d^s\, (d-1)^{n-s} \binom{n}{s} \tag{4}$$

from Nie and Ranestad [50, Theorem 2.2] on the size of W. For example, when $d = 2$, we have $\mathfrak{c} = 2^s(n+1)$ while $\tilde{\mathfrak{c}} = 2^s\binom{n}{s}$. More generally, when d and s are fixed, $\mathfrak{c}$ is polynomial in n (since it is bounded above by $d^s(n+d-1)^d$) while $\tilde{\mathfrak{c}}$ is exponential in n (since it is greater than $(d-1)^n$). When s is fixed and $d = n$, $\mathfrak{c}$ is $n^{O(1)}2^n$, whereas $\tilde{\mathfrak{c}}$ is $n^{O(1)}(n-1)^{n-s}$. Observe additionally that when $d \simeq n^\alpha$ with $\alpha < 1$, $\mathfrak{c}$ is subexponential in n. The following complexity result illustrates the importance of these facts, as the computation of $W(\phi, \boldsymbol{f})$ is polynomial in $\mathfrak{c}, d^s$ and $\binom{n}{s+1}$.

Theorem 5. Let $\boldsymbol{f} = (f_1, \ldots, f_s)$ and ϕ be $\mathcal{S}_n$-invariant polynomials in $\mathbb{K}[x_1, \ldots, x_n]$, with degree at most $d \geq 2$. Suppose further that $W = W(\phi, \boldsymbol{f})$ is finite. Then there exists a randomized algorithm that takes $\boldsymbol{f}, \phi$ as input and outputs a symmetric representation for the set W, and whose runtime is polynomial in d^s, $\binom{n+d}{d}$, and $\binom{n}{s+1}$. The total number of points described by the output is at most $d^s\binom{n+d-1}{n}$.

Hence, $W(\phi, \boldsymbol{f})$ can be computed in time polynomial in n when both d and s are fixed, which match some statements based on the degree principle in [53]. Note that when s is fixed, but $d \leq n^\alpha$ with $\alpha < 1$, one novelty is that our algorithm runs in time subexponential in n. Section 5 gives a more precise estimate on the runtime of the algorithm.

1.4. Some ingredients

In view of the previous discussion, our algorithm will naturally want to compute descriptions of the sets W'_λ rather than W_λ. Of course we will also explain how one would recover the later knowing the former. While there are a number of ways to represent algebraic sets, in our case it is convenient make use of a representation based on univariate polynomials. If $Y \subset \overline{\mathbb{K}}^n$ is a zero-dimensional variety, then a *zero-dimensional parametrization* $\mathscr{R} = ((v, v_1, \ldots, v_n), \mu)$ of Y consists of

(i) a squarefree polynomial q in $\mathbb{K}[y]$, where y is a new indeterminate and $\deg(q) = |Y|$,

(ii) polynomials $(v_1, \ldots, v_n)$ in $\mathbb{K}[y]$ with $\deg(v_i) < \deg(v)$ for all i, and satisfying $Y = \{(\frac{v_1(\tau)}{v'(\tau)}, \ldots, \frac{v_n(\tau)}{v'(\tau)}) \in \overline{\mathbb{K}}^m \mid v(\tau) = 0\}$ with $v' = \frac{\partial v}{\partial y}$,

(iii) a linear form μ in n variables such that $\mu(v_1, \ldots, v_n) = yv'$.

When these conditions hold, we write $Y = Z(\mathscr{R})$.

The last condition says that the roots of v are the values taken by the linear form μ on y. This representation was first introduced in the works of Kronecker and König [44] and has been widely used in computer algebra [1, 27, 28, 29, 30, 57]. The output of our algorithm will be a collection of zero-dimensional parametrizations, one for each of the sets W'_λ. We will call such a parametrization a *symmetric representation* of W (precise definitions are in Section 2).

At this stage we can use Gröbner bases to compute the descriptions giving a deterministic algorithm. However as we are also interested in determining a good complexity we will instead use *symbolic homotopy continuation*, as this will allow us to precisely control the cost of the computation. Homotopy continuation has become a foundational tool for numerical algorithms while the use of symbolic homotopy continuation algorithms is more recent. Such algorithms first appeared in [10, 36], for general inputs, and later for sparse [42, 37, 38, 39] and multi-homogeneous systems [59, 35, 34].

In our case, we can make use of a recent sparse symbolic homotopy method given in [45] specifically designed to handle determinantal systems over weighted polynomial rings, that is, multivariate polynomial rings where each variable has a weighted degree, which is a positive integer. These domains arise naturally for our orbits: the domain arising from an orbit parameter λ has variables $e_{i,k}$ which are defined corresponding to elementary symmetric polynomials $\eta_{i,k}$. Since $\eta_{i,k}$ has degree k, the variable $e_{i,k}$ will naturally be assigned weight k.

Further, we use standard notions and notations of commutative algebra and algebraic geometry which can be found in [16, 19]. We will assume that the reader is familiar with concepts such as dimension, Zariski topology, equidimensional algebraic set and the degree of an algebraic set.

Throughout the paper, the multivariate polynomials we work with are encoded using their dense representation. A possible alternative would be using *straight-line programs*. Some of our subroutines can naturally be written with that data-structure in mind: this is the case for the symbolic homotopy continuation algorithm, but also for the Symmetrize algorithm in Section 3 (it follows by previous work by Bläser and Jindal, which was written in a straight-line program model). However, a complete analysis in this model still remains to be done.

1.5. Organization of paper

The remainder of the paper is organized as follows. In the next section, we provide several properties of invariant polynomials and discuss in detail the sets W_λ and W'_λ mentioned above. Section 3 gives our Symmetrize algorithm for constructing invariant generators of our invariant ideals, while Section 4 contains our main Critical_Points_Per_Orbit algorithm along with its proof of correctness. The runtime of this algorithm is analyzed in Section 5, finishing the proof of Theorem 5. Experiments to validate our new algorithm are given in Section 6 followed by a concluding section which gives topics for future research. Section 7 also includes a discussion on how our results can decide emptiness of $\mathcal{S}_n$-invariant algebraic sets over a real field. Finally, the appendices include the proofs of three technical statements.

2. Partitions and distinct coordinates of $\mathcal{S}_n$-invariants

As noted in the introduction $(f_1, \ldots, f_s)$ and ϕ being $\mathcal{S}_n$-invariant does not imply that the equations in (1) are invariant. Fundamental to our results is the fact that $W(\phi, \boldsymbol{f})$ is invariant under the action of the symmetric group, a direct consequence of the chain rule, as shown below.

Lemma 6. Let g be in $\mathbb{K}[x_1, \ldots, x_n]$ and σ in $\mathcal{S}_n$. Then for k in $\{1, \ldots, n\}$, we have

$$\sigma\left(\frac{\partial g}{\partial x_k}\right) = \frac{\partial(\sigma(g))}{\partial x_{\sigma(k)}}. \tag{5}$$

Corollary 7. The algebraic set $W(\phi, \boldsymbol{f})$ is $\mathcal{S}_n$-invariant.

Proof. Let $\boldsymbol{\xi}$ be in W and σ be in $\mathcal{S}_n$. We need to show that $\sigma(\boldsymbol{\xi})$ is in W, that is, $f_i(\sigma(\boldsymbol{\xi})) = 0$ for all i and $\mathrm{Jac}(\boldsymbol{f}, \phi)$ has rank at most s at $\sigma(\boldsymbol{\xi})$.

The first statement is clear, since $\boldsymbol{\xi}$ cancels $\boldsymbol{f}$ and $\boldsymbol{f}$ is $\mathcal{S}_n$-invariant. For the second claim, since all f_i's and ϕ are $\mathcal{S}_n$-invariant, Lemma 6 implies that the Jacobian matrix $\mathrm{Jac}(\boldsymbol{f}, \phi)$ at $\sigma(\boldsymbol{\xi})$ is equal to $(\mathrm{Jac}(\boldsymbol{f}, \phi)(\boldsymbol{\xi}))\boldsymbol{A}^{-1}$, where $\boldsymbol{A}$ is the matrix of σ. Therefore, as with $\mathrm{Jac}(\boldsymbol{f}, \phi)(\boldsymbol{\xi})$, it has rank at most s. □

The invariance of $W(\phi, \boldsymbol{f})$ allows us to split of $W(\phi, \boldsymbol{f})$ into subsets defined by the orbits of the symmetric group $\mathcal{S}_n$. An orbit is a set of the form $\mathcal{S}_n(\boldsymbol{\xi})$, for some point $\boldsymbol{\xi}$ in $\overline{\mathbb{K}}^n$, that is, the set of all $\mathcal{S}_n$-conjugates of $\boldsymbol{\xi}$. As mentioned in the introduction, the size of an orbit $\mathcal{S}_n(\boldsymbol{\xi})$ will depend on the number of pairwise distinct coordinates of $\boldsymbol{\xi}$. For example, with $n = 3$, a point of the form (ξ_1, ξ_2, ξ_2) will have an orbit of size 3, unless we have $\xi_1 = \xi_2$ (in which case the orbit has size 1). As a result, in this section we will consider the separation of distinct coordinates in an orbit.

2.1. Partitions

Partitions play a major role in describing our orbits, with each orbit represented by a single point. In this subsection, we gather the basic definitions of partitions and a number of properties used throughout this section. A detailed description of these partitions can be found in many combinatorics books, for example, in [47].

A sequence $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$, where $n_1 < \cdots < n_r$ and the ℓ_i's and n_i's are positive integers, is called a *partition* of n, sometimes denoted by $\lambda \vdash n$, if $n_1\ell_1 + n_2\ell_2 + \cdots + n_r\ell_r = n$. The number $\ell = \sum_{i=1}^{r} \ell_i$ is called the *length* of the partition λ. To any partition $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$, we can associate (in a one-to-one manner) the ordered list $(n_1, \dots, n_1, \dots, n_r, \dots, n_r)$, with each n_i repeated ℓ_i times.

We will make use of the *refinement order* on partitions, with the naming based on the fact that $\lambda \le \lambda'$ if and only if partition λ refines λ'. Formally the definition makes use of unions of partitions: if λ and λ' are partitions of a and a', respectively, then $\lambda \cup \lambda'$ is the partition of $a + a'$ whose ordered list is obtained by merging those of λ and λ'. Then for two partitions $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$ and $\lambda' = (m_1^{k_1}\, m_2^{k_2}\, \dots\, m_s^{k_s})$ of the same integer n, we write $\lambda \le \lambda'$ if λ' is the union of some partitions $(\mu_{i,j})_{1 \le i \le s, 1 \le j \le k_i}$, where $\mu_{i,j}$ is a partition of m_i for all i, j. We also say that λ *refines* λ' in this case.

Example 8. For the partitions of $n = 3$, we have $(1^3) \le (1^1 2^1) \le (3^1)$ since $(1^1 2^1)$ is a partition of 3 while (1^2) is a partition of 2.

The refinement order on partitions will later allow us, in Subsection 4.2, to study only partitions of the length at least s, the size of the vector $\boldsymbol{f}$.

Let $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$ be a partition of n having length ℓ. For $1 \le k \le r$, we will denote a sequence of ℓ_k indeterminates by $\boldsymbol{Z}_k = (z_{k,1}, \dots, z_{k,\ell_k})$. When convenient, we denote $(\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r) = (z_{1,1}, \dots, z_{r,\ell_r})$ as $(z_1, \dots, z_\ell)$, so that $z_1 = z_{1,1}, \dots, z_\ell = z_{r,\ell_r}$. We will let $\mathcal{S}_\lambda$ be the group

$$\mathcal{S}_\lambda = \mathcal{S}_{\ell_1} \times \cdots \times \mathcal{S}_{\ell_r}.$$

The group $\mathcal{S}_\lambda$ acts naturally on $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$, and we let $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]^{\mathcal{S}_\lambda}$ be the $\mathbb{K}$-algebra of $\mathcal{S}_\lambda$-invariant polynomials. Note that $\mathcal{S}_\lambda$ can be seen as a subgroup of the permutation group $\mathcal{S}_\ell$ of $\{1, \dots, \ell\}$, where $\mathcal{S}_{\ell_1}$ acts on the first ℓ_1 indices, $\mathcal{S}_{\ell_2}$ acts on the next ℓ_2 indices, etc.

2.2. $\mathcal{S}_\lambda$-invariant polynomials: the Symmetric_Coordinates *algorithm*

Let $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$ be a partition of n having length ℓ, and, for $i = 1, \dots, r$, let $\boldsymbol{e}_i = (e_{i,1}, \dots, e_{i,\ell_i})$ be a set of ℓ_i new variables. Then, by the fundamental theorem of symmetric polynomials [17, Theorem 3.10.1], for any f in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]^{\mathcal{S}_\lambda}$, there exists a unique $\bar{f}$ in $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r]$ with

$$f(\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r) = \bar{f}(\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_r), \tag{6}$$

where $\boldsymbol{\eta}_i = (\eta_{i,1}, \ldots, \eta_{i,\ell_i})$ denotes the vector of elementary symmetric polynomials in variables $\boldsymbol{Z}_i$, with each $\eta_{i,j}$ having degree j for all i, j. We will need a quantitative version of this existence result, which gives an estimate on the cost of computing $\bar{f}$ from f.

Our $\mathsf{Symmetric\_Coordinates}$, formalized in the next lemma, is a slight generalization of the procedure described in the proof of Bläser and Jindal's algorithm [9, Theorem 4], which was written only for the case of $r = 1$, and for polynomials represented as straight-line programs. The main idea of their algorithm is to use the fact that $z_{i,j}$ can be written as a function of $\eta_{i,j}$. For example, consider $r = 1$, $\boldsymbol{Z}_1 = (z_1, z_2)$ and $\boldsymbol{\eta}_1 = (\eta_1, \eta_2) = (z_1 + z_2, z_1 z_2)$. Then z_1 and z_2 are the roots of polynomial

$$P(T) = T^2 - (z_1 + z_2)T + z_1 z_2 = T^2 - \eta_1 T + \eta_2,$$

and so $z_1 = \frac{\eta_1 + \sqrt{\eta_1^2 - 4\eta_2}}{2}$ and $z_2 = \frac{\eta_1 - \sqrt{\eta_1^2 - 4\eta_2}}{2}$. If we substitute these functions to f we obtain $\bar{f}$. However, these functions are neither polynomials nor power series. In order to deal with this situation, we use the substitution $\eta_1 = \eta_1 - 3$ and $\eta_2 = \eta_2 - 2$ which will allow us to express z_1, z_2 in the power series of η_1, η_2 by using Taylor expansion.

Lemma 9. There exists an algorithm $\mathsf{Symmetric\_Coordinates}(\lambda, f)$ which, given a partition λ of n and f of degree at most d in $\mathbb{K}[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_r]^{\mathcal{S}_\lambda}$, returns $\bar{f}$ such that $f = \bar{f}(\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_r)$, using $O\tilde{}(\binom{\ell+d}{d}^2)$ operations in $\mathbb{K}$.[1]

Proof. The key to the algorithm is the following. Let $\mathbb{L}$ be the ring of multivariate power series $\mathbb{L} = \mathbb{K}[[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r]]$. Then this ring contains $\mathbb{K}[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r]$ and vectors $\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r$ where for each i, $\boldsymbol{\zeta}_i = (\zeta_{i,1}, \ldots, \zeta_{i,\ell_i}) \in \mathbb{L}^{\ell_i}$ are the ℓ_i pairwise distinct roots of

$$P_i(T) = T^{\ell_i} - (e_{i,1} + \rho_{i,1})T^{\ell_i - 1} + \cdots + (-1)^{\ell_i} (e_{i,\ell_i} + \rho_{i,\ell_i}),$$

and where $\rho_{i,1}, \ldots, \rho_{i,\ell_i}$ are the elementary symmetric polynomials evaluated at $1, \ldots, \ell_i$.

Thus $\bar{f}$ satisfies $\bar{f}(e_{1,1} + \rho_{1,1}, \ldots, e_{r,\ell_r} + \rho_{r,\ell_r}) = f(\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r)$. Our construction, involving the shifts by $(\rho_{1,1}, \ldots, \rho_{r,\ell_r})$ shows that at $\boldsymbol{e}_1 = \cdots = \boldsymbol{e}_r = 0$, $P_i(T)$ factors as $(T-1)\cdots(T-\ell_i)$.

Applying Newton's iteration, we deduce the existence of the requested power series roots $\boldsymbol{\zeta}_i = (\zeta_{i,1}, \ldots, \zeta_{i,\ell_i})$. In order to obtain the polynomial $\bar{f}$, we only need truncations of these roots at precision d. For $i = 1, \ldots, r$, we can obtain the truncation of $\boldsymbol{\zeta}_i$ using $O\tilde{}(\ell_i \binom{\ell_i + d}{d})$ operations in $\mathbb{K}$, where the factor $\binom{\ell_i + d}{d}$ accounts for the cost of multivariate power series arithmetic [46]. Taking all i's into account, this adds up to $O\tilde{}(\ell \binom{\ell + d}{d})$ arithmetic operations.

We then evaluate f at these truncated power series. Since f has degree at most d, this can be done using $O(\binom{\ell+d}{d})$ $(+, \times)$ operations on ℓ-variate power series truncated in degree d, for a total of $O\tilde{}(\binom{\ell+d}{d}^2)$ operations in $\mathbb{K}$. This gives us $\bar{f}(e_{1,1} + \rho_{1,1}, \ldots, e_{r,\ell_r} + \rho_{r,\ell_r})$. We then apply the translation $(e_{i,j})_{i,j} \leftarrow (e_{i,j} - \rho_{i,j})_{i,j}$ in order to obtain the polynomial $\bar{f}$, also at a cost of $O\tilde{}(\binom{\ell+d}{d}^2)$ operations in $\mathbb{K}$: through successive multiplications, we incrementally compute the translates of all monomials of degree up to d and then, before combining, using the coefficients of $\bar{f}(e_{1,1} + \rho_{1,1}, \ldots, e_{r,\ell_r} + \rho_{r,\ell_r})$. □

[1] Here and in the rest of our paper we use $O\tilde{}(\cdot)$ to indicate that polylogarithmic factors are omitted, that is, f is $O\tilde{}(g)$ if there exists a constant k such that f is $O(g \log^k(g))$.

2.3. Symmetric representations

In this subsection we describe the geometry of $\mathcal{S}_n$-orbits in $\overline{\mathbb{K}}^n$, define the data structure we use to represent $\mathcal{S}_n$-invariant sets, and finally present some basic algorithms related to these invariant sets.

2.3.1. The mapping E_λ and its fibers.

For a partition $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$ of n, we define the following two subsets of $\overline{\mathbb{K}}^n$:

(i) $\mathcal{C}_\lambda$: the set of all points $\boldsymbol{\xi}$ in $\overline{\mathbb{K}}^n$ that can be written as

$$\boldsymbol{\xi} = \big(\underbrace{\xi_{1,1},\dots,\xi_{1,1}}_{n_1},\ \dots,\ \underbrace{\xi_{1,\ell_1},\dots,\xi_{1,\ell_1}}_{n_1},\ \dots,\ \underbrace{\xi_{r,1},\dots,\xi_{r,1}}_{n_r},\ \dots,\ \underbrace{\xi_{r,\ell_r},\dots,\xi_{r,\ell_r}}_{n_r}\big). \quad (7)$$

(ii) $\mathcal{C}_\lambda^{\mathrm{strict}}$: the set of all $\boldsymbol{\xi}$ in $\mathcal{C}_\lambda$ for which the $\xi_{i,j}$'s in (7) are pairwise distinct.

To any point $\boldsymbol{\xi}$ in $\overline{\mathbb{K}}^n$ we can associate its *type*: this is the unique partition λ of n such that there exists σ in $\mathcal{S}_n$ for which $\sigma(\boldsymbol{\xi})$ lies in $\mathcal{C}_\lambda^{\mathrm{strict}}$. Since all points in an orbit have the same type, we can then define the type of an orbit as the type of any point in it. Any orbit of type $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$ has size

$$\gamma_\lambda = \binom{n}{n_1,\dots,n_1,\dots,n_r,\dots,n_r} = \frac{n!}{{n_1!}^{\ell_1}\cdots {n_r!}^{\ell_r}}$$

since the stabilizer of a point in $\mathcal{C}_\lambda^{\mathrm{strict}}$ is $\mathcal{S}_{n_1}^{\ell_1} \times \cdots \times \mathcal{S}_{n_r}^{\ell_r}$.

For efficiency purposes it is very important in our work to not count points in a given orbit multiple times. This is where the refinement order on partitions plays an important role. Notice that all points in $\mathcal{C}_\lambda^{\mathrm{strict}}$ have type λ. However the definition of refinement order implies that $\mathcal{C}_\lambda$ contains points of type λ' for all $\lambda' \geq \lambda$. More precisely, $\mathcal{C}_\lambda$ is the disjoint union of all $\mathcal{C}_{\lambda'}^{\mathrm{strict}}$ for all $\lambda' \geq \lambda$.

Example 10. For the partitions of $n = 3$, we have $(1^3) < (1^1 2^1) < (3^1)$. In addition,

(a) $\mathcal{C}_{(1^3)}$ is $\overline{\mathbb{K}}^3$, while $\mathcal{C}_{(1^3)}^{\mathrm{strict}}$ is the set of all points $\boldsymbol{\xi}$ with pairwise distinct coordinates.

(b) $\mathcal{C}_{(1^1 2^1)}$ is the set of points that can be written $\boldsymbol{\xi} = (\xi_{1,1}, \xi_{2,1}, \xi_{2,1})$, while $\mathcal{C}_{(1^1 2^1)}^{\mathrm{strict}}$ is the subset satisfying $\xi_{1,1} \neq \xi_{2,1}$.

(c) $\mathcal{C}_{(3^1)} = \mathcal{C}_{(3^1)}^{\mathrm{strict}}$ is the set of points $\boldsymbol{\xi} = (\xi_{3,1}, \xi_{3,1}, \xi_{3,1})$ whose coordinates are all equal.

For the partition $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$, we define a mapping $E_\lambda : \mathcal{C}_\lambda \to \overline{\mathbb{K}}^\ell$ by

$$E_\lambda : \boldsymbol{\xi} \text{ as in (7)} \mapsto (\eta_i(\xi_{i,1},\dots,\xi_{i,\ell_i}),\dots,\eta_{\ell_i}(\xi_{i,1},\dots,\xi_{i,\ell_i}))_{1\leq i\leq r},$$

where for $i = 1,\dots,r$ and $j = 1,\dots,\ell_i$, $\eta_j(\xi_{i,1},\dots,\xi_{i,\ell_i})$ is the degree j elementary symmetric function in $\xi_{i,1},\dots,\xi_{i,\ell_i}$. One should see this mapping as a means to compress orbits: through the application of E_λ, one can represent an entire orbit $\mathcal{O}$ of type λ, which has size γ_λ, by the single point $E_\lambda(\mathcal{O} \cap \mathcal{C}_\lambda) = E_\lambda(\mathcal{O} \cap \mathcal{C}_\lambda^{\mathrm{strict}})$.

To put this into practice, we need to be able to recover an orbit from its image. The mapping E_λ is onto: for $\boldsymbol{\varepsilon} = (\varepsilon_{1,1},\dots,\varepsilon_{r,\ell_r})$ in $\overline{\mathbb{K}}^\ell$, define polynomials $P_1(T),\dots,P_r(T)$ by

$$P_i(T) = T^{\ell_i} - \varepsilon_{i,1}T^{\ell_i - 1} + \cdots + (-1)^{\ell_i}\varepsilon_{i,\ell_i}.$$

We can then find a point $\boldsymbol{\xi}$ in the preimage $E_\lambda^{-1}(\boldsymbol{\varepsilon})$ by finding the roots $\xi_{i,1},\dots,\xi_{i,\ell_i}$ of $P_i(T)$. Since we will use this idea often, we will write $E_\lambda^*(\boldsymbol{\varepsilon}) = \mathcal{S}_n(\boldsymbol{\xi})$ for the orbit of any such point $\boldsymbol{\xi}$ in $E_\lambda^{-1}(\boldsymbol{\varepsilon})$. This is well-defined, as all points in this fiber are $\mathcal{S}_n$-conjugate. More generally, for a set G in $\overline{\mathbb{K}}^\ell$, we will write $E_\lambda^*(G)$ for the union of the orbits $E_\lambda^*(\boldsymbol{\varepsilon})$, for $\boldsymbol{\varepsilon}$ in G.

The image $E_\lambda(\mathcal{C}_\lambda^{\mathrm{strict}})$ of points having type λ is an open subset $O_\lambda \subsetneq \overline{\mathbb{K}}^\ell$, defined by the condition that the polynomials P_i above are pairwise coprime and squarefree. For $\boldsymbol{\varepsilon}$ in $\overline{\mathbb{K}}^\ell \setminus O_\lambda$, the orbit $E_\lambda^*(\boldsymbol{\varepsilon})$ does not have type λ, but rather type λ', for some partition $\lambda' > \lambda$.

Example 11. With $n = 3$ and $\lambda = (1^1 2^1)$, a partition of length $\ell = 2$, we see that E_λ maps points of the form $(\xi_{1,1}, \xi_{2,1}, \xi_{2,1})$ to $(\xi_{1,1}, \xi_{2,1})$. In this case we have two polynomials, P_1, P_2 given by $P_1(T) = T - \varepsilon_{1,1}$ and $P_2(T) = T - \varepsilon_{2,1}$, with O_λ defined by $\varepsilon_{1,1} \neq \varepsilon_{2,1}$.

The point $\boldsymbol{\varepsilon} = (2, 3)$ is in O_λ with the orbit $E_\lambda^*(2, 3)$ being $\{(2, 3, 3), (3, 2, 3), (3, 3, 2)\}$. On the other hand, $\boldsymbol{\varepsilon} = (1, 1)$ is not in O_λ. In this case the orbit $E_\lambda^*(1, 1)$ is the point $\{(1, 1, 1)\}$, and has type $(3^1) > (1^1 2^1)$. Finally, if we define $G = \{(1, 1), (2, 3)\}$, then $E_\lambda^*(G)$ is the set $W = \{(1, 1, 1), (2, 3, 3), (3, 2, 3), (3, 3, 2)\}$.

We will need an algorithm that computes the type λ' of the orbit $E_\lambda^*(\boldsymbol{\varepsilon})$, for a given $\boldsymbol{\varepsilon}$ in $\mathbb{K}^\ell$, and also computes the value that the actual compression mapping $E_{\lambda'}$ takes at this orbit. The algorithm's specification assumes inputs in $\mathbb{K}$ (since our computation model is a RAM over $\mathbb{K}$) but the procedure makes sense over any field extension of $\mathbb{K}$. We will use this remark later in the proof of Lemma 17.

Lemma 12. There exists an algorithm $\mathsf{Type\_Of\_Fiber}(\lambda, \boldsymbol{\varepsilon})$ which takes as input a partition λ of n with length ℓ and a point $\boldsymbol{\varepsilon}$ in $\mathbb{K}^\ell$, and returns a partition λ' of n of length k and a tuple $\boldsymbol{f}$ in $\mathbb{K}^k$, such that

(i) λ' is the type of the orbit $\mathcal{O} := E_\lambda^*(\boldsymbol{\varepsilon})$ and

(ii) $E_{\lambda'}(\mathcal{O} \cap \mathcal{C}_{\lambda'}^{\mathrm{strict}}) = \{\boldsymbol{f}\}$.

The algorithm runs in time $O\tilde{~}(n)$.

Proof. Write $\boldsymbol{\varepsilon} = (\varepsilon_{1,1}, \ldots, \varepsilon_{r,\ell_r})$. The points in $E_\lambda^{-1}(\boldsymbol{\varepsilon})$ are obtained as permutations of

$$\boldsymbol{\xi} = (\underbrace{\xi_{1,1}, \ldots, \xi_{1,1}}_{n_1}, \quad \ldots, \quad \underbrace{\xi_{1,\ell_1}, \ldots, \xi_{1,\ell_1}}_{n_1}, \quad \ldots, \quad \underbrace{\xi_{r,1}, \ldots, \xi_{r,1}}_{n_r}, \quad \ldots, \quad \underbrace{\xi_{r,\ell_r}, \ldots, \xi_{r,\ell_r}}_{n_r}),$$

where for $i = 1, \ldots, r$, $\xi_{i,1}, \ldots, \xi_{i,\ell_i}$ are the roots of

$$P_i(T) = T^{\ell_i} - \varepsilon_{i,1} T^{\ell_i - 1} + \cdots + (-1)^{\ell_i} \varepsilon_{i,\ell_i} = 0.$$

Finding the type of such a point $\boldsymbol{\xi}$ amounts to finding the duplicates among the $\xi_{i,j}$'s. The latter can be done by computing the product

$$P = \left(T^{\ell_1} - \varepsilon_{1,1} T^{\ell_1 - 1} + \cdots + (-1)^{\ell_1} \varepsilon_{1,\ell_1}\right)^{n_1} \cdots \left(T^{\ell_r} - \varepsilon_{r,1} T^{\ell_r - 1} + \cdots + (-1)^{\ell_r} \varepsilon_{r,\ell_r}\right)^{n_r}$$

and its square-free factorization $P = Q_1^{m_1} \cdots Q_s^{m_s}$, with $m_1 < \cdots < m_s$ and all Q_i's squarefree and pairwise coprime. If $k_i = \deg(Q_i)$ then $\boldsymbol{\xi}$ has type $\lambda' = (m_1^{k_1} m_2^{k_2} \ldots m_s^{k_s})$ with $\lambda' > \lambda$. If we write

$$Q_i = T^{k_i} - f_{i,1} T^{k_i - 1} + \cdots + (-1)^{k_i} f_{i,k_i}, \quad 1 \le i \le s,$$

then our output is $(\lambda', \boldsymbol{f})$, where $\boldsymbol{f} = (f_{1,1}, \ldots, f_{s,k_s})$.

Using subproduct tree techniques [26, Chapter 10] to compute P and fast GCD [26, Chapter 14], all computations take quasi-linear time $O\tilde{~}(n)$. □

Example 13. Let $n = 3$ and $\lambda = (1^1 2^1)$, with $E_\lambda(\xi_{1,1}, \xi_{2,1}, \xi_{2,1}) = (\xi_{1,1}, \xi_{2,1})$. We saw that for $\boldsymbol{\varepsilon} = (1, 1) \in \mathbb{K}^2$, the orbit $E^*_\lambda(1, 1)$ is $\{(1, 1, 1)\}$, with type $\lambda' = (3^1)$.

Since $n_1 = 1$ and $n_2 = 2$, the $\mathsf{Type\_Of\_Fiber}$ algorithm first expands $(T - 1)(T - 1)^2$ as $T^3 - 3T^2 + 3T - 1$ and then computes its squarefree factorization as $(T - 1)^3$. From this, we read off that $s = 1$, $m_1 = 3$ and $k_1 = 1$, so that λ' is (3^1). The output is $(\lambda', E_{\lambda'}(1, 1, 1))$, the latter being equal to (1).

2.3.2. Representing $\mathcal{S}_n$-invariant sets.

The previous setup allows us to represent invariant sets in $\overline{\mathbb{K}}^n$ as follows. Let W be a set in $\overline{\mathbb{K}}^n$, invariant under the action of $\mathcal{S}_n$. For a partition λ of n with ℓ, we write

$$W_\lambda = \mathcal{S}_n(W \cap \mathcal{C}^{\text{strict}}_\lambda) \subset \overline{\mathbb{K}}^n \quad \text{and} \quad W'_\lambda = E_\lambda(W \cap \mathcal{C}^{\text{strict}}_\lambda) \subset \overline{\mathbb{K}}^\ell, \tag{8}$$

where $\mathcal{S}_n(W \cap \mathcal{C}^{\text{strict}}_\lambda)$ is the orbit of $W \cap \mathcal{C}^{\text{strict}}_\lambda$ under $\mathcal{S}_n$, or, equivalently, the set of points of type λ in W (so this matches the notation used in the introduction).

For two distinct partitions λ, λ' of n, W_λ and $W_{\lambda'}$ are disjoint, so that any invariant set W can be written as the disjoint union $W = \sqcup_{\lambda \vdash n} W_\lambda$. When W is finite, we then can represent W_λ by describing the image W'_λ. Indeed, the cardinality of the set W'_λ is smaller than that of the orbit W_λ by a factor of γ_λ, and we can recover W_λ as $W_\lambda = E^*_\lambda(W'_\lambda)$. Altogether, we are led to the following definition.

Definition 14. Let W be a finite set in $\overline{\mathbb{K}}^n$, defined over $\mathbb{K}$ and $\mathcal{S}_n$-invariant. A *symmetric representation* of W is a sequence $(\lambda_i, \mathscr{R}_i)_{1 \le i \le N}$, where the λ_i's are all the partitions of n for which W_{λ_i} is not empty, and, for each i, $\mathscr{R}_i$ is a zero-dimensional parametrization of W'_{λ_i}.

Example 15. Suppose $n = 3$ and

$$W = \{(1, 1, 1), (2, 3, 3), (3, 2, 3), (3, 3, 2)\}.$$

Then with $\lambda = (1^1 2^1)$ we have $W_\lambda = \{(2, 3, 3), (3, 2, 3), (3, 3, 2)\}$, $W'_\lambda = \{(2, 3)\} \subset \overline{\mathbb{K}}^2$ and $\gamma_\lambda = 3$, while with $\lambda' = (3^1)$, we have $W_{\lambda'} = \{(1, 1, 1)\}$, $W'_{\lambda'} = \{(1)\} \subset \overline{\mathbb{K}}^1$, and $\gamma_{\lambda'} = 1$.

A symmetric representation of W would consist of $(\lambda, \mathscr{R}_\lambda)$ and $(\lambda', \mathscr{R}_{\lambda'})$, with $Z(\mathscr{R}_\lambda) = \{(2, 3)\}$ and $Z(\mathscr{R}_{\lambda'}) = \{(1)\}$.

Our main algorithm will have to deal with the following situation. As input, we will be given a representation of the set G in $\overline{\mathbb{K}}^\ell$; possibly, some points in G will not be in the open set O_λ (that is, may correspond to orbits having type λ', for some $\lambda' > \lambda$). As usual, the finite set G will be described by means of a zero-dimensional parametrization. Our goal will then be to compute a symmetric representation of $E^*_\lambda(G)$ when G is finite.

Example 16. Take $n = 3$, and again let $\lambda = (1^1 2^1)$, with $E_\lambda(\xi_{1,1}, \xi_{2,1}, \xi_{2,1}) = (\xi_{1,1}, \xi_{2,1})$. Assume we are given $G = \{(1, 1), (2, 3)\} \subset \overline{\mathbb{K}}^2$. In this case, $E^*_\lambda(G)$ is the set W seen in Examples 11 and 15, and the output we seek is a distinct coordinates representation of W, as discussed in Example 15.

Lemma 17. There exists a randomized algorithm $\mathsf{Decompose}(\lambda, \mathscr{R})$, which takes as input a partition λ of n with length ℓ and a zero-dimensional parametrization $\mathscr{R}$ of a set $G \subset \overline{\mathbb{K}}^\ell$ and returns a symmetric representation of $E^*_\lambda(G)$. The expected runtime is $O\tilde{~}(D^2 n)$ operations in $\mathbb{K}$, with $D = \deg(\mathscr{R}) = |G|$.

Proof. In the first step, we apply our algorithm $\mathsf{Type\_Of\_Fiber}$ from Lemma 12 where the input fiber is given not with coefficients in $\mathbb{K}$, but as the points described by $\mathscr{R}$. A general algorithmic principle, known as *dynamic evaluation*, allows us to do this as follows. Let $\mathscr{R} = ((q, v_1, \dots, v_\ell), \mu)$, with q and the v_i's in $\mathbb{K}[y]$. We then call $\mathsf{Type\_Of\_Fiber}$ with input coordinates $(v_1, \dots, v_\ell)$, and attempt to run the algorithm over the residue class ring $\mathbb{K}[y]/q$, as if q were irreducible.

If q is irreducible, $\mathbb{K}[y]/q$ is a field, and we encounter no problem. However, in general, $\mathbb{K}[y]/q$ is only a product of fields, so the algorithm may attempt to invert a zero-divisor. When this occurs, a "splitting" of the computation occurs. This amounts to discovering a non-trivial factorization of q. A direct solution then consists of running the algorithm again modulo the two factors that were discovered. Overall, this computes a sequence $(\mathscr{R}_i, \lambda_i, \boldsymbol{f}_i)_{1\le i\le N}$, where for $i = 1, \dots, N$,

(i) $\mathscr{R}_i = ((q_i, v_{i,1}, \dots, v_{i,\ell}), \mu_i)$ is a zero-dimensional parametrization that describes a set $F_i \subset F$. In addition F is the disjoint union of $F_1, \dots, F_N$;

(ii) λ_i is a partition of n, of length ℓ_i;

(iii) $\boldsymbol{f}_i$ is a sequence of ℓ_i elements with entries in the residue class ring $\mathbb{K}[y]/q_i$;

(iv) for any $\boldsymbol{\varepsilon}$ in F_i, corresponding to a root τ of q_i,

$$\mathsf{Type\_Of\_Fiber}(\lambda, \boldsymbol{\varepsilon}) = (\lambda_i, \boldsymbol{f}_i(\tau)).$$

Since $\mathsf{Type\_Of\_Fiber}$ takes time $O\tilde{~}(n)$, this process takes time $O\tilde{~}(D^2 n)$, with $D = \deg(\mathscr{R})$. The overhead $O\tilde{~}(D^2)$ is the penalty incurred by a straightforward application of dynamic evaluation techniques.

For $i = 1, \dots, N$, let $V_i = E_\lambda^{-1}(F_i)$, so that $W = \mathcal{S}_n(V)$ is the union of the orbits $W_i = \mathcal{S}_n(V_i)$. Then, from (iv) above we see that all points in W_i have type λ_i and that $(W_i)'_{\lambda_i}$ is the set $G_i = \{\boldsymbol{f}_i(\tau) \mid q_i(\tau) = 0\} \subset \overline{\mathbb{K}}^{\ell_i}$. Using the algorithm of [52, Proposition 1], we can compute a zero-dimensional parametrization $\mathscr{S}_i$ of G_i in time $O\tilde{~}(D_i^2 n)$, with $D_i = \deg(\mathscr{R}_i)$. The total cost is thus $O\tilde{~}(D^2 n)$.

The λ_i's may not be pairwise distinct. Up to changing indices, we may assume that $\lambda_1, \dots, \lambda_s$ are representatives of the pairwise distinct values among them. Then, for $i = 1, \dots, s$, we compute a zero-dimensional parametrization $\mathscr{T}_i$ that describes the union of those $Z(\mathscr{S}_j)$, for j such that $\lambda_j = \lambda_i$. Using algorithm [52, Lemma 3], this takes a total of $O\tilde{~}(D^2 n)$ operations in $\mathbb{K}$. Finally, we return $(\lambda_i, \mathscr{T}_i)_{1\le i\le s}$. □

3. $\mathcal{S}_\lambda$-equivariant polynomials: the Symmetrize algorithm

As noted previously the set of polynomials in our zero sets are globally invariant under the symmetric group, but its generators are not necessarily invariant. The goal of this section is to construct invariant generators. More precisely, if we let $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \dots\, n_r^{\ell_r})$ be a partition of n of length $\ell = \sum_{i=1}^r \ell_i$, then we will define $\mathcal{S}_\lambda$-*equivariant* systems of polynomials and give a detailed description of an algorithm, called $\mathsf{Symmetrize}$, that turns an $\mathcal{S}_\lambda$-equivariant system into one which is $\mathcal{S}_\lambda$-invariant. We recall that $\mathcal{S}_\lambda = \mathcal{S}_{\ell_1} \times \cdots \times \mathcal{S}_{\ell_r}$. We note that Hubert [40] also has an algorithm which symmetrizes polynomials constructed via a generating set of rational invariants. In our case we wish to avoid rational functions for our polynomial system solving as these will require case analysis for the zeros of the denominators (c.f. [41, Example 5.5]).

Consider a sequence of polynomials $\boldsymbol{q} = (q_1, \dots, q_\ell)$ in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$, where $\boldsymbol{Z}_i = (z_{i,1}, \dots, z_{i,\ell_i})$ is a set of ℓ_i variables. As mentioned, we also index $(\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r) =$

$(z_{1,1}, \dots, z_{r,\ell_r})$ as $(z_1, \dots, z_\ell)$. We say that $\boldsymbol{q}$ is $\mathcal{S}_\lambda$-*equivariant* if for any σ in $\mathcal{S}_\lambda$ and i in $\{1, \dots, \ell\}$, we have $\sigma(q_i) = q_{\sigma(i)}$, or equivalently

$$q_i(z_{\sigma(1)}, \dots, z_{\sigma(\ell)}) = q_{\sigma(i)}(z_1, \dots, z_\ell).$$

Here, we are implicitly seeing the elements of $\mathcal{S}_\lambda$ as permutations of $\{1, \dots, \ell\}$, as explained in Section 2.1.

In geometric terms, the zero-set $V(\boldsymbol{q}) \subset \overline{\mathbb{K}}^\ell$ of such a system is $\mathcal{S}_\lambda$-invariant, even though the equations themselves may not be invariant. In what follows, we describe how to derive equations $\boldsymbol{p} = (p_1, \dots, p_\ell)$ that generate the same ideal as $\boldsymbol{q}$ (in a suitable localization of $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$) and which are actually $\mathcal{S}_\lambda$-invariant. We will need an assumption, discussed below, that $z_i - z_j$ divides $q_i - q_j$ for all pairwise distinct indices i, j. We later show that this is always satisfied for our sets of critical points.

In order to construct a set of invariant generators we make use of *divided differences* of $\boldsymbol{q} = (q_1, \dots, q_\ell)$. These are defined as $q_{\{i\}} = q_i$ for i in $\{1, \dots, \ell\}$, and for each set of k distinct integers $I := \{i_1, \dots, i_k\} \subset \{1, \dots, \ell\}$, with $k \geq 2$,

$$q_I = \frac{q_{\{i_1,\dots,i_{r-1},i_{r+1},\dots,i_k\}} - q_{\{i_1,\dots,i_{q-1},i_{q+1},\dots,i_k\}}}{z_{i_r} - z_{i_q}}, \tag{9}$$

for any choice of i_r, i_q in I, with $i_r \neq i_q$. Indeed, it is known (see e.g., [25, Theorem 1]) that this defines q_I unambiguously (independently of the choice of i_r, i_q). A useful property of divided differences is the following:

(i) if $z_i - z_j$ divides $q_i - q_j$ for all $1 \leq i < j \leq \ell$, then q_I is a polynomial for all $I \subset \{1, \dots, \ell\}$.

The following proposition gives our construction of the polynomials $\boldsymbol{p}$. In what follows, for $i \geq 0$, $\eta_i(y_1, \dots, y_s)$ denotes the degree i elementary symmetric function in variables $y_1, \dots, y_s$. Define integers $\{\tau_k\}$ by $\tau_0 = 0$ and $\tau_k = \sum_{i=1}^k \ell_i$, for $k = 1, \dots, r$. Then any index i in $1, \dots, \ell$ can be written uniquely as $i = \tau_{k-1} + u$, for some k in $1, \dots, r$ and u in $1, \dots, \ell_k$. Thus, the indeterminates $z_{k,1}, \dots, z_{k,\ell_k}$ are numbered $z_{\tau_{k-1}+1}, \dots, z_{\tau_k}$, with $\tau_r = \ell$.

Proposition 18. Suppose the sequence $\boldsymbol{q} = (q_1, \dots, q_\ell)$ in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$ is $\mathcal{S}_\lambda$-equivariant and satisfies $z_i - z_j$ divides $q_i - q_j$ for $1 \leq i < j \leq \ell$. For $0 \leq k \leq r-1$ and $1 \leq j < \ell_{k+1}$, define

$$\begin{aligned} p_{\tau_{k+1}} &= \sum_{i=\tau_k+1}^{\tau_{k+1}} q_{\{i,\tau_{k+1}+1,\dots,\tau_r\}}, \\ p_{\tau_k+j} &= \sum_{s=1}^{j} \eta_{j-s}(z_{\tau_k+s+2}, \dots, z_{\tau_{k+1}}) \Big(\sum_{i=\tau_k+1}^{\tau_k+s} q_{\{i,\tau_k+s+1,\dots,\tau_r\}} \Big). \end{aligned}$$

Then the sequence

$$\boldsymbol{p} = \big(p_1, \dots, p_{\tau_1},\ p_{\tau_1+1}, \dots, p_{\tau_2},\ \dots, p_{\tau_{r-1}+1}, \dots, p_{\tau_r}\big)$$

is in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]^{\mathcal{S}_\lambda}$. If all q_i's have degree at most d, then $\deg(p_i) \leq d - \ell + i$ holds for $i = 1, \dots, \ell$. In particular, if $\ell \geq d + 2$, then $p_i = 0$ for all $i = 1, \dots, \ell - d - 1$.

The degree bound comes by inspection. We defer the rest of the proof (which follows by induction) to Appendix A. Our main idea is to use divided differences to reduce the factor $\prod_{i,j,i',j'}(z_{i,j} - z_{i',j'})$ and the elementary symmetric functions $\eta_{j-s}(\cdot)$ to add the missing monomials in a step by step fashion in order to obtain an invariant system.

Example 19. Let $\mathcal{S}_\lambda = \mathcal{S}_2 \times \mathcal{S}_1$. We take $\boldsymbol{q} = (q_1, q_2, q_3)$, where

$$\begin{aligned} q_1 &= z_2 z_3^2(z_1 + z_2 + 2z_3) + z_1 z_2 z_3^2, \\ q_2 &= z_1 z_3^2(z_1 + z_2 + 2z_3) + z_1 z_2 z_3^2, \\ q_3 &= z_1 z_2 z_3(z_1 + z_2 + 2z_3) + z_1 z_2 z_3^2. \end{aligned}$$

These polynomials satisfy both the equivariance property and the divisibility property. Using these divided differences and elementary symmetric functions, our procedure will produce the polynomials:

$$\begin{aligned} p_1 &= (z_1 + z_2 + 2z_3) z_3, \\ p_2 &= (z_1 + z_2 + 2z_3) z_2 z_3 + (z_1 + z_2 + 2z_3) z_1 z_3, \\ p_3 &= z_1 z_2 z_3(z_1 + z_2 + 2z_3) + z_1 z_2 z_3^2. \end{aligned}$$

The polynomials (p_1, p_2, p_3) are symmetric in (z_1, z_2) and (z_3), that is, are $\mathcal{S}_2 \times \mathcal{S}_1$-invariant. They also generate the same ideal as (q_1, q_2, q_3) in the localization ring $\mathbb{K}[z_1, z_2, z_3]_{(z_1 - z_2)(z_1 - z_3)(z_2 - z_3)}$.

We can also show that $\boldsymbol{q}$ can be written as a linear combination of $\boldsymbol{p}$, that is, we can find an $\ell \times \ell$ matrix polynomial $\mathbf{U}$ such that $\boldsymbol{p}\mathbf{U} = \boldsymbol{q}$. The construction of $\mathbf{U}$ proceeds as follows. Let $\mathbf{M}$ be the block-diagonal matrix with blocks $\mathbf{M}_1, \ldots, \mathbf{M}_r$ given by

$$\mathbf{M}_{k+1} = \begin{pmatrix} 1 & \eta_1(z_{\tau_k+3}, \ldots, z_{\tau_{k+1}}) & \eta_2(z_{\tau_k+3}, \ldots, z_{\tau_{k+1}}) & \cdots & \eta_{\ell_{k+1}-2}(z_{\tau_k+3}, \ldots, z_{\tau_{k+1}}) & 0 \\ 0 & 1 & \eta_1(z_{\tau_k+4}, \ldots, z_{\tau_{k+1}}) & \cdots & \eta_{\ell_{k+1}-3}(z_{\tau_k+4}, \ldots, z_{\tau_{k+1}}) & 0 \\ 0 & 0 & 1 & \cdots & \eta_{\ell_{k+1}-4}(z_{\tau_k+5}, \ldots, z_{\tau_{k+1}}) & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix},$$

for all $0 \le k \le r-1$. The matrices $\mathbf{M}_{k+1}$'s basically represent the elementary symmetric functions $\eta_{j-s}(\cdot)$ in the construction of $\boldsymbol{p}$. Here $\det(\mathbf{M}_{k+1}) = 1$ for all k, then $\det(\mathbf{M}) = 1$.

For a non-negative integer u, denote by $\mathbf{I}_u$ the identity matrix of size u and by $\mathbf{0}$ a zero matrix. The matrices $\mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ below encode the divided differences operators in the formulas of $\boldsymbol{p}$. For $k = 0, \ldots, r-1$ and $j = 1, \ldots, \ell_{k+1}$, we define the following $\tau_r \times \tau_r$ polynomial matrices. Set $\mathbf{B}_{\tau_0+1} = \mathbf{I}_{\tau_r}$, $\mathbf{C}_{\tau_0+1} = \mathbf{I}_{\tau_r}$, $\mathbf{D}_{\tau_0+j} = \mathbf{I}_{\tau_r}$, and

$$\mathbf{B}_{\tau_k+j} = \left(\begin{array}{c|c|c} \mathbf{I}_{\tau_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{E}_{k,j} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{\tau_r - \tau_{k+1}} \end{array}\right), \text{ with } \mathbf{E}_{k,j} = \left(\begin{array}{c|c|c} & z_{\tau_k+j} - z_{\tau_k+1} & \\ \mathbf{I}_{j-1} & \vdots & \mathbf{0} \\ & z_{\tau_k+j} - z_{\tau_k+j-1} & \\ \hline 0 \ \ldots \ 0 & -1 & \mathbf{0} \\ \hline \mathbf{0} & 0 & \mathbf{I}_{\ell_{k+1}-j} \end{array}\right),$$

$$\mathbf{C}_{\tau_k+j} = \left(\begin{array}{c|c|c} \mathbf{I}_{\tau_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{F}_{k,j} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{\tau_r-\tau_{k+1}} \end{array}\right), \text{ with } \mathbf{F}_{k,j} = \left(\begin{array}{c|c|c} \mathbf{diag}(z_{\tau_k+j} - z_{\tau_k+t})_{t=1}^{j-1} & \mathbf{0} & \mathbf{0} \\ \hline \frac{-1}{j} \ \cdots \ \frac{-1}{j} & \frac{-1}{j} & \mathbf{0} \\ \hline \mathbf{0} & 0 & \mathbf{I}_{\ell_{k+1}-j} \end{array}\right),$$

$$\mathbf{D}_{\tau_k+j} = \left(\begin{array}{c|c|c} \mathbf{diag}(z_{\tau_k+j} - z_t)_{t=1}^{\tau_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{G}_{k,j} & \mathbf{I}_{\ell_{k+1}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{\tau_r-\tau_{k+1}} \end{array}\right), \text{ with } \mathbf{G}_{k,j} \ : \ j^{th} \, \text{row is } (1,\ldots,1), \text{ rest zeros.}$$

We use the above matrices in the following.

Proposition 20. Suppose the sequence $\boldsymbol{q} = (q_1, \ldots, q_\ell)$ in $\mathbb{K}[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_r]^\ell$ satisfies the conditions of Proposition 18. Let $\Delta = \prod_{1 \le i < j \le \ell}(z_i - z_j)$ be the Vandermonde determinant associated with $z_1, \ldots, z_\ell$. Then the matrix $\mathbf{U}$ in $\mathbb{K}[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_r]^{\ell \times \ell}$, defined by

$$\mathbf{M} \cdot \mathbf{U} = \left(\prod_{k=0}^{r-1} \prod_{j=1}^{\ell_{k+1}} \mathbf{B}_{\tau_k+j} \, \mathbf{C}_{\tau_k+j} \, \mathbf{D}_{\tau_k+j} \right)$$

has determinant a unit in $\mathbb{K}[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_r, 1/\Delta]$ and satisfies $\boldsymbol{p}\mathbf{U} = \boldsymbol{q}$.

The proof of Proposition 20 follows by induction and is deferred to Appendix B.

Example 21. Consider again the polynomials $\boldsymbol{q} = (q_1, q_2, q_3)$ and $\boldsymbol{p} = (p_1, p_2, p_3)$ of Example 19. The matrix $\mathbf{U}$ which relates $\boldsymbol{p}$ to $\boldsymbol{q}$ is constructed as follows. For $k = 0$ and $j = 1, 2$ let

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{C}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{D}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{B}_2 = \begin{pmatrix} 1 & z_2 - z_1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{C}_2 = \begin{pmatrix} z_2 - z_1 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{D}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

while for $k = 1$ and $j = 1$ we have

$$\mathbf{B}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \mathbf{C}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \mathbf{D}_3 = \begin{pmatrix} z_3 - z_1 & 0 & 0 \\ 0 & z_3 - z_2 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

In the case $\lambda = (1^2\,2^1)$, $\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and hence

$$\mathbf{U} = (\mathbf{B}_1\mathbf{C}_1\mathbf{D}_1)(\mathbf{B}_2\mathbf{C}_2\mathbf{D}_2)(\mathbf{B}_3\mathbf{C}_3\mathbf{D}_3) = \begin{pmatrix} \frac{1}{2}(z_3 - z_1)(z_2 - z_1) & \frac{-1}{2}(z_2 - z_1)(z_3 - z_2) & 0 \\ \frac{1}{2}(z_3 - z_1) & \frac{1}{2}(z_3 - z_2) & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Note that $\det(\mathbf{U}) = \frac{1}{2}(z_3 - z_1)(z_3 - z_2)(z_2 - z_1)$.

The formulas defining $\boldsymbol{p}$ are straightforward to implement. The following proposition describes the resulting algorithm, called Symmetrize, and gives the cost of this procedure.

Proposition 22. There exists an algorithm $\mathsf{Symmetrize}(\lambda, \boldsymbol{q})$ which takes as input $\boldsymbol{q}$ as in Proposition 18 and a partition λ of n, and returns $\boldsymbol{p}$ as defined in that proposition. For $\boldsymbol{q}$ of degree at most d, the runtime is $O\tilde{~}(\ell^3\binom{\ell+d}{d})$ operations in $\mathbb{K}$.

The proof occupies the rest of this section. Write $\boldsymbol{q} = (q_1, \ldots, q_\ell)$, and recall the expressions defining $\boldsymbol{p} = (p_1, \ldots, p_\ell)$: for $k = 0, \ldots, r-1$, we have

$$p_{\tau_k + \ell_{k+1}} = \sum_{i=\tau_k+1}^{\tau_{k+1}} q_{\{i, \tau_{k+1}+1, \ldots, \tau_r\}}$$

and for $j = 1, \ldots, \ell_{k+1} - 1$,

$$p_{\tau_k + j} = \sum_{s=1}^{j} \eta_{j-s}(z_{\tau_k+s+2}, \ldots, z_{\tau_{k+1}})\Big(\sum_{i=1}^{s} q_{\{\tau_k+i, \tau_k+s+1, \ldots, \tau_r\}}\Big).$$

The main issue is to compute the divided differences $q_{\{\tau_k+i, \tau_k+s+1, \ldots, \tau_r\}}$ appearing in these expressions, for $k = 0, \ldots, r-1$ and $1 \le i \le s \le \ell_{k+1}$. Once this is done, the combinations necessary to obtain p_{τ_k+j} are easily carried out. The main ingredient in the proof is the following lemma which describes the computation of a single divided difference.

Lemma 23. There exists an algorithm $\mathsf{Divided\_Difference}(\boldsymbol{q}, I)$ that takes as input $\boldsymbol{q}$ as in Proposition 22 and a subset $I = \{i_1, \ldots, i_k\}$ of $\{1, \ldots, \ell\}$, and returns q_I. For $\boldsymbol{q}$ of degree at most d, the runtime is $O\tilde{~}(\ell\binom{\ell+d}{d})$ operations in $\mathbb{K}$.

Proof. For $j = 1, \ldots, k-1$, we claim that given $q_{\{i_1, \ldots, i_{j-1}\}}$, we can obtain $q_{\{i_1, \ldots, i_j\}}$ using $O\tilde{~}(\binom{\ell+d}{d})$ operations in $\mathbb{K}$.

To see this note that $q_{\{i_1, \ldots, i_{k-1}\}}$ has degree at most d. In order to compute $q_{\{i_1, \ldots, i_j\}}$, we use evaluation/interpolation. Choosing $\binom{\ell+d}{d}$ points as prescribed in [14], the algorithm given there allows us to compute the values of both numerator and denominator in (9) in $O\tilde{~}(\binom{\ell+d}{d})$ operations, then compute their ratio, and finally interpolate $q_{\{i_1, \ldots, i_j\}}$ in the same asymptotic runtime. The result then follows. □

Our Symmetrize algorithm now proceeds as follows. Apply algorithm Divided_Difference from Lemma 23 to all $[\tau_k+i, \tau_k+s+1, \dots, \tau_r]$, for $k = 0, \dots, r-1$ and $1 \le i \le s \le \ell_{k+1}$. There are $O(\ell^2)$ such indices, so this step takes $O\tilde{~}(\ell^3\binom{\ell+d}{d})$ operations in $\mathbb{K}$, allowing us to compute all sums $\sum_{i=1}^{s} q_{\{\tau_k+i,\tau_k+s+1,\dots,\tau_r\}}$ for the same asymptotic cost.

For $k = 0, \dots, r-1$, $j = 1, \dots, \ell_{k+1}-1$ and $s = 1, \dots, j$, we then determine the elementary symmetric polynomial $\eta_{j-s}(z_{\tau_k+s+2}, \dots, z_{\tau_{k+1}})$, which does not involve any arithmetic operations. We multiply it by the above sum, with cost $O\tilde{~}(\binom{\ell+d}{d})$, since the polynomials involved in the product have degree sum at most d and at most ℓ variables. Taking all indices k, j, s into account, this adds another $O\tilde{~}(\ell^3\binom{\ell+d}{d})$ steps to the total.

4. Algorithms for computing critical points

We can now turn to the main question in this article. Let $\boldsymbol{f} = (f_1, \dots, f_s)$ be polynomials in $\mathbb{K}[x_1, \dots, x_n]^{\mathcal{S}_n}$, with $s \le n$, and with $V = V(\boldsymbol{f}) \subset \overline{\mathbb{K}}^n$ denoting the algebraic set defined by $f_1 = \cdots = f_s = 0$. Given a polynomial ϕ in $\mathbb{K}[x_1, \dots, x_n]^{\mathcal{S}_n}$, we are interested in describing the algebraic set $W = W(\phi, \boldsymbol{f})$ defined by the simultaneous vanishing of the polynomials

$$f_1, \dots, f_s, \quad M_{s+1}(\mathrm{Jac}(\boldsymbol{f}, \phi))$$

where $M_{s+1}(\mathrm{Jac}(\boldsymbol{f}, \phi))$ is the set of $(s+1)$-minors of the Jacobian matrix $\mathrm{Jac}(\boldsymbol{f}, \phi) \in \mathbb{K}[x_1, \dots, x_n]^{(s+1)\times n}$.

4.1. Description of the algebraic set W

Sincce the algebraic set W is invariant under the action of the symmetric group by Corollary 7, the discussion in Section 2.3 applies to W. In particular, for a partition λ of n, the sets W_λ and W'_λ of (8) are well-defined. In what follows, we fix a partition $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2} \, \dots \, n_r^{\ell_r})$ of n and we let ℓ be its length; we explain how to compute a description of W'_λ along the lines of Section 2.3. For this, we let $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r$ be the indeterminates associated to λ, as defined in Section 2.1, with $\boldsymbol{Z}_i = (z_{i,1}, \dots, z_{i,\ell_i})$. As in that section, we also write all indeterminates $z_{1,1}, \dots, z_{r,\ell_r}$ as $z_1, \dots, z_\ell$.

Definition 24. With λ and $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r$ as above, we define $\mathbb{T}_\lambda$, the $\mathbb{K}$-algebra homomorphism $\mathbb{K}[x_1, \dots, x_n] \to \mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$ mapping $x_1, \dots, x_n$ to

$$\underbrace{z_{1,1}, \dots, z_{1,1}}_{n_1}, \ \dots, \ \underbrace{z_{1,\ell_1}, \dots, z_{1,\ell_1}}_{n_1}, \ \dots, \ \underbrace{z_{r,1}, \dots, z_{r,1}}_{n_r}, \ \dots, \ \underbrace{z_{r,\ell_r}, \dots, z_{r,\ell_r}}_{n_r}. \tag{10}$$

The operator $\mathbb{T}_\lambda$ extends to vectors or matrices of polynomials entry-wise.

We can now define

$$\boldsymbol{f}^{[\lambda]} = \mathbb{T}_\lambda(\boldsymbol{f}) = (f_1^{[\lambda]}, \dots, f_s^{[\lambda]}) \ \text{ and } \ \mathbf{J}^{[\lambda]} = \mathbb{T}_\lambda(\mathrm{Jac}(\boldsymbol{f}, \phi)) = \left[J_{i,j}^{[\lambda]}\right]_{1\le i\le s+1, 1\le j\le n}.$$

Notice that for f in $\mathbb{K}[x_1, \dots, x_n]^{\mathcal{S}_n}$, and for any indices j, k in $\{1, \dots, n\}$ for which $\mathbb{T}_\lambda(x_j) = \mathbb{T}_\lambda(x_k)$, we have

$$\mathbb{T}_\lambda\left(\frac{\partial f}{\partial x_j}\right) = \mathbb{T}_\lambda\left(\frac{\partial f}{\partial x_k}\right);$$

this follows by applying Lemma 6 to f and the transposition $(j\,k)$. Thus

$$\mathbb{T}_\lambda\left(\frac{\partial f}{\partial x_1},\dots,\frac{\partial f}{\partial x_n}\right) = (\underbrace{f^{[\lambda]}_{1,1},\dots,f^{[\lambda]}_{1,1}}_{n_1},\dots,\underbrace{f^{[\lambda]}_{1,\ell_1},\dots,f^{[\lambda]}_{1,\ell_1}}_{n_1},\dots,\underbrace{f^{[\lambda]}_{r,1},\dots,f^{[\lambda]}_{r,1}}_{n_r},\dots,\underbrace{f^{[\lambda]}_{r,\ell_r},\dots,f^{[\lambda]}_{r,\ell_r}}_{n_r}),$$

where $f^{[\lambda]}_{i,j}$ are polynomials in the variables $(\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_r)$. Consequently, we have the following.

Lemma 25. The columns of the transformed Jacobian matrix $\mathbf{J}^{[\lambda]}$ have the form:

$$\mathbf{J}^{[\lambda]} = (\underbrace{J^{[\lambda]}_{1,1},\dots,J^{[\lambda]}_{1,1}}_{n_1},\dots,\underbrace{J^{[\lambda]}_{1,\ell_1},\dots,J^{[\lambda]}_{1,\ell_1}}_{n_1},\dots,\underbrace{J^{[\lambda]}_{r,1},\dots,J^{[\lambda]}_{r,1}}_{n_r},\dots,\underbrace{J^{[\lambda]}_{r,\ell_r},\dots,J^{[\lambda]}_{r,\ell_r}}_{n_r}). \quad (11)$$

We will then let $\mathbf{G}^{[\lambda]} = [G^{[\lambda]}_{i,j}]_{1\le i\le s+1, 1\le j\le \ell}$ be the matrix with entries in $\mathbb{K}[\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_r]$ obtained from $\mathrm{Jac}(\boldsymbol{f},\phi)$ by first applying $\mathbb{T}_\lambda$ and then keeping only one representative among all repeated columns highlighted in the previous lemma.

Example 26. Let $s=1$ and $n=5$, so we consider two polynomials f_1,ϕ in $\mathbb{K}[x_1,\dots,x_5]$, and take $\lambda=(1^1\,2^2)$. Then

$$f_1^{[\lambda]}(z_{1,1},z_{2,1},z_{2,2}) = \mathbb{T}_\lambda(f_1) = f_1(z_{1,1},z_{2,1},z_{2,1},z_{2,2},z_{2,2}),$$

and

$$\mathbf{G}^{[\lambda]} = \begin{pmatrix} \mathbb{T}_\lambda(\frac{\partial f_1}{\partial x_1}) & \mathbb{T}_\lambda(\frac{\partial f_1}{\partial x_2}) & \mathbb{T}_\lambda(\frac{\partial f_1}{\partial x_4}) \\ \\ \mathbb{T}_\lambda(\frac{\partial \phi}{\partial x_1}) & \mathbb{T}_\lambda(\frac{\partial \phi}{\partial x_2}) & \mathbb{T}_\lambda(\frac{\partial \phi}{\partial x_4}) \end{pmatrix} \in \mathbb{K}[z_{1,1},z_{2,1},z_{2,2}]^{2\times 3}.$$

It is easy to see that the polynomials $\boldsymbol{f}^{[\lambda]}$ are $\mathcal{S}_\lambda$-invariant, where $\mathcal{S}_\lambda$ is the permutation group $\mathcal{S}_{\ell_1}\times\cdots\times\mathcal{S}_{\ell_r}$ introduced in the previous section. However, this is generally not the case for the entries of $\mathbf{G}^{[\lambda]}$.

Lemma 27. Let $\boldsymbol{g}^{[\lambda]} = (g^{[\lambda]}_1,\dots,g^{[\lambda]}_\ell)$ be a row of $\mathbf{G}^{[\lambda]}$. Then

(i) $z_i - z_j$ divides $g^{[\lambda]}_i - g^{[\lambda]}_j$ for $1\le i<j\le\ell$ and

(ii) $\boldsymbol{g}^{[\lambda]}$ is $\mathcal{S}_\lambda$-equivariant.

Proof. For the sake of definiteness, let us assume that $\boldsymbol{g}^{[\lambda]}$ is the row corresponding to the gradient of f_1, with the other cases treated similarly.

For statement (i), we start from indices i,j as in the lemma and let S be the $\mathbb{K}$-algebra homomorphism $\mathbb{K}[\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_r]\to\mathbb{K}[\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_r]$ that maps z_i to z_j, leaving all other variables unchanged. Let u,v in $\{1,\dots,n\}$ be indices such that $g^{[\lambda]}_i = \mathbb{T}_\lambda(\partial f_1/\partial x_u)$ and $g^{[\lambda]}_j = \mathbb{T}_\lambda(\partial f_1/\partial x_v)$ and $\sigma\in\mathcal{S}_n$ the transposition $(u\,v)$. From Lemma 6, we have that $\sigma(\partial f_1/\partial x_u) = \partial f_1/\partial x_v$ and applying $S\circ\mathbb{T}_\lambda$ gives $S(\mathbb{T}_\lambda(\sigma(\partial f_1/\partial x_u))) = S(\mathbb{T}_\lambda(\partial f_1/\partial x_v))$. For any $h\in\mathbb{K}[x_1,\dots,x_n]$ we have, by construction, $S(\mathbb{T}_\lambda(\sigma(h))) = S(\mathbb{T}_\lambda(h))$. Applying this on the left-hand side of the previous equality gives $S(g^{[\lambda]}_i) = S(g^{[\lambda]}_j)$. As a result, $z_i - z_j$ divides $g^{[\lambda]}_i - g^{[\lambda]}_j$, as claimed.

For statement (ii), we take indices k in $\{1,\dots,r\}$ and j,j' in $\{1,\dots,\ell_k\}$. We let $\sigma\in\mathcal{S}_\lambda$ be the transposition that maps (k,j) to (k,j') and prove that $\sigma(g^{[\lambda]}_{k,j}) = g^{[\lambda]}_{k,j'}$. As before, there exist indices u, v in $\{1, \dots, n\}$ such that $g_{k,j}^{[\lambda]} = \mathbb{T}_\lambda(\partial f_1/\partial x_u)$ and $g_{k,j'}^{[\lambda]} = \mathbb{T}_\lambda(\partial f_1/\partial x_v)$. Without loss of generality, assume that u and v are the smallest such indices. Then $\mathbb{T}_\lambda$ maps $x_u, \dots, x_{u+\ell_k-1}$ to $z_{k,j}$ and $x_v, \dots, x_{v+\ell_k-1}$ to $z_{k,j'}$.

Let $\tau \in \mathcal{S}_n$ be permutation that permutes $(u, \dots, u+\ell_k-1)$ with $(v, \dots, v+\ell_k-1)$. From Lemma 6, we get $\tau(\partial f_1/\partial x_v) = \partial f_1/\partial x_u$. Then $\mathbb{T}_\lambda(\tau(\partial f_1/\partial x_u)) = \mathbb{T}_\lambda(\partial f_1/\partial x_v) = g_{k,j'}^{[\lambda]}$. By the construction, the left-hand side is equal to $\sigma(\mathbb{T}_\lambda(\partial f_1/\partial x_u))$, that is, $\sigma(g_{k,j}^{[\lambda]})$. □

Lemma 27 implies that we can apply Algorithm Symmetrize from Section 3 to each row of $\mathbf{G}^{[\lambda]}$. The result is a polynomial matrix $\mathbf{H}^{[\lambda]}$ in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$, whose rows are all $\mathcal{S}_\lambda$-invariant, and such that $\mathbf{H}^{[\lambda]}\mathbf{U}^{[\lambda]} = \mathbf{G}^{[\lambda]}$, for some polynomial matrix $\mathbf{U}^{[\lambda]}$ in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]^{\ell \times \ell}$. Applying Algorithm Symmetric_Coordinates from Lemma 9 to the entries of both $\boldsymbol{f}^{[\lambda]}$ and $\mathbf{H}^{[\lambda]}$ gives polynomials $\bar{\boldsymbol{f}}^{[\lambda]}$ and a matrix $\bar{\mathbf{H}}^{[\lambda]}$, all with entries in $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r]$, with variables $\boldsymbol{e}_i = e_{i,1}, \dots, e_{i,\ell_1}$ for all i, and such that $\boldsymbol{f}^{[\lambda]} = \bar{\boldsymbol{f}}^{[\lambda]}(\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_r)$ and $\mathbf{H}^{[\lambda]} = \bar{\mathbf{H}}^{[\lambda]}(\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_r)$.

The following summarizes the main properties of this construction. For the definitions of the sets $\mathcal{C}_\lambda$, $\mathcal{C}_\lambda^{\text{strict}}$, the mapping E_λ and the open set $O_\lambda \subset \overline{\mathbb{K}}^\ell$, see Section 2.3.

Proposition 28. Let λ be a partition of n of length ℓ.

(i) If $\ell \le s$, then $E_\lambda(W \cap \mathcal{C}_\lambda)$ is the zero-set of $\bar{\boldsymbol{f}}^{[\lambda]}$ in $\overline{\mathbb{K}}^\ell$.

(ii) If $\ell > s$, then $W'_\lambda = E_\lambda(W \cap \mathcal{C}_\lambda^{\text{strict}})$ is the zero-set of $\bar{\boldsymbol{f}}^{[\lambda]}$ and all $(s+1)$-minors of $\bar{\mathbf{H}}^{[\lambda]}$ in $O_\lambda \subset \overline{\mathbb{K}}^\ell$.

Proof. Let $\boldsymbol{\xi}$ be in the set $\mathcal{C}_\lambda$ defined in Section 2.3, and write

$$\boldsymbol{\xi} = \big(\underbrace{\xi_{1,1}, \dots, \xi_{1,1}}_{n_1}, \dots, \underbrace{\xi_{1,\ell_1}, \dots, \xi_{1,\ell_1}}_{n_1}, \dots, \underbrace{\xi_{r,1}, \dots, \xi_{r,1}}_{n_r}, \dots, \underbrace{\xi_{r,\ell_r}, \dots, \xi_{r,\ell_r}}_{n_r}\big).$$

Set $\boldsymbol{\zeta} = (\xi_{1,1}, \xi_{1,2}, \dots, \xi_{r,\ell_r}) \in \overline{\mathbb{K}}^\ell$ and $\boldsymbol{\varepsilon} = E_\lambda(\boldsymbol{\xi}) \in \overline{\mathbb{K}}^\ell$. By definition, we have $\boldsymbol{f}(\boldsymbol{\xi}) = \boldsymbol{f}^{[\lambda]}(\boldsymbol{\zeta})$ and $\text{Jac}(\boldsymbol{f}, \phi)(\boldsymbol{\xi}) = \mathbf{J}^{[\lambda]}(\boldsymbol{\zeta})$. Thus, $\boldsymbol{\xi}$ is in $W \cap \mathcal{C}_\lambda$ if and only if it cancels $\boldsymbol{f}$ and $\text{Jac}(\boldsymbol{f}, \phi)$ has rank at most s at $\boldsymbol{\xi}$, that is, if $\boldsymbol{f}^{[\lambda]}(\boldsymbol{\zeta}) = 0$ and $\mathbf{J}^{[\lambda]}(\boldsymbol{\zeta})$ has rank at most s. The point $\boldsymbol{\xi}$ is in $W \cap \mathcal{C}_\lambda^{\text{strict}}$ if all the entries of $\boldsymbol{\zeta}$ are also pairwise distinct.

In addition, we have $\boldsymbol{f}^{[\lambda]}(\boldsymbol{\zeta}) = \bar{\boldsymbol{f}}^{[\lambda]}(\boldsymbol{\varepsilon})$ and, by construction,

$$\text{rank}(\mathbf{J}^{[\lambda]}(\boldsymbol{\zeta})) = \text{rank}(\mathbf{G}^{[\lambda]}(\boldsymbol{\zeta})).$$

If $\ell \le s$ then, since $\mathbf{G}^{[\lambda]}$ has ℓ columns, we see that $\boldsymbol{\xi}$ is in $W \cap \mathcal{C}_\lambda$ if and only if $\boldsymbol{\varepsilon} = E_\lambda(\boldsymbol{\xi})$ cancels $\bar{\boldsymbol{f}}^{[\lambda]}$. Since $E_\lambda : \mathcal{C}_\lambda \to \overline{\mathbb{K}}^\ell$ is onto, this implies our first claim.

Suppose further that $\boldsymbol{\xi}$ is in $\mathcal{C}_\lambda^{\text{strict}}$, so that $\boldsymbol{\varepsilon}$ is in O_λ. From Proposition 20, we have $\mathbf{H}^{[\lambda]}\mathbf{U}^{[\lambda]} = \mathbf{G}^{[\lambda]}$. Our assumption on $\boldsymbol{\xi}$ implies that $\mathbf{U}^{[\lambda]}(\boldsymbol{\zeta})$ is invertible, so that $\mathbf{G}^{[\lambda]}$ and $\mathbf{H}^{[\lambda]}$ have the same rank at $\boldsymbol{\zeta}$. Finally, we have $\mathbf{H}^{[\lambda]}(\boldsymbol{\zeta}) = \bar{\mathbf{H}}^{[\lambda]}(\boldsymbol{\varepsilon})$. All this combined shows that $\boldsymbol{\xi}$ is in $W'_\lambda = E_\lambda(W \cap \mathcal{C}_\lambda^{\text{strict}})$ if and only if $\boldsymbol{\varepsilon} = E_\lambda(\boldsymbol{\xi})$ cancels $\bar{\boldsymbol{f}}^{[\lambda]}$ and all $(s+1)$-minors of $\bar{\mathbf{H}}^{[\lambda]}$. Since the restriction $E_\lambda : \mathcal{C}_\lambda^{\text{strict}} \to O_\lambda$ is onto, this implies the second claim. □

4.2. *The* Critical_Points_Per_Orbit *algorithm*

The main algorithm of this paper is Critical_Points_Per_Orbit which takes as input symmetric $\boldsymbol{f} = (f_1, \dots, f_s)$ and ϕ in $\mathbb{K}[x_1, \dots, x_n]$ and, if finite, outputs a symmetric representation of the critical point set $W = W(\phi, \boldsymbol{f})$. Using our notation from Section 2, this means that we want to compute zero-dimensional parametrizations of $W'_\lambda = E_\lambda(W \cap$

$\mathcal{C}_\lambda^{\mathrm{strict}}$), for all partitions λ of n for which this set is not empty. The algorithm is based on Proposition 28, with a minor modification, as we will see that it is enough to consider partitions of n of length ℓ either exactly equal to s, or at least $s+1$.

For any partition λ, we first need to transform $\boldsymbol{f}$ and ϕ, in order to obtain the polynomials in Proposition 28.

Lemma 29. There exists an algorithm $\mathsf{Prepare\_F}(\boldsymbol{f}, \lambda)$ which takes as input $\boldsymbol{f}$ as above and a partition λ, and returns $\bar{\boldsymbol{f}}^{[\lambda]}$. If $\boldsymbol{f}$ has degree at most d, the algorithm takes $O\tilde{~}(n\binom{n+d}{d}^2)$ operations in $\mathbb{K}$. Similarly, there exists an algorithm $\mathsf{Prepare\_F\_H}(\boldsymbol{f}, \phi, \lambda)$ which takes as input $\boldsymbol{f}, \phi$ as above and a partition λ, and returns $\bar{\boldsymbol{f}}^{[\lambda]}$ and $\bar{\mathbf{H}}^{[\lambda]}$. If $\boldsymbol{f}$ and ϕ have degree at most d, then the algorithm takes $O\tilde{~}(n^4\binom{n+d}{d}^2)$ operations in $\mathbb{K}$.

Proof. In the first case, applying $\mathbb{T}_\lambda$ to $\boldsymbol{f}$ takes linear time in the number of monomials $O(n\binom{n+d}{d})$ and gives us $\boldsymbol{f}^{[\lambda]}$. We then invoke $\mathsf{Symmetric\_Coordinates}$ $(\lambda, \boldsymbol{f}^{[\lambda]})$, using Lemma 9, in order to obtain $\bar{\boldsymbol{f}}^{[\lambda]}$ with the cost being $O\tilde{~}(n\binom{n+d}{d}^2)$ operations in $\mathbb{K}$.

In the second case, we obtain $\boldsymbol{f}^{[\lambda]}$ as above. We also compute the matrix $\mathrm{Jac}(\boldsymbol{f}, \phi)$, which takes $O(n^2\binom{n+d}{d})$ operations. For the same cost, we apply $\mathbb{T}_\lambda$ to all its entries and remove redundant columns, as specified in Lemma 25, so as to yield the matrix $\mathbf{G}^{[\lambda]}$. We then apply Algorithm $\mathsf{Symmetrize}$ from Proposition 22 to all rows of $\mathbf{G}^{[\lambda]}$, which takes $O\tilde{~}(n^4\binom{n+d}{d})$ operations, and returns $\mathbf{H}^{[\lambda]}$. Finally, we apply $\mathsf{Symmetric\_Coordinates}$ to all entries of this matrix which gives $\bar{\mathbf{H}}^{[\lambda]}$ and takes $O\tilde{~}(n^2\binom{n+d}{d}^2)$ operations in $\mathbb{K}$. □

At the core of the algorithm, we need a procedure for finding isolated solutions of certain polynomial systems. In our main algorithm, we solve such systems using procedures called $\mathsf{Isolated\_Points}(\boldsymbol{g})$ and $\mathsf{Isolated\_Points}(\boldsymbol{g}, \mathbf{H}, k)$. Given polynomials $\boldsymbol{g}$, the former returns a zero-dimensional parametrization of the isolated points of $V(\boldsymbol{g})$. The latter takes as input polynomials $\boldsymbol{g}$, a polynomial matrix $\mathbf{H}$ and an integer k, and returns a zero-dimensional parametrization of the isolated points of $V(\boldsymbol{g}, M_k(\mathbf{H}))$, where $M_k(\mathbf{H})$ denotes the set of k-minors of $\mathbf{H}$ (note that the former procedure can be seen as a particular case of the latter, where we take $\mathbf{H}$ to be a matrix with no row and $k=-1$). To establish correctness of the main algorithm, any implementation of these procedures is suitable.

Apart from the subroutines discussed above and the function $\mathsf{Decompose}$ from Lemma 17, our algorithm also requires a procedure $\mathsf{Remove\_Duplicates}(S)$. This inputs a list $S = (\lambda_i, \mathscr{R}_i)_{1\le i\le N}$, where each λ_i is a partition of n and $\mathscr{R}_i$ a zero-dimensional parametrization. As all λ_i's may not be distinct in this list, this procedure removes pairs $(\lambda_i, \mathscr{R}_i)$ from S so as to ensure that all resulting partitions are pairwise distinct (the choice of which entries to remove is arbitrary; it does not affect correctness of the overall algorithm).

Proposition 30. Algorithm $\mathsf{Critical\_Points\_Per\_Orbit}$ is correct.

Proof. The goal of the algorithm is to compute zero-dimensional representations of $W'_\lambda = E_\lambda(W \cap \mathcal{C}_\lambda^{\mathrm{strict}})$ for all partitions λ of n for which this set is not empty.

To understand the first loop, recall first that W is assumed to be finite. Hence this also holds for all $W \cap \mathcal{C}_\lambda$, and thus for all $E_\lambda(W \cap \mathcal{C}_\lambda)$. As a result, for λ of length s, Proposition 28(i) implies that at Step 2b , $\mathsf{Isolated\_Points}(\bar{\boldsymbol{f}}_\lambda)$ returns a zero-dimensional parametrization of $G := E_\lambda(W \cap \mathcal{C}_\lambda)$. Then, we recall from Lemma 17 that the output of $\mathsf{Decompose}(\lambda, \mathscr{R}_\lambda)$ is a symmetric representation of $E_\lambda^*(G)$. Note that the latter set is

Algorithm 1 Critical_Points_Per_Orbit($\boldsymbol{f}, \phi$)

```
Input: $\boldsymbol{f} = (f_1, \dots, f_s)$ and $\phi$ in $\mathbb{K}[x_1, \dots, x_n]^{\mathcal{S}_n}$ such that $W = W(\phi, V(\boldsymbol{f}))$ is finite.

Output: A symmetric representation of $W$.
  (1) $S = [\,]$
  (2) For $\lambda \vdash n$ of length $s$
        (a) $\bar{\boldsymbol{f}}^{[\lambda]} = $ Prepare_F($\boldsymbol{f}, \lambda$)
        (b) $\mathscr{R}_\lambda = $ Isolated_Points($\bar{\boldsymbol{f}}^{[\lambda]}$)
        (c) append the output of Decompose($\lambda, \mathscr{R}_\lambda$) to $S$
  (3) For any partition $\lambda$ of $n$ of length in $\{s+1, \dots, n\}$
        (a) $\bar{\boldsymbol{f}}^{[\lambda]}, \bar{\mathbf{H}}^{[\lambda]} = $ Prepare_F_H($\boldsymbol{f}, \phi, \lambda$)
        (b) $\mathscr{R}_\lambda = $ Isolated_Points($\bar{\boldsymbol{f}}^{[\lambda]}, \bar{\mathbf{H}}^{[\lambda]}, s+1$)
        (c) $(\lambda_i, \mathscr{R}_i)_{1 \le i \le N} = $ Decompose($\lambda, \mathscr{R}_\lambda$)
        (d) append $(\lambda_{i_0}, \mathscr{R}_{i_0})$ to $S$, where $i_0$ is such that $\lambda_{i_0} = \lambda$, if such an $i_0$ exists
  (4) Return Remove_Duplicates($S$)
```

the orbit of $W \cap \mathcal{C}_\lambda$, that is, the set of all orbits contained in W whose type λ' satisfies $\lambda' \ge \lambda$. Taking into account all partitions λ of length s, the set of partitions $\lambda' \ge \lambda$ covers all partitions of length $\ell \in \{1, \dots, s\}$, so that at the end of Step 2, we have zero-dimensional parametrizations of W'_λ for all partitions of length $\ell \in \{1, \dots, s\}$ (with possible repetitions). Calling Remove_Duplicates(S) will remove any duplicates among this list.

The second loop deals with partitions λ of length at least $s+1$. Since we assume that W is finite, W'_λ is finite for any such λ. Proposition 28(ii) then implies that the points in W'_λ are isolated points of the zero-set of $\bar{\boldsymbol{f}}^{[\lambda]}$ and of the $(s+1)$-minors of $\bar{\mathbf{H}}^{[\lambda]}$. As a result, W'_λ is a subset of $Z(\mathscr{R}_\lambda)$, for $\mathscr{R}_\lambda$ computed in Step 3b with all other points in $Z(\mathscr{R}_\lambda)$ corresponding to points in W with type $\lambda' > \lambda$. In particular, after the call to Decompose, it suffices to keep the entry in the list corresponding to the partition λ, to obtain a description of W'_λ. □

5. Cost of the Critical_Points_Per_Orbit Algorithm

In this section we provide a complexity analysis of our Critical_Points_Per_Orbit algorithm and hence also complete the proof of Theorem 5.

At the core of the Critical_Points_Per_Orbit algorithm is a procedure, Isolated_Points. Recall that on input polynomials $\boldsymbol{g}$, a polynomial matrix $\mathbf{H}$ and an integer k, it returns a zero-dimensional parametrization of the isolated points of $V(\boldsymbol{g}, M_k(\mathbf{H}))$, where $M_k(\mathbf{H})$ denotes the set of k-minors of $\mathbf{H}$. We apply this procedure to polynomials with entries in $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r] = \mathbb{K}[e_{1,1}, \dots, e_{1,\ell_1}, e_{2,1}, \dots, e_{2,\ell_2}, \dots, e_{r,1}, \dots, e_{r,\ell_r}]$.

Rather than using classical methods for solving these polynomial systems, we use the *symbolic homotopy method for weighted domains* given in [45], as this algorithm is well suited to handle a weighted-degree structure exhibited by such systems. Indeed, the polynomial ring, $\mathbb{K}[\boldsymbol{e}_1, \dots, \boldsymbol{e}_r]$, arising from an orbit parameter by λ is obtained through a correspondence between the variable $e_{i,k}$ and the elementary symmetric polynomial $\eta_{i,k}(x_{j_1}, \dots, x_{j_m})$, for certain indices $j_1, \dots, j_m$. More precisely, for any f in

$\mathbb{K}[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_r]^{\mathcal{S}_\lambda}$, let $\bar{f}$ be the polynomial in $\mathbb{K}[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r]$ satisfying

$$f(\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_r) = \bar{f}(\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_r),$$

with $\boldsymbol{\eta}_i = (\eta_{i,1}, \ldots, \eta_{i,\ell_i})$ for all i. Since each $\eta_{i,k}$ has degree k, it is natural to assign a weight k to variable $e_{i,k}$, so that the weighted degree of $\bar{f}$ equals the degree of f. Our vector of variable weights is then $\boldsymbol{w} = (1, \ldots, \ell_1, 1, \ldots, \ell_2, \ldots, 1, \ldots, \ell_r)$.

5.1. Solving weighted determinantal systems

In this section, we briefly review the algorithm for solving determinantal systems over a ring of weighted polynomials.

Suppose we work with polynomials in $\mathbb{K}[\boldsymbol{Y}] = \mathbb{K}[y_1, \ldots, y_m]$, where each variable y_i has weight $w_i \geq 1$ (denoted by $\mathrm{wdeg}(y_i) = w_i$). The weighted degree of a monomial $y_1^{\alpha_1} \cdots y_m^{\alpha_m}$ is then $\sum_{i=1}^m w_i \alpha_i$, and the weighted degree of a polynomial is the maximum of the weighted degree of its terms with non-zero coefficients. The *weighted column degrees* of a polynomial matrix is the sequence of the weighted degrees of its columns, where the weighted degree of a column is simply the maximum of the weighted degrees of its entries.

Let $\boldsymbol{f} = (f_1, \ldots, f_\tau)$ be a sequence of polynomials in $\mathbb{K}[\boldsymbol{Y}]$ and $\mathbf{G} = [g_{i,j}] \in \mathbb{K}[\boldsymbol{Y}]^{p \times q}$ a matrix of polynomials such that $p \leq q$ and $m = q - p + \tau + 1$, and let $V_p(\mathbf{G}, \boldsymbol{f})$ denote the set of points in $\overline{\mathbb{K}}$ at which all polynomials in $\boldsymbol{f}$ and all p-minors of $\mathbf{G}$ vanish. In [45], a symbolic homotopy algorithm for weighted domains is presented which constructs a symbolic homotopy from a generic start system to the system defining $V_p(\mathbf{G}, \boldsymbol{f})$ and then uses this to efficiently determine the isolated points of $V_p(\mathbf{G}, \boldsymbol{f})$. The main theorem of [45], in the special case of weighted polynomial rings, is given in terms of a number of parameters.

Let $(\gamma_1, \ldots, \gamma_\tau)$ be the weighted degrees of $(f_1, \ldots, f_\tau)$, let $(\delta_1, \ldots, \delta_q)$ be the weighted column degrees of $\mathbf{G}$, let d be the maximum of the degrees (in the usual sense) of all $\boldsymbol{f}, \mathbf{G}$ and set

$$\Gamma = m^2 \binom{m+d}{m} + m^4 \binom{q}{p}.$$

The following quantities are all related to the degrees of some geometric objects present in the algorithm. We define

$$c = \frac{\gamma_1 \cdots \gamma_\tau \ \cdot \eta_{m-\tau}(\delta_1, \ldots, \delta_q)}{w_1 \cdots w_m} \quad \text{and}$$
$$e = \frac{(\gamma_1 + 1) \cdots (\gamma_\tau + 1) \cdot \eta_{m-\tau}(\delta_1 + 1, \ldots, \delta_q + 1)}{w_1 \cdots w_m},$$

where η_{n-s} is the elementary symmetric polynomial of degree $n - s$. For a subset $\boldsymbol{i} = \{i_1, \ldots, i_{m-\tau}\} \subset \{1, \ldots, q\}$, we further let $(d_{\boldsymbol{i},1}, \ldots, d_{\boldsymbol{i},m})$ denote the sequence obtained by sorting $(\gamma_1, \ldots, \gamma_\tau, \delta_{i_1}, \ldots, \delta_{i_{m-\tau}})$ in non-decreasing order, and we write

$$\kappa_{\boldsymbol{i}} = \max_{1 \leq k \leq m} (d_{\boldsymbol{i},1} \cdots d_{\boldsymbol{i},k} w_{k+1} \cdots w_m) \quad \text{and} \quad \kappa = \sum_{\boldsymbol{i} = \{i_1, \ldots, i_{m-\tau}\} \subset \{1, \ldots, q\}} \kappa_{\boldsymbol{i}}. \tag{12}$$

Note that without loss of generality, in these equations, we may also assume that the weights $w_1, \ldots, w_m$ are reordered to form a non-decreasing sequence.

Theorem 31. [45, Theorem 5.3] Let $\mathbf{G}$ be a matrix in $\mathbb{K}[\mathbf{Y}]^{p\times q}$ and $\boldsymbol{f} = (f_1, \ldots, f_\tau)$ be polynomials in $\mathbb{K}[\mathbf{Y}]$ such that $p \leq q$ and $m = q - p + \tau + 1$. There exists a randomized algorithm which takes $\mathbf{G}$ and $\boldsymbol{f}$ as input and computes a zero-dimensional parametrization of these isolated solutions using

$$O\tilde{}\Big(\big(c(e + c^5) + d^2 \,(\frac{\kappa}{w_1 \cdots w_m})^2\big)m^4\Gamma\Big)$$

operations in $\mathbb{K}$. Moreover, the number of solutions in the output is at most c.

When there is no matrix $\mathbf{G}$, so $\tau = m$, then the runtimes reported above remain the same with the term Γ becoming $\Gamma = m^2\binom{m+d}{m}$. In this case, the term κ is simply equal to $\kappa = \max_{1\leq k\leq m}(\gamma_1 \cdots \gamma_k w_{k+1} \cdots w_m)$, assuming that the degrees $\gamma_1, \ldots, \gamma_k$ are given in non-decreasing order.

We finish this subsection with an observation in those cases with $m > q - p + \tau + 1$.

Remark 32. When $m > q - p + \tau + 1$, there are no isolated points in $V_p(\mathbf{G}, \boldsymbol{f})$. Indeed if we let $I \subset \overline{\mathbb{K}}[\mathbf{Y}]$ be the ideal generated by the p-minors of $\mathbf{G}$ then a result due to Eagon and Northcott [18, Section 6] implies that all irreducible components of $V(I)$ have codimension at most $q - p + 1$. By Krull's theorem the irreducible components of $V_p(\mathbf{G}, \boldsymbol{f}) = V(I + \langle f_1, \ldots, f_\tau\rangle)$ then have codimension at most $q-p+1+\tau$. This implies that the irreducible components of $V_p(\mathbf{G}, \boldsymbol{f})$ in $\overline{\mathbb{K}}^m$ have dimension at least $m - (q - p + \tau + 1)$, which is positive when $m > q - p + \tau + 1$.

5.2. *The complexity of the* Isolated_Points *procedure*

Estimating the runtimes for the Isolated_Points algorithms follows from Theorem 31, for the weighted domains associated to various partitions of n. Therefore we let $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \ldots\, n_r^{\ell_r})$ be a partition of length ℓ, with $\ell \geq s$.

The parameters that appear in Theorem 31 can be determined as follows. The weights of variables $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r)$ are $\boldsymbol{w} = (1, \ldots, \ell_1, \ldots, 1, \ldots, \ell_r)$. For $i = 1, \ldots, s$, the weighted degree of $\bar{f}_i^{[\lambda]}$ is the same as the degree of $f_i^{[\lambda]}$ and so is at most d.

For $j = 1, \ldots, \ell$, the weighted column degree of the j-th column of $\bar{\mathbf{H}}^{[\lambda]}$ is at most $\delta_j = d - 1 - \ell + j$ (note that all entries of the Jacobian matrix of $\boldsymbol{f}, \phi$ have degree at most $d - 1$; then apply Proposition 18). In particular, if $\ell > d$, then all entries on the j-th column of $\bar{\mathbf{H}}^{[\lambda]}$ equal zero for $j = 1, \ldots, \ell - d$. Finally, in what follows, we let

$$\Gamma = n^2\binom{n+d}{d} + n^4\binom{n}{s+1}.$$

5.2.1. *Partitions of length s.*

We recall that when the length ℓ of the partition λ equals s, we do not need to deal with a matrix $\bar{\mathbf{H}}^{[\lambda]}$. In this situation, one only needs to compute the isolated points of $V(\bar{\boldsymbol{f}}^{[\lambda]})$.

Consider such a partition $\lambda = (n_1^{\ell_1}\, n_2^{\ell_2}\, \ldots\, n_r^{\ell_r})$ of n and the corresponding variables $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r)$, with $\mathrm{wdeg}(e_{i,k}) = k$ for all $i = 1, \ldots, r$ and $k = 1, \ldots, \ell_i$. We make the following claim: *if there exists i such that $\ell_i > d$, then there is no isolated point in* $V(\bar{\boldsymbol{f}}^{[\lambda]})$. Indeed, in such a case, variable e_{i,ℓ_i} does not appear in $\bar{\boldsymbol{f}}^{[\lambda]}$, for weighted degree reasons, so that the zero-set of this system is invariant with respect to translations along the e_{i,ℓ_i} axis. In particular, it admits no isolated solution.

Therefore we can suppose that all ℓ_i's are at most d. In this case, the quantities c, e, κ used in Theorem 31 become respectively

$$\mathfrak{c}_\lambda = \frac{d^s}{w_\lambda}, \quad \mathfrak{e}_\lambda = \frac{n(d+1)^s}{w_\lambda}, \quad \kappa_\lambda = d^s = w_\lambda \mathfrak{c}_\lambda,$$

with $w_\lambda = \ell_1! \cdots \ell_r!$. In this case Theorem 31 implies that $V(\bar{\boldsymbol{f}}^{[\lambda]})$ contains at most $\mathfrak{c}_\lambda$ isolated points, and that and one can compute all of them using

$$O\tilde{~}\left(\left(\mathfrak{c}_\lambda(\mathfrak{e}_\lambda + \mathfrak{c}_\lambda^5) + d^2\mathfrak{c}_\lambda^2\right)n^4\Gamma_\lambda\right) \subset O\tilde{~}\left(d^2\mathfrak{c}_\lambda(\mathfrak{e}_\lambda + \mathfrak{c}_\lambda^5)n^4\Gamma\right)$$

operations in $\mathbb{K}$.

5.2.2. Partitions of length greater than s.

For a partition λ of length ℓ greater than s, we have to take into account the minors of the matrix $\bar{\mathbf{H}}^{[\lambda]}$. Note that the assumptions of Theorem 31 are satisfied: the matrix $\bar{\mathbf{H}}^{[\lambda]}$ is in $\mathbb{K}[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r]^{(s+1)\times\ell}$, with $\ell \geq s+1$, and we have s equations $\bar{\boldsymbol{f}}^{[\lambda]}$ in $\mathbb{K}[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r]$, so the number of variables ℓ does indeed satisfy $\ell = \ell - (s+1) + s + 1$.

We claim that if $\ell > d$, then the algebraic set $V_{s+1}(\bar{\mathbf{H}}^{[\lambda]}, \bar{\boldsymbol{f}}^{[\lambda]})$ does not have any isolated points. Indeed, in this case, we pointed out above that the columns of indices 1 to $\ell - d$ in $\bar{\mathbf{H}}^{[\lambda]}$ are identically zero. After discarding these zero-columns from $\bar{\mathbf{H}}^{[\lambda]}$, we obtain a matrix $\mathbf{L}^{[\lambda]}$ of dimension $(s+1)\times d$ such that $V_{s+1}(\bar{\mathbf{H}}^{[\lambda]}, \bar{\boldsymbol{f}}^{[\lambda]}) = V_{s+1}(\mathbf{L}^{[\lambda]}, \bar{\boldsymbol{f}}^{[\lambda]})$, and using Remark 32 with $p = s+1, q = d, \tau = s$, and $m \geq \ell$ shows that this algebraic set has no isolated points.

Thus, let us now assume that $\ell \leq d$. The matrix $\bar{\mathbf{H}}^{[\lambda]}$ has weighted column degrees $(\delta_1, \ldots, \delta_\ell) = (d - \ell, \ldots, d-1)$, whereas the weighted degrees of all polynomials in $\bar{\boldsymbol{f}}^{[\lambda]}$ are at most d. To estimate the runtime of the $\mathsf{Isolated\_Points}(\bar{\mathbf{H}}^{[\lambda]}, \bar{\boldsymbol{f}}^{[\lambda]})$, we will need the following property.

Lemma 33. Let κ be defined as in (12) with $m = \ell, \tau = s, p = s+1, q = \ell, (\delta_1, \ldots, \delta_\ell) = (d-1-\ell, \ldots, d-1)$, and $(\gamma_1, \ldots, \gamma_s) = (d, \ldots, d)$. Then, for partitions of length ℓ at most d, one has

$$\kappa = d^s \eta_{\ell-s}(d-1, \ldots, d-\ell).$$

Proof. Without loss of generality, we reorder the weights $\boldsymbol{w}$ as $\boldsymbol{w}' = (w'_1, \ldots, w'_\ell)$ such that $w'_1 \leq \cdots \leq w'_\ell$.

Take $\boldsymbol{i} = (i_1, \ldots, i_{\ell-s}) \subset \{1, \ldots, \ell\}$, and let $d_{\boldsymbol{i}} = (d_{\boldsymbol{i},1}, \ldots, d_{\boldsymbol{i},\ell})$ be the sequence obtained by reordering $(d, \ldots, d, \delta_{i_1}, \ldots, \delta_{i_{\ell-s}})$ in non-decreasing order; we first compute the value of $\kappa_{\boldsymbol{i}}$ from (12). If $d_{\boldsymbol{i},1} = 0$ (which can happen only if $\ell = d$), then $\kappa_{\boldsymbol{i}} = 0$. Otherwise, the sequence $d_{\boldsymbol{i}}$ starts with $d_{\boldsymbol{i},1} \geq 1$ and increases until index $\ell - s$, after which it keeps the value d. On the other hand, the ordered sequence of weights never increases by more than 1, so that for all $k = 1, \ldots, \ell$, we have $w'_k \leq d_{\boldsymbol{i},k}$. In this case,

$$\kappa_{\boldsymbol{i}} = \max_{1 \leq k \leq \ell}(d_{\boldsymbol{i},1} \cdots d_{\boldsymbol{i},k} w_{k+1} \cdots w_m) = d_{\boldsymbol{i},1} \cdots d_{\boldsymbol{i},\ell} = d^s \delta_{i_1} \cdots \delta_{i_{\ell-s}};$$

note that this equality also holds if $d_{\boldsymbol{i},1} = 0$, since then both sides vanish. Since $\kappa = \sum_{\boldsymbol{i}=\{i_1,\ldots,i_{\ell-s}\}\subset\{1,\ldots,q\}} \kappa_{\boldsymbol{i}}$, we get

$$\kappa = \sum_{\boldsymbol{i}=\{i_1,\ldots,i_{\ell-s}\}\subset\{1,\ldots,\ell\}} d^s \delta_{i_1} \cdots \delta_{i_{\ell-s}} = d^s \eta_{\ell-s}(d-1, \ldots, d-\ell). \tag{13}$$

as claimed. □

The procedure $\mathsf{Isolated\_Points}\big(\bar{\boldsymbol{f}}^{[\lambda]}, \bar{\mathbf{H}}^{[\lambda]}\big)$ then uses the algorithm in Theorem 31 with input $\big(\bar{\boldsymbol{f}}^{[\lambda]}, \bar{\mathbf{H}}^{[\lambda]}\big)$. Writing as before $w_\lambda = \ell_1! \cdots \ell_r!$, the quantities used in the theorem become

$$\begin{aligned}
\mathfrak{c}_\lambda &= \frac{d^s \eta_{\ell-s}(d-1,\dots,d-\ell)}{w_\lambda},\\
\mathfrak{e}_\lambda &= \frac{n(d+1)^s \eta_{\ell-s}(d,\dots,d-\ell+1)}{w_\lambda},\\
\kappa_\lambda &= d^s \eta_{\ell-s}(d-1,\dots,d-\ell) = w_\lambda \mathfrak{c}_\lambda.
\end{aligned}$$

This implies that running $\mathsf{Isolated\_Points}\big(\bar{\boldsymbol{f}}^{[\lambda]}, \bar{\mathbf{H}}^{[\lambda]}\big)$ uses

$$O\tilde{~}\left(\left(\mathfrak{c}_\lambda(\mathfrak{e}_\lambda + \mathfrak{c}_\lambda^5) + d^2\mathfrak{c}_\lambda^2\right)n^4\Gamma\right)$$

operations which is again in

$$O\tilde{~}\left(d^2\mathfrak{c}_\lambda(\mathfrak{e}_\lambda + \mathfrak{c}_\lambda^5)n^4\Gamma\right).$$

As before, the number of solutions in the output is at most $\mathfrak{c}_\lambda$.

5.3. Finishing the proof of Theorem 5

We can now finish estimating the runtime of the $\mathsf{Critical\_Points\_Per\_Orbit}$ Algorithm. For partitions of length s we only need to compute $\bar{\boldsymbol{f}}^{[\lambda]}$ which takes $O\tilde{~}(n\binom{n+d}{d}^2)$ operations in $\mathbb{K}$ at Step 2a as per Lemma 29. At Step 2b, the procedure $\mathsf{Isolated\_Points}(\bar{\boldsymbol{f}}^{[\lambda]})$ takes at most

$$O\tilde{~}\left(d^2\mathfrak{c}_\lambda(\mathfrak{e}_\lambda + \mathfrak{c}_\lambda^5)n^4\Gamma\right)$$

operations in $\mathbb{K}$, as we saw in Subsection 5.2.1. The output of this procedure contains at most $\mathfrak{c}_\lambda$ points; then, by Lemma 17, the cost of the call to $\mathsf{Decompose}$ at Step 2c is $O\tilde{~}(\mathfrak{c}_\lambda^2\, n)$, which is negligible compared to the previous costs.

For partitions of length greater than s, computing $\bar{\boldsymbol{f}}^{[\lambda]}$ and $\bar{\mathbf{H}}^{[\lambda]}$ at Step 3a takes $O\tilde{~}(n^4\binom{n+d}{d}^2)$ operations in $\mathbb{K}$, by Lemma 29. The procedure $\mathsf{Isolated\_Points}\ \big(\bar{\boldsymbol{f}}^{[\lambda]}, \bar{\mathbf{H}}^{[\lambda]}\big)$ at Step 3b requires at most

$$O\tilde{~}\left(d^2\mathfrak{c}_\lambda(\mathfrak{e}_\lambda + \mathfrak{c}_\lambda^5)n^4\Gamma\right)$$

operations in $\mathbb{K}$, as we saw in Subsection 5.2.2. Again, since the number of solutions in the output is at most $\mathfrak{c}_\lambda$, the cost of $\mathsf{Decompose}$ at Step 3c is still $O\tilde{~}(\mathfrak{c}_\lambda^2\, n)$ which, as before, is negligible in comparison to the other costs. To complete our analysis, we need the following lemma.

Lemma 34. With all notation being as above, the following holds

$$\sum_{\lambda\vdash n, \ell_\lambda \geq s} \mathfrak{c}_\lambda \leq \mathfrak{c} \quad \text{and} \quad \sum_{\lambda\vdash n, \ell_\lambda \geq s} \mathfrak{e}_\lambda \leq \mathfrak{e},$$

where $\mathfrak{c} = d^s\binom{n+d-1}{n}$ and $\mathfrak{e} = n\,(d+1)^s\binom{n+d}{n}$.

Proof. The proof relies on the combinatorics of integer partitions and properties of elementary symmetric functions. Details are given in Appendix C. □

As a result, the total cost incurred by our calls to Isolated_Points and Decompose is

$$O\tilde{~}\left(\mathfrak{c}(\mathfrak{c}+\mathfrak{c}^5)n^9d^2\left(\binom{n+d}{d}+\binom{n}{s+1}\right)\right).$$

Since $\binom{n+d}{d} \leq (n+1)\binom{n+d-1}{d}$, we will simplify this further, by noticing that for $d \geq 2$ we have $\mathfrak{c} = n\,(d+1)^s\,\binom{n+d}{n} \leq n(n+1)d^{5s}\,\binom{n+d-1}{n}^5 = n(n+1)\mathfrak{c}^5$ so this is

$$O\tilde{~}\left(\mathfrak{c}^6n^{11}d^2\left(\binom{n+d}{d}+\binom{n}{s+1}\right)\right).$$

For the remaining operations, the total cost of Prepare_F and Prepare_F_H is

$$n^4\sum_{\lambda\vdash n,\ell_\lambda\geq s}\binom{n+d}{d}^2.$$

Since $\binom{n+d}{d} \leq (n+1)\binom{n+d-1}{d}$, the binomial term in the sum is in $O(n^2\mathfrak{c}^2)$, so the total is $O(n^5\mathfrak{c}^3)$, and can be neglected. Similarly, the cost of Remove_Duplicates is negligible. Therefore, the total complexity of Critical_Points_Per_Orbit is then in

$$O\tilde{~}\left(n^{11}d^{6s+2}\binom{n+d}{d}^6\left(\binom{n+d}{d}+\binom{n}{s+1}\right)\right)\subset\left(d^s\binom{n+d}{d}\binom{n}{s+1}\right)^{O(1)}.$$

Finally, the total number of solutions reported by our algorithm is at most $\sum_{\lambda\vdash n,\ell_\lambda\geq s}\mathfrak{c}_\lambda$, which itself is at most $\mathfrak{c}$.

6. Experimental results

In this section, we report on an implementation and set of experimental runs supporting the results in this paper. We compare our Critical_Points_Per_Orbit algorithm from Section 4.2 with a naive algorithm which computes a zero-dimensional parametrization of $V(I)$, where I is the ideal generated by $\boldsymbol{f}$ and the $(s+1)$-minors of $\mathrm{Jac}(\boldsymbol{f},\phi)$. Since no implementation of the weighted sparse determinantal homotopy algorithm is available at this time, we use Gröbner bases to solve polynomial systems for each orbit description. In practical terms the use of Gröbner bases solving is sufficient to see the advantage when the symmetric structure is exploited in our algorithm.

Our experiments are run using the Maple computer algebra system running on a computer with 16 GB RAM. The Gröbner basis algorithm in Maple uses the implementation of the F_4 and FGLM algorithms from the FGb package [21]. The symmetric polynomials $\boldsymbol{f}$ and ϕ are chosen uniformly at random in $\mathbb{K}[x_1,\ldots,x_n]$, with $\mathbb{K} = \mathrm{GF}(65521)$, and have the same degree n as the number of variables, that is, $\deg(f_1) = \cdots = \deg(f_s) = \deg(\phi) = n$. The number s of equations $\boldsymbol{f}$ ranges from 2 to $n-1$.

Our experimental results support the theoretical advantage gained by exploiting the symmetric structure of the input polynomials. In Table 1, we first report the number of points, denoted by D, that we compute using our algorithm. That is, D is the sum of the degrees $\deg(\mathscr{R}_\lambda)$ that we obtain for all partitions λ of length at least s. The next column is $\left\lceil\sum_{\ell_\lambda\geq s}\mathfrak{c}_\lambda\right\rceil$, which is an upper bound on D (here, $\mathfrak{c}_\lambda$ is as in Subsection 5.2). As we can see, this bound is quite sharp in general. We next give the upper bound $\mathfrak{c}$ from (3), which was proved in Lemma 34. While this bound is sufficient to prove asymptotic results

Table 1. Degrees and bounds

n	s	D	$\lceil \sum_{\ell_\lambda \geq s} \mathfrak{c}_\lambda \rceil$	$\mathfrak{c}$	$\deg(I)$	$\tilde{\mathfrak{c}}$
4	2	79	80	560	856	864
4	3	47	48	2240	744	768
5	2	425	432	3150	15575	16000
5	3	357	370	15750	18760	20000
5	4	143	157	78750	11160	12500
6	2	2222	2227	16632	-	337500
6	3	2439	2453	99792	-	540000
6	4	1482	1503	598752	-	486000
6	5	470	486	3592512	-	233280

(for fixed input degree, for instance, see the discussion in the introduction), we see that it is far from sharp.

Finally, we give the number of points $\deg(I)$ computed by the naive algorithm, together with the upper bound $\tilde{\mathfrak{c}}$ from (4). When the naive algorithm could not complete its computations within a 24 hour time period the $\deg(I)$ was unavailable. We see that in all cases, the output of our algorithm is much smaller than the one from the direct approach.

In Table 2 we report on our timings in a detailed fashion. Here, we give the time needed to compute the zero-dimensional representations $\deg(\mathscr{R}_\lambda)$ obtained by our algorithm, together with their degrees; Time(total) denotes the total time spent in our algorithm. On the other hand, Time(naive) is the time to compute a zero-dimensional parametrization for the algebraic set $V(I)$ using the naive algorithm. In the case of $n = 6$ our algorithm was efficient (with a maximum time of 1650 seconds) while the naive computations were all stopped since the computation had gone past 24 hours.

In our experiments, the output $\mathscr{R}_\lambda$ was always empty for partitions of length less than s. Indeed, for any partition λ of length at most $s-1$, $Z(\mathscr{R}_\lambda) = V(\bar{f}_1^{[\lambda]}, \ldots, \bar{f}_s^{[\lambda]})$, where the $\bar{f}_i^{[\lambda]}$ are s polynomials in less than s variables derived from the input $\boldsymbol{f}$. Since the polynomials $\boldsymbol{f}$ are chosen at random, the evaluated block symmetric polynomials $f_1^{[\lambda]}, \ldots, f_s^{[\lambda]}$ are generic. Using [45, Proposition 2.1.(ii)] or modifying slightly the proof of [45, Proposition 4.5], we indeed expect $Z(\mathscr{R}_\lambda)$ to be empty for such partitions λ of length less than s. However, we point out that this output can be non-trivial in the general, non-generic case.

Table 2. Algorithm Timings

n	s	Partition(λ)	Time($\mathscr{R}_\lambda$)	deg($\mathscr{R}_\lambda$)	$\lceil \mathfrak{c}_\lambda \rceil$	Time(total)	Time(naive)	deg(I)
4	2	$\lambda = (1^4)$	1.524s	7	8	3.136s	0.905s	856
		$\lambda = (1^2\,2^1)$	0.684s	48	48			
		$\lambda = (2^2)$	0.200s	8	8			
		$\lambda = (1^1 3^1)$	0.380s	16	16			
5	2	$\lambda = (1^5)$	9.236s	9	11	34.944s	2143.144s	15575
		$\lambda = (1^3\,2^1)$	6.832s	142	146			
		$\lambda = (1^2\,3)$	2.128s	112	113			
		$\lambda = (1^1\,2^2)$	2.816s	112	113			
		$\lambda = (1^1\,4^1)$	0.316s	25	25			
		$\lambda = (2^1\,3^1)$	0.392s	25	25			
5	3	$\lambda = (1^5)$	18.829s	31	37	48.019s	3423.660s	18760
		$\lambda = (1^3\,2^1)$	18.120s	202	209			
		$\lambda = (1^2\,3)$	4.607s	62	63			
		$\lambda = (1^1\,2^2)$	5.316s	62	63			
5	4	$\lambda = (1^5)$	17.080s	44	53	37.372s	969.396s	11160
		$\lambda = (1^3\,2^1)$	12.024s	99	105			

7. Conclusion and topics for future research

In this paper we have provided a new algorithm for efficiently describing the critical point set of a function ϕ a variety $V(\boldsymbol{f})$ with ϕ and the defining functions of the variety all symmetric. The algorithm takes advantage of the symmetries and lower bounds for describing the generators of the set of critical points and as a result is more efficient than previous approaches.

When $\boldsymbol{f} = (f_1, \ldots, f_s) \subset \mathbb{R}[x_1, \ldots, x_n]$, with $\mathbb{R}$ is a real field, then computing the critical points of polynomial maps restricted to $V(\boldsymbol{f})$ finds numerous applications in computational real algebraic geometry. As mentioned in the introduction, such computations provide an effective Morse-theoretic approach to many problems such as real root finding, quantifier elimination or answering connectivity queries (see [5]). We view the complexity estimates in this paper as a possible first step towards better algorithms for studying real algebraic sets defined by $\mathcal{S}_n$-invariant polynomials.

For instance, let d be the maximum degree of the entries in $\boldsymbol{f} = (f_1, \ldots, f_s)$ and assume that $\boldsymbol{f}$ generates an $(n-s)$-equidimensional ideal whose associated algebraic set is smooth. Then under these assumptions, we observe that the set $W(\phi_u, V(\boldsymbol{f}))$ with

$$\phi_u : (x_1, \ldots, x_n) \to (x_1 - u)^2 + \cdots + (x_n - u)^2$$

and $u \in \mathbb{R}$, has a non-empty intersection with all connected components of $V(\boldsymbol{f}) \cap \mathbb{R}^n$. Hence, when $W(\phi_u, \boldsymbol{f})$ is finite for a generic choice of u, then one can use our algorithm to decide whenever $V(\boldsymbol{f}) \cap \mathbb{R}^n$ is empty. This is done in time polynomial in d^s, $\binom{n+d}{d}$, $\binom{n}{s+1}$.

Obtaining an algorithm to decide whether $V(\boldsymbol{f}) \cap \mathbb{R}^n$ is empty in time polynomial in d^s, $\binom{n+d}{d}$, $\binom{n}{s+1}$, without assuming that $W(\phi_u, \boldsymbol{f})$ is finite for a generic $u \in \mathbb{R}$, is still an open problem.

Finally, note that many more involved algorithmic problems with symmetric semi-algebraic sets arise, as illustrated by [12, 7].

Acknowledgements. G. Labahn is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant number RGPIN-2020-04276. É. Schost is supported by an NSERC Discovery Grant. T. X. Vu is supported by a labex CalsimLab fellowship/scholarship. The labex CalsimLab, reference ANR-11-LABX-0037-01, is funded by the program "Investissements d'avenir" of the Agence Nationale de la Recherche, reference ANR-11-IDEX-0004-02. M. Safey El Din and T. X. Vu are supported by the ANR grants ANR-18-CE33-0011 Sesame, ANR-19-CE40-0018 De Rerum Natura and ANR-19-CE48-0015 ECARP and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement N. 813211 (POEMA).

References

[1] M.-E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. Zeros, multiplicities, and idempotents for zero-dimensional systems. In *Algorithms in Algebraic Geometry and Applications*, pages 1–15. Springer, 1996.

[2] P. Aubry, F. Rouillier, and M. Safey El Din. Real solving for positive dimensional systems. *Journal of Symbolic Computation*, 34(6):543–560, 2002.

[3] B. Bank, M. Giusti, J. Heintz, and M. Safey El Din. Intrinsic complexity estimates in polynomial optimization. *Journal of Complexity*, 30(4):430–443, 2014.

[4] B. Bank, M. Giusti, J. Heintz, M. Safey El Din, and É. Schost. On the geometry of polar varieties. *Applicable Algebra in Engineering, Communication and Computing*, pages 33–83, 2010.

[5] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[6] S. Basu, M.-F. Roy, M. Safey El Din, and É. Schost. A baby-step giant-step roadmap algorithm for general real algebraic sets. *Foundations of Computational Mathematics*, 14(6):1117–1172, 2014.

[7] Saugata Basu and Cordian Riener. Bounding the equivariant betti numbers of symmetric semi-algebraic sets. *Advances in Mathematics*, 305:803–855, 2017.

[8] G. M. Besana, S. Di Rocco, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Cell decomposition of almost smooth real algebraic surfaces. *Numer. Algorithms*, 63(4):645–678, 2013.

[9] M. Bläser and G. Jindal. On the Complexity of Symmetric Polynomials. In A. Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 47:1–47:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[10] A. Bompadre, G. Matera, R. Wachenchauzer, and A. Waissbein. Polynomial equation solving by lifting procedures for ramified fibers. *Theoretical Computer Science*, 315(2-3):335–369, May 2004.

[11] D. A. Brake, D. J. Bates, W. Hao, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Algorithm 976: {B}ertini_real: numerical decomposition of real algebraic curves and surfaces. *ACM Trans. Math. Software*, 44(1):Art. 10, 30, 2017.

[12] L. Bröcker. On symmetric semialgebraic sets and orbit spaces. *Banach Center Publications*, 44(1):37–50, 1998.

[13] L. Busé and A. Karasoulou. Resultant of an equivariant polynomial system with respect to the symmetric group. *Journal of Symbolic Computation*, 76:142–157, 2016.

[14] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *Proceedings of the 1989 International Symposium on Symbolic and Algebraic Computation*, ISSAC '89, pages 121–128. ACM, 1989.

[15] A. Colin. Solving a system of algebraic equations with symmetries. *Journal of Pure and Applied Algebra*, 117-118:195 – 215, 1997.

[16] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3rd ed.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[17] H. Derksen and G. Kemper. *Computational Invariant Theory.* Invariant Theory and Algebraic Transformation Groups, I. Springer-Verlag, Berlin, 2002. Encyclopedia of Mathematical Sciences, 130.

[18] J. A. Eagon and D. G. Northcott. Ideals defined by matrices and a certain complex associated with them. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 269(1337):188–204, 1962.

[19] D. Eisenbud. *Commutative Algebra: with a View Toward Algebraic Geometry.* Graduate Texts in Mathematics. Springer, New York, Berlin, Heildelberg, 1995.

[20] N.-E. Fahssi. Polynomial triangles revisited. https://arxiv.org/abs/1202.0228, 2012.

[21] J.-C. Faugère. FGb: A Library for Computing Gröbner Bases. In Komei Fukuda, Joris Hoeven, Michael Joswig, and Nobuki Takayama, editors, *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 84–87, Berlin, Heidelberg, September 2010. Springer Berlin / Heidelberg.

[22] J.-C. Faugère, M. Hering, and J. Phan. The membrane inclusions curvature equations. *Advances in Applied Mathematics*, 31(4):643 – 658, 2003.

[23] J.-C. Faugère and S. Rahmany. Solving systems of polynomial equations with symmetries using SAGBI-Gröbner bases. In *Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, ISSAC '09, pages 151–158, New York, NY, USA, 2009. ACM.

[24] J.-C. Faugère, M. Safey El Din, and P.-J. Spaenlehauer. Critical points and Gröbner bases: The unmixed case. In *Proceedings of the 2012 International Symposium on Symbolic and Algebraic Computation*, ISSAC '12, pages 162–169, New York, NY, USA, 2012. ACM.

[25] J.-C. Faugère and J. Svartz. Solving polynomial systems globally invariant under an action of the symmetric group and application to the equilibria of N vortices in the plane. In *Proceedings of the 2012 International Symposium on Symbolic and Algebraic Computation*, ISSAC '12, pages 170–178, New York, NY, USA, 2012. ACM.

[26] J. V. Z. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[27] P. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using Gröbner bases. In *AAECC*, volume 356 of *LNCS*, pages 247–257. Springer, 1989.

[28] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124:101–146, 1998.

[29] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. When polynomial equation systems can be solved fast? In *AAECC-11*, volume 948 of *LNCS*, pages 205–231. Springer, 1995.

[30] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner-free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001.

[31] A. Greuet and M. Safey El Din. Probabilistic algorithm for polynomial optimization over a real algebraic set. *SIAM Journal on Optimization*, 24(3):1313–1343, 2014.

[32] F. Guo, M. Safey El Din, and L. Zhi. Global optimization of polynomials using generalized critical values and sums of squares. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, ISSAC '10, pages 107–114, New York, NY, USA, 2010. ACM.

[33] J. D. Hauenstein. Numerically computing real points on algebraic sets. *Acta Appl. Math.*, 125:105–119, 2013.

[34] J. D. Hauenstein, M. Safey El Din, É. Schost, and T.X. Vu. Solving determinantal systems using homotopy techniques. *Journal of Symbolic Computation*, 104:754–804, 2021.

[35] J. Heintz, G. Jeronimo, J. Sabia, and P. Solerno. Intersection theory and deformation algorithms: the multi-homogeneous case, 2002.

[36] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *Journal of Complexity*, 16(1):70 – 109, 2000.

[37] M. I. Herrero, G. Jeronimo, and J. Sabia. Computing isolated roots of sparse polynomial systems in affine space. *Theoretical Computer Science*, 411(44):3894 – 3904, 2010.

[38] M. I. Herrero, G. Jeronimo, and J. Sabia. Affine solution sets of sparse polynomial systems. *Journal of Symbolic Computation*, 51:34 – 54, 2013. Effective Methods in Algebraic Geometry.

[39] M. I. Herrero, G. Jeronimo, and J. Sabia. Elimination for generic sparse polynomial systems. *Discrete & Computational Geometry*, 51(3):578–599, 2014.

[40] E. Hubert. Invariant algebraic sets and symmetrization of polynomial systems. *Journal of Symbolic Computation*, 95:53–67, 2019.

[41] E. Hubert and G. Labahn. Computation of invariants of finite abelia groups. *Mathematics of Computation*, 85(302):3029–3050, 2016.

[42] G. Jeronimo, G. Matera, P. Solernó, and A. Waissbein. Deformation techniques for sparse systems. *Foundations of Computational Mathematics*, 9(1):1–50, February 2009.

[43] G. Jeronimo and D. Perrucci. A probabilistic symbolic algorithm to find the minimum of a polynomial function on a basic closed semialgebraic set. *Discrete & Computational Geometry*, 52(2):260–277, 2014.

[44] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *Journal für die Reine und Angewandte Mathematik*, 92:1–122, 1882.

[45] G. Labahn, M. Safey El Din, É. Schost, and T.X. Vu. Homotopy techniques for solving sparse column support determinantal polynomial systems. *Journal of Complexity*, 66:101557, 2021.

[46] G. Lecerf and É. Schost. Fast multivariate power series multiplication in characteristic zero. *SADIO Electronic Journal on Informatics and Operations Research*, 5(1):1–10, September 2003.

[47] I. G. Macdonald. *Symmetric functions and Hall polynomials.* Oxford university press, 1998.

[48] P. Moustrou, C. Riener, and H. Verdure. Symmetric ideals, specht polynomials and solutions to symmetric systems of equations. *Journal of Symbolic Computation*, 107:106–121, 2021.

[49] J. Nie, J. Demmel, and B. Sturmfels. Minimizing polynomials via sum of squares over the gradient ideal. *Mathematical programming*, 106(3):587–606, 2006.

[50] J. Nie and K. Ranestad. Algebraic degree of polynomial optimization. *SIAM Journal on Optimization*, 20(1):485–502, April 2009.

[51] N. Perminov and Sh. Shakirov. Discriminants of symmetric polynomials. *arXiv preprint arXiv:0910.5757*, 2009.

[52] A. Poteaux and É. Schost. On the complexity of computing with zero-dimensional triangular sets. *Journal of Symbolic Computation*, 50:110–138, 2013.

[53] C. Riener. On the degree and half-degree principle for symmetric polynomials. *Journal of Pure and Applied Algebra*, 216(4):850 – 856, 2012.

[54] C. Riener. Symmetric semi-algebraic sets and non-negativity of symmetric polynomials. *Journal of Pure and Applied Algebra*, 220(8):2809 – 2815, 2016.

[55] C. Riener and M. Safey El Din. Real root finding for equivariant semi-algebraic systems. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, pages 335–342, New York, NY, USA, 2018. ACM.

[56] C. Riener, T. Theobald, L. J. Andrén, and J. B. Lasserre. Exploiting symmetries in sdp-relaxations for polynomial optimization. *Mathematics of Operations Research*, 38(1):122–141, 2013.

[57] F. Rouillier. Solving zero-dimensional systems through the Rational Univariate Representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[58] M. Safey El Din and É. Schost. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *J. ACM*, 63(6):48:1–48:37, 2017.

[59] M. Safey El Din and É. Schost. Bit complexity for multi-homogeneous polynomial system solving - application to polynomial minimization. *Journal of Symbolic Computation*, 87:176–206, 2018.

[60] P.-J. Spaenlehauer. On the complexity of computing critical points with Gröbner bases. *SIAM Journal on Optimization*, 24(3):1382–1401, 2014.

[61] S. Stefan. Gröbner bases of symmetric ideals. *Journal of Symbolic Computation*, 54:72–86, 2013.

[62] B. Sturmfels. *Algorithms in Invariant Theory.* Springer-Verlag, Berlin, Heidelberg, 1993.

[63] V. Timofte. On the positivity of symmetric polynomial functions.: Part I: General results. *Journal of Mathematical Analysis and Applications*, 284(1):174 – 190, 2003.

[64] T. X. Vu. Computing critical points for algebraic systems defined by hyperoctahedral invariant polynomials. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, ISSAC '22, New York, NY, USA, 2022. ACM.

A. Proof of Proposition 18

The proof of Proposition 18 will be done in stages. We start with some rather straightforward lemmas.

Lemma 35. Consider an $\mathcal{S}_\lambda$-equivariant sequence $\boldsymbol{q} = (q_1, \dots, q_\ell)$ in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$. Then, for any $I \subset \{1, \dots, \ell\}$ and any σ in $\mathcal{S}_\lambda$, we have $\sigma(q_I) = q_{\sigma(I)}$.

Proof. By induction on the size of I. □

Lemma 36. Consider a sequence $\boldsymbol{q} = (q_1, \dots, q_\ell)$ in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]$, and suppose that

(i) $z_i - z_j$ divides $q_i - q_j$ for $1 \le i < j \le \ell$,

(ii) $\boldsymbol{q}$ is $\mathcal{S}_\lambda$-equivariant.

Then, for k in $\{1, \dots, r\}$ and s in $\{1, \dots, \ell_k\}$, the polynomial $\sum_{i=\tau_k+1}^{\tau_k+s} q_{\{i, \tau_k+s+1, \dots, \ell\}}$ is invariant under any permutation of $\{z_{\tau_k+1}, \dots, z_{\tau_k+s}\}$.

Proof. For any $\sigma \in \mathcal{S}_\lambda$ permuting only $\{z_{\tau_k+1}, \dots, z_{\tau_k+s}\}$, we have, using the previous lemma,

$$\sigma\Big(\sum_{i=1}^{\tau_k+s} q_{\{i, \tau_k+s+1, \dots, \ell\}}\Big) = \sum_{i=\tau_k+1}^{\tau_k+s} \sigma\big(q_{\{i, \tau_k+s+1, \dots, \ell\}}\big) = \sum_{i=\tau_k+1}^{\tau_k+s} q_{\{\sigma(i), \tau_k+s+1, \dots, \ell\}}.$$

Since σ permutes $\{z_{\tau_k+1}, \dots, z_{\tau_k+s}\}$ and the last sum runs over all $i = \tau_k + 1, \dots, \tau_k + s$, it equals $\sum_{i=\tau_k+1}^{\tau_k+s} q_{\{i, \tau_k+s+1, \dots, \ell\}}$. □

We can now prove the proposition. The fact that all entries of $\boldsymbol{p}$ are polynomials follows from our first assumption. Proving that they are $\mathcal{S}_\lambda$-invariant requires more work, as we have to deal with numerous cases. While most are straightforward, the last case does involve nontrivial calculations.

Fix $k \in \{0, \dots, r-1\}$. We first prove that for s in $\{1, \dots, \ell_{k+1}\}$, i in $\{\tau_k+1, \dots, \tau_k+s\}$, and m in $\{0, \dots, r-1\}$, with $m \neq k$, then the term $q_{\{i, \tau_k+s+1, \dots, \tau_r\}}$ is symmetric in $\{z_{\tau_m+1}, \dots, z_{\tau_{m+1}}\}$. Indeed, consider a permutation $\sigma \in \mathcal{S}_\lambda$ that acts only on variables $\{z_{\tau_m+1}, \dots, z_{\tau_{m+1}}\}$. By Lemma 35, $\sigma(q_{\{i, \tau_k+s+1, \dots, \tau_r\}})$ is equal to $q_{\{\sigma(i), \sigma(\tau_k+s+1), \dots, \sigma(\tau_r)\}}$. If $m < k$, then all indices $i, \tau_k + s + 1, \dots, \tau_r$ are left invariant by σ while for $m > k$, $[\sigma(i), \sigma(\tau_k + s + 1), \dots, \sigma(\tau_r)]$ is a permutation of $[i, \tau_k + s + 1, \dots, \tau_r]$. In both cases,

$$q_{\{\sigma(i), \sigma(\tau_k+s+1), \dots, \sigma(\tau_r)\}} = q_{\{i, \tau_k+s+1, \dots, \tau_r\}},$$

as claimed.

Consider first the invariance of $p_{\tau_{k+1}}$. By Lemma 36, the sum $\sum_{i=\tau_k+1}^{\tau_{k+1}} q_{\{i, \tau_{k+1}+1, \dots, \tau_r\}}$ is symmetric in $\{z_{\tau_k+1}, \dots, z_{\tau_{k+1}}\}$. Next, for i in $\{\tau_k+1, \dots, \tau_{k+1}\}$ and m in $\{0, \dots, r-1\}$, with $m \neq k$, each term $q_{\{i, \tau_{k+1}+1, \dots, \tau_r\}}$ is symmetric in $\{z_{\tau_m+1}, \dots, z_{\tau_{m+1}}\}$, making use of the previous paragraph with $s = \ell_{k+1}$. As a result, $p_{\tau_{k+1}}$ is $\mathcal{S}_\lambda$-invariant.

Next, for j in $\{1, \ldots, \ell_{k+1} - 1\}$ and σ in $\mathcal{S}_\lambda$, we prove that $\sigma(p_{\tau_k+j}) = p_{\tau_k+j}$. Assume first that σ acts only on $\{z_{\tau_m+1}, \ldots, z_{\tau_{m+1}}\}$, for some m in $\{0, \ldots, r-1\}$ with $m \neq k$. For s in $\{1, \ldots, j\}$, the polynomial $\eta_{j-s}(z_{\tau_k+s+2}, \ldots, z_{\tau_{k+1}})$ depends only on $\{z_{\tau_k+1}, \ldots, z_{\tau_{k+1}}\}$ and so is σ-invariant. Using our earlier argument we see that for i in $\{\tau_k+1, \ldots, \tau_k+s\}$ the divided difference $q_{\{i,\tau_k+s+1,\ldots,\tau_r\}}$ is σ-invariant. As a result, p_{τ_k+j} itself is σ-invariant.

It remains to prove that p_{τ_k+j} is σ-invariant for a permutation σ of $\{\tau_k + 1, \ldots, \tau_{k+1}\}$. We do this first for $\sigma = (\tau_k + 1, \tau_k + 2)$, by proving that all summands in the definition of p_{τ_k+j} are σ-invariant. For any s in $\{2, \ldots, j\}$, $\eta_{j-s}(z_{\tau_k+s+2}, \ldots, z_{\tau_{k+1}})$ does not depend on $(z_{\tau_k+1}, z_{\tau_k+2})$, so it is σ-invariant. For s in $\{2, \ldots, j\}$, the sum $\sum_{i=\tau_k+1}^{\tau_k+s} q_{\{i,\tau_k+s+1,\ldots,\tau_r\}}$ is symmetric in $(\tau_k + 1, \tau_k + 2)$, since σ just permutes two terms in the sum while for $s = 1$, $q_{\{\tau_k+1,\tau_k+2,\ldots,\tau_r\}}$ is symmetric in $(z_{\tau_k+1}, z_{\tau_k+2})$ by Lemma 35. Thus, our claim is proved for $\sigma = (\tau_k + 1, \tau_k + 2)$.

It remains to prove that p_{τ_k+j} is invariant in $(z_{\tau_k+2}, \ldots, z_{\tau_{k+1}})$. For any $t = 1, \ldots, j$, set

$$p_{\tau_k+j,t} = \sum_{s=t}^{j} \eta_{j-s}(z_{\tau_k+t+2}, \ldots, z_{\tau_{k+1}}) \Big(\sum_{i=\tau_k+1}^{\tau_k+s} q_{\{i,\tau_k+s+1,\ldots,\tau_r\}} \Big). \tag{A.1}$$

Then $p_{\tau_k+j} = p_{\tau_k+j,1}$ and we have the recursive identity

$$p_{\tau_k+j,t-1} = p_{\tau_k+j,t} + \eta_{j-t+1}(z_{\tau_k+t+1}, \ldots, z_{\tau_{k+1}}) \Big(\sum_{i=\tau_k+1}^{\tau_k+t-1} q_{\{i,\tau_k+t,\ldots,\tau_r\}} \Big). \tag{A.2}$$

For any t, set $\boldsymbol{z}_{:t} = (z_{\tau_k+1}, \ldots, z_{\tau_k+t})$ and $\boldsymbol{z}_{t:} = (z_{\tau_k+t}, \ldots, z_{\tau_{k+1}})$. We will show that for $t = 1, \ldots, j$, the polynomial $p_{\tau_k+j,t}$ satisfies:

$$p_{\tau_k+j,t} \text{ is block symmetric in } \boldsymbol{z}_{:t} \text{ and } \boldsymbol{z}_{t+1:} \tag{A.3}$$

Taking $t = 1$ implies that $p_{\tau_k+j} = p_{\tau_k+j,1}$ is symmetric in $\boldsymbol{z}_{2:} = (z_{\tau_k+2}, \ldots, z_{\tau_{k+1}})$, as claimed.

To prove statement (A.3) we use decreasing induction on $t = j, \ldots, 1$. The statement is true when $t = j$ since in this case $p_{\tau_k+j,j} = \sum_{i=\tau_k+1}^{\tau_k+j} q_{\{i,\tau_k+j+1,\ldots,\tau_r\}}$, which is symmetric in $\boldsymbol{z}_{:j}$ by Lemma 36, while each summand $q_{\{i,\tau_k+j+1,\ldots,\tau_r\}}$ is symmetric in $\boldsymbol{z}_{j+1:}$ by Lemma 35. Assume now that (A.3) is true for some index t in $\{2, \ldots, j\}$; we show that it also holds for $t - 1$. That is, we have $p_{\tau_k+j,t}$ is block symmetric in $\boldsymbol{z}_{:t}$ and $\boldsymbol{z}_{t+1:}$ and need to show that $p_{\tau_k+j,t-1}$ is block symmetric in $\boldsymbol{z}_{:t-1}$ and $\boldsymbol{z}_{t:}$.

From Lemma 36, we have that $\sum_{i=\tau_k+1}^{\tau_k+t-1} q_{\{i,\tau_k+t,\ldots,\tau_r\}}$ is symmetric in $\boldsymbol{z}_{:t-1}$. Furthermore, from our induction hypothesis, the polynomial $p_{\tau_k+j,t}$ is symmetric in $\boldsymbol{z}_{:t-1}$, while $\eta_{j-t+1}(z_{\tau_k+t+1}, \ldots, \tau_{k+1})$ depends only on $\boldsymbol{z}_{t:}$. Thus, in view of (A.2), we see that $p_{\tau_k+j,t-1}$ is symmetric in $\boldsymbol{z}_{:t-1}$. It remains to prove that it is also symmetric in $\boldsymbol{z}_{t:}$.

We will prove this by showing $\sigma(p_{\tau_k+j,t-1}) = p_{\tau_k+j,t-1}$ for any $\sigma = (\tau_k + t + 1, \tau_k + \epsilon)$ with $\epsilon \in \{t, t + 2, \ldots, \ell_{k+1}\}$. For any such σ with $t + 2 \leq \epsilon \leq \ell_{k+1}$, our induction hypothesis implies that $\sigma(p_{\tau_k+j,t}) = p_{\tau_k+j,t}$, while $\sigma(\eta_{j-t+1}(z_{\tau_k+t+1}, \ldots, \tau_{k+1})) = \eta_{j-t+1}(z_{\tau_k+t+1}, \ldots, \tau_{k+1})$ and $\sigma\big(q_{\{i,\tau_k+t,\ldots,\tau_r\}}\big) = q_{\{i,\tau_k+t,\ldots,\tau_r\}}$ hold for all i. Together with (A.2), we get $\sigma(p_{\tau_k+j,t-1}) = p_{\tau_k+j,t-1}$. Finally, if $\sigma = (\tau_k + t + 1, \tau_k + t)$, then we have

$$\sigma(\eta_{j-t+1}(z_{\tau_k+t+1}, \ldots, \tau_{k+1})) = \eta_{j-t+1}(z_{\tau_k+t}, z_{\tau_k+t+2}, \ldots, \tau_{k+1})$$

and $\sigma\big(q_{\{i,\tau_k+t,\dots,\tau_r\}}\big) = q_{\{i,\tau_k+t,\dots,\tau_r\}}$ for all $i = \tau_k+1,\dots,\tau_k+t-1$. Notice that

$$\eta_{j-t+1}(z_{\tau_k+t}, z_{\tau_k+t+2},\dots,\tau_{k+1}) - \eta_{j-t+1}(z_{\tau_k+t+1},\dots,\tau_{k+1}) = (z_{\tau_k+t} - z_{\tau_k+t+1})\,\eta_{j-t}(z_{\tau_k+t+2},\dots,z_{\tau_{k+1}}).$$

Therefore,

$$\begin{aligned}\sigma(p_{\tau_k+j,t-1}) - p_{\tau_k+\hat{\imath},t-1} &= \sigma(p_{\tau_k+j,t}) - p_{\tau_k+j,t} \\ &\quad + (z_{\tau_k+t} - z_{\tau_k+t+1})\,\eta_{j-t}(z_{\tau_k+t+2},\dots,z_{\tau_{k+1}})\Big(\sum_{i=\tau_k+1}^{\tau_k+t-1} q_{\{i,\tau_k+t,\dots,\tau_r\}}\Big) \\ &= \sigma(p_{\tau_k+j,t}) - p_{\tau_k+j,t} + \eta_{j-t}(z_{\tau_k+t+2},\dots,z_{\tau_{k+1}}) \\ &\quad \Big(\sum_{i=\tau_k+1}^{\tau_k+t-1}(q_{\{i,\tau_k+t+1,\tau_k+t+2,\dots,\tau_r\}} - q_{\{i,\tau_k+t,\tau_k+t+2,\dots,\tau_r\}})\Big), \qquad \text{(A.4)}\end{aligned}$$

where the last equality follows from the definition of divided differences. In particular,

$$\sigma(p_{\tau_k+j,j-1}) - p_{\tau_k+j,j-1} = \sigma(p_{\tau_k+j,j}) - p_{\tau_k+j,j} + \sum_{i=\tau_k+1}^{\tau_k+j-1}(q_{\{i,\tau_k+j+1,\dots,\tau_r\}} - q_{\{i,\tau_k+j,\tau_k+j+2,\dots,\tau_r\}}).$$

In addition, since $p_{\tau_k+j,j} = \sum_{i=\tau_k+1}^{\tau_k+j} q_{\{i,\tau_k+j+1,\dots,\tau_r\}}$, then when $\sigma = (\tau_k+j+1, \tau_k+j)$, we have $\sigma(p_{\tau_k+j,j}) - p_{\tau_k+j,j} = \sum_{i=\tau_k+1}^{\tau_k+j-1}(q_{\{i,\tau_k+j,\tau_k+j+2,\dots,\tau_r\}} - q_{\{i,\tau_k+j+1,\dots,\tau_r\}})$. This implies that $\sigma(p_{\tau_k+j,j-1}) - p_{\tau_k+j,j-1} = 0$.

When $t \le j-1$, from (A.2), taken at index $t+1$, if $\sigma = (\tau_k+t+1, \tau_k+t)$, we also have

$$\sigma(p_{\tau_k+j,t}) = \sigma(p_{\tau_k+j,t+1}) + \eta_{j-t}(z_{\tau_k+t+2},\dots,z_{\tau_{k+1}})\Big(\sum_{i=\tau_k+1}^{\tau_k+t-1} q_{\{i,\tau_k+t,\tau_k+t+2,\dots,\tau_r\}} + q_{\{\tau_k+t,\tau_k+t+1,\dots,\tau_{k+1}\}}\Big).$$

Then, by subtraction:

$$\begin{aligned}\sigma(p_{\tau_k+j,t}) - p_{\tau_k+j,t} &= \sigma(p_{\tau_k+j,t+1}) - p_{\tau_k+j,t+1} + \eta_{j-t}(z_{\tau_k+t+2},\dots,z_{\tau_{k+1}}) \\ &\quad \Big(\sum_{i=\tau_k+1}^{\tau_k+t-1}(q_{\{i,\tau_k+t,\tau_k+t+2,\dots,\tau_r\}} - q_{\{i,\tau_k+t+1,\dots,\tau_r\}})\Big)\end{aligned}$$

and so

$$\begin{aligned}\sigma(p_{\tau_k+j,t+1}) - p_{\tau_k+j,t+1} =& \sigma(p_{\tau_k+j,t}) - p_{\tau_k+j,t} + \eta_{j-t}(z_{\tau_k+t+2},\dots,z_{\tau_{k+1}}) \\ & \Big(\sum_{i=\tau_k+1}^{\tau_k+t-1}(q_{\{i,\tau_k+t+1,\dots,\tau_r\}} - q_{\{i,\tau_k+t,\tau_k+t+2,\dots,\tau_r\}})\Big). \qquad \text{(A.5)}\end{aligned}$$

Combining (A.4) and (A.5) gives $\sigma(p_{\tau_k+j,t-1}) - p_{\tau_k+j,t-1} = \sigma(p_{\tau_k+j,t+1}) - p_{\tau_k+j,t+1}$. By induction, we have that $p_{\tau_k+j,t+1}$ is symmetric in $\boldsymbol{z}_{:t+1}$ and so $\sigma(p_{\tau_k+j,t+1}) = p_{\tau_k+j,t+1}$ for $\sigma = (\tau_k+t+1, \tau_k+t)$ which in turn implies that $\sigma(p_{\tau_k+j,t-1}) = p_{\tau_k+j,t-1}$. This gives our result.

B. Proof of Proposition 20

Define the row vector

$$\boldsymbol{h} = \big(h_{\tau_0+1}, \ldots, h_{\tau_1}, \ldots, h_{\tau_{r-1}+1}, \ldots, h_{\tau_r}\big)$$

where, for $k = 0, \ldots, r-1$ and $j = 1, \ldots, \ell_{k+1}$,

$$h_{\tau_k+j} = \sum_{i=\tau_k+1}^{\tau_k+j} q_{\{i,\tau_k+j+1,\ldots,\tau_r\}}. \tag{B.1}$$

Then for all $i = 1, \ldots, m$, $k = 0, \ldots, r-1$, $p_{\tau_k+\ell_{k+1}} = h_{\tau_k+\ell_{k+1}}$, and for $j = 1, \ldots, \ell_{k+1} - 1$,

$$p_{\tau_k+j} = \sum_{s=1}^{j} \eta_{j-s}(z_{\tau_k+s+2}, \ldots, z_{\tau_{k+1}})\, h_{\tau_k+s}.$$

Then $\boldsymbol{h} = \boldsymbol{p}\mathbf{M}$, where we recall that $\mathbf{M}$ is the block-diagonal matrix with blocks $\mathbf{M}_1, \ldots, \mathbf{M}_r$ where

$$\mathbf{M}_{k+1} = \begin{pmatrix} 1 & \eta_1(z_{\tau_k+3}, \ldots, z_{\tau_{k+1}}) & \eta_2(z_{\tau_k+3}, \ldots, z_{\tau_{k+1}}) & \cdots & \eta_{\ell_{k+1}-2}(z_{\tau_k+3}, \ldots, z_{\tau_{k+1}}) & 0 \\ 0 & 1 & \eta_1(z_{\tau_k+4}, \ldots, z_{\tau_{k+1}}) & \cdots & \eta_{\ell_{k+1}-3}(z_{\tau_k+4}, \ldots, z_{\tau_{k+1}}) & 0 \\ 0 & 0 & 1 & \cdots & \eta_{\ell_{k+1}-4}(z_{\tau_k+5}, \ldots, z_{\tau_{k+1}}) & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Then $\det(\mathbf{M}) = 1$ and $\mathbf{N} = \mathbf{M}^{-1}$ is also a polynomial matrix in $\mathbb{K}[\boldsymbol{Z}]$ with $\det(\mathbf{N}) = 1$.

We construct a matrix $\mathbf{J}$ which defines the column operations converting $\boldsymbol{h}$ into $\boldsymbol{q}$ as follows. Recall that for $k = 0, \ldots, r-1$ and $j = 1, \ldots, \ell_{k+1}$, we have defined the following $\tau_r \times \tau_r$ polynomial matrices. Set $\mathbf{B}_{\tau_0+1} = \mathbf{I}_{\tau_r}$, $\mathbf{C}_{\tau_0+1} = \mathbf{I}_{\tau_r}$, $\mathbf{D}_{\tau_0+j} = \mathbf{I}_{\tau_r}$, and

$$\mathbf{B}_{\tau_k+j} = \left(\begin{array}{c|c|c} \mathbf{I}_{\tau_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{E}_{k,j} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{\tau_r-\tau_{k+1}} \end{array}\right), \text{ with } \mathbf{E}_{k,j} = \left(\begin{array}{c|c|c} & z_{\tau_k+j} - z_{\tau_k+1} & \\ \mathbf{I}_{j-1} & \vdots & \mathbf{0} \\ & z_{\tau_k+j} - z_{\tau_k+j-1} & \\ \hline 0 \;\ldots\; 0 & -1 & \mathbf{0} \\ \hline \mathbf{0} & 0 & \mathbf{I}_{\ell_{k+1}-j} \end{array}\right);$$

$$\mathbf{C}_{\tau_k+j} = \left(\begin{array}{c|c|c} \mathbf{I}_{\tau_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{F}_{k,j} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{\tau_r-\tau_{k+1}} \end{array}\right), \text{ with } \mathbf{F}_{k,j} = \left(\begin{array}{c|c|c} \mathbf{diag}(z_{\tau_k+j} - z_{\tau_k+t})_{t=1}^{j-1} & \mathbf{0} & \mathbf{0} \\ \hline \frac{-1}{j} \;\ldots\; \frac{-1}{j} & \frac{-1}{j} & \mathbf{0} \\ \hline \mathbf{0} & 0 & \mathbf{I}_{\ell_{k+1}-j} \end{array}\right);$$

$$\mathbf{D}_{\tau_k+j} = \left(\begin{array}{c|c|c} \mathbf{diag}(z_{\tau_k+j}-z_t)_{t=1}^{\tau_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{G}_{k,j} & \mathbf{I}_{\ell_{k+1}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{\tau_r-\tau_{k+1}} \end{array}\right), \begin{array}{c}\text{with } j^{th} \text{ row of } \mathbf{G}_{k,j} \text{ is } (1,\dots,1) \text{ and} \\ \text{others are zeros}\end{array}$$

Let

$$\mathbf{J} = \prod_{k=0}^{r-1} \prod_{j=1}^{\ell_{k+1}} \mathbf{B}_{\tau_k+j}\, \mathbf{C}_{\tau_k+j}\, \mathbf{D}_{\tau_k+j} \quad \in \quad \mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r]^{\tau_r \times \tau_r}.$$

We will prove that this matrix satisfies $\boldsymbol{q} = \boldsymbol{h}\,\mathbf{J}$. Note first that, for $k = 0, \dots, r-1$ and $j = 1, \dots, \ell_{k+1}$ we have $\det(\mathbf{B}_{\tau_k+j}) = \det(\mathbf{E}_{k,j}) = -1$, $\det(\mathbf{C}_{\tau_k+j}) = \det(\mathbf{F}_{k,j}) = \frac{-1}{j} \prod_{t=1}^{j-1} (z_{\tau_k+j} - z_t)$, and $\det(\mathbf{D}_{\tau_k+j}) = \prod_{t=1}^{\tau_k} (z_{\tau_k+j} - z_t)$. This implies that

$$\det(\mathbf{J}) = \alpha \prod_{k=0}^{r-1} \prod_{j=1}^{\ell_{k+1}} \prod_{t=1}^{j-1} (z_{\tau_k+j} - z_t) \prod_{t=1}^{\tau_k} (z_{\tau_k+j} - z_t) = \alpha\Delta \text{ for some } \alpha \in \mathbb{K}_{\neq 0}.$$

Define $\mathbf{U} = \mathbf{N}\,\mathbf{J}$. Then $\boldsymbol{p} = \boldsymbol{q}\,\mathbf{U}$, and $\det(\mathbf{U})$ is a unit in $\mathbb{K}[\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_r, 1/\Delta]$, as claimed.

It remains to prove $\boldsymbol{q} = \boldsymbol{h}\,\mathbf{J}$. For $s = 0, \dots, \tau_r$, define

$$\boldsymbol{q}_s = \begin{pmatrix} q_{\{1,s+1,\dots,\tau_r\}} & \cdots & q_{\{s,s+1,\dots,\tau_r\}} & h_{s+1} & \cdots & h_{\tau_r} \end{pmatrix},$$

so that for $s = 0$ we have $\boldsymbol{q}_0 = \boldsymbol{h}$, whereas for $s = \tau_r$ we have $\boldsymbol{q}_{\tau_r} = \boldsymbol{q}$. We prove the following: for k in $\{0, \dots, r-1\}$ and j in $\{1, \dots, \ell_k\}$,

$$\boldsymbol{q}_{\tau_k+j} = \boldsymbol{q}_{\tau_k+j-1} \mathbf{B}_{\tau_k+j}\, \mathbf{C}_{\tau_k+j}\, \mathbf{D}_{\tau_k+j}. \tag{B.2}$$

Our claim $\boldsymbol{q} = \boldsymbol{h}\,\mathbf{J}$ then follows from a direct induction, taking into account the values of $\boldsymbol{q}_0$ and $\boldsymbol{q}_{\tau_r}$ given above.

Take k in $\{0, \dots, r-1\}$ and j in $\{1, \dots, \ell_k\}$. Right-multiplying $\boldsymbol{q}_{\tau_k+j-1}$ by $\mathbf{B}_{\tau_k+j}$ only affects the entry at index $\tau_k + j$. It replaces h_{τ_k+j} by

$$\sum_{i=1}^{j-1} q_{\{\tau_k+i,\tau_k+j,\dots,\tau_r\}} (z_{\tau_k+j} - z_{\tau_k+i}) \quad - \quad h_{\tau_k+j}.$$

Using the defining relation of divided differences, we get

$$q_{\{\tau_k+i,\tau_k+j,\dots,\tau_r\}} (z_{\tau_k+j} - z_{\tau_k+i}) = q_{\{\tau_k+i,\tau_k+j+1,\dots,\tau_r\}} - q_{\{\tau_k+j,\tau_k+j+1,\dots,\tau_r\}}.$$

With the definition of h_{τ_k+j} in (B.1), the new entry at index $\tau_k + j$ simplifies as $-j q_{\{\tau_k+j,\tau_k+j+1,\dots,\tau_r\}}$. When we multiply the resulting vector by $\mathbf{C}_{\tau_k+j}$, we affect only entries from indices $\tau_k + 1$ to $\tau_k + j$. More precisely, the previous relation shows that we obtain the vector

$$\big(q_{\{1,\tau_k+j,\dots,\tau_r\}}, \dots, q_{\{\tau_k,\tau_k+j,\dots,\tau_r\}}, q_{\{\tau_k+1,\tau_k+j+1,\dots,\tau_r\}}, \dots, q_{\{\tau_k+j,\tau_k+j+1,\dots,\tau_r\}}, \\ h_{\tau_k+j+1}, \dots, h_{\tau_r}\big).$$

Finally, right-multiplication by $\mathbf{D}_{\tau_k+j}$ affects entries of indices $1, \dots, \tau_k$. For $i = 1, \dots, \tau_k$, it replaces $q_{\{i,\tau_k+j,\dots,\tau_r\}}$ by

$$q_{\{i,\tau_k+j,\dots,\tau_r\}} (z_{\tau_k+j} - z_i) + q_{\{\tau_k+j,\tau_k+j+1,\dots,\tau_r\}} = q_{\{i,\tau_k+j+1,\dots,\tau_r\}}.$$

Thus, the resulting vector is

$$\big(q_{\{1,\tau_k+j+1,\dots,\tau_r\}},\dots,q_{\{\tau_k,\tau_k+j+1,\dots,\tau_r\}},q_{\{\tau_k+1,\tau_k+j+1,\dots,\tau_r\}},\dots,q_{\{\tau_k+j,\tau_k+j+1,\dots,\tau_r\}},\\ h_{\tau_k+j+1},\dots,h_{\tau_r}\big)$$

which is precisely $\boldsymbol{q}_{\tau_k+j}$, as claimed in (B.2).

C. Proof of Lemma 34

To simplify our notation, for all $1 \le s \le \ell$, we abbreviate $\eta_{\ell-s}(d-1,\dots,d-\ell)$ to $g_{\ell-s}$. Then, we claim that one has

$$g_{\ell-s} < d(d-1)\cdots(d-\ell+1).$$

Indeed, let $f(t)=(t+d-1)(t+d-2)\cdots(t+d-\ell)$, so that $f(1)=d(d-1)\cdots(d-\ell+1)$. From Vieta's formula we have

$$f(t)=\sum_{s=0}^{\ell} g_{\ell-s}\,t^s$$

and so we also have $f(1)=\sum_{s=0}^{\ell} g_{\ell-s}$. Therefore,

$$d(d-1)\cdots(d-\ell+1)=\sum_{s=0}^{\ell} g_{\ell-s}$$

and so $g_{\ell-s} < d(d-1)\cdots(d-\ell+1)$ for all $1 \le s \le \ell$.

Now, for any partition $\lambda=(n_1^{\ell_1}\,\dots\,n_r^{\ell_r}) \vdash n$ of length ℓ_λ, we denote by $w_\lambda=\prod_{i=1}^r \ell_i!$. Then we have

$$\mathfrak{c}_\lambda = d^s\,\frac{g_{\ell_\lambda-s}}{w_\lambda} = d^s\,\frac{\ell_\lambda!}{\prod_{i=1}^r \ell_i!}\,\frac{g_{\ell_\lambda-s}}{\ell_\lambda!} = d^s h(\lambda)\,\mathscr{F}_{d,\ell_\lambda,s},$$

where $h(\lambda)=\frac{\ell_\lambda!}{\prod_{i=1}^r \ell_i!}=\binom{\ell_\lambda}{\ell_1,\dots,\ell_r}$ and $\mathscr{F}_{d,\ell_\lambda,s}=\frac{g_{\ell_\lambda-s}}{\ell_\lambda!}$. From our previous inequality we have

$$\mathscr{F}_{d,\ell_\lambda,s} \le \frac{d(d-1)\cdots(d-\ell_\lambda+1)}{\ell_\lambda!} = \binom{d}{\ell_\lambda}$$

and so

$$\sum_{\lambda\vdash n,\,\ell_\lambda\ge s} \mathfrak{c}_\lambda \le d^s\left(\sum_{\lambda\vdash n,\,\ell_\lambda\ge s} h(\lambda)\binom{d}{\ell_\lambda}\right). \tag{C.1}$$

Let $\mathbf{a}$ be a sequence of $m+1$ numbers $(a_0,a_1,\dots,a_m)$ and let $p_{\mathbf{a}}(t)=\sum_{i=0}^m a_i\,t^i$ be its generating polynomial. The *polynomial coefficients* associated to $\mathbf{a}$ are defined by

$$\binom{k}{n}_{\mathbf{a}} = \begin{cases} [t^n]\,(p_{\mathbf{a}}(t)^k), & \text{if } 0\le n\le mk\\ 0, & \text{if } n<0 \text{ or } n>mk \end{cases}$$

where $[t^n]\sum_i c_i t_i = c_n$ is the coefficient of t^n in the series $\sum_i c_i t_i$. For any partition λ of n, let further λ' be its conjugate partition. By [20, Lemma 2.1], we have

$$\binom{k}{n}_{\boldsymbol{a}} = \sum_{\substack{\lambda\vdash n,\\ \ell_{\lambda'}\le n}} a_0^{k-\ell_{\lambda'}}\,h(\lambda)w_{\boldsymbol{a}}(\lambda)\binom{k}{\ell_\lambda}, \tag{C.2}$$

where $w_{\boldsymbol{a}}(\lambda)$ is the function $w_{\boldsymbol{a}}(\lambda) = \prod_{i=1}^{m} a_i^{\ell_i}$, and ℓ_λ and $\ell_{\lambda'}$ are the respective lengths of λ and λ'. If we consider $m = n$, $\boldsymbol{a} = (1, \ldots, 1) = \mathbf{1}$, and $k = d$, then equation (C.2) becomes

$$\binom{d}{n}_{\mathbf{1}} = \sum_{\substack{\lambda \vdash n, \\ \ell_{\lambda'} \leq n}} h(\lambda) \binom{d}{\ell_{\lambda'}}.$$

For any partition λ of n, the length of its conjugate satisfies $\ell_{\lambda'} \leq n$ and so

$$[t^n](1 + t + \cdots + t^n)^d = \binom{d}{n}_{\mathbf{1}} = \sum_{\lambda \vdash n} h(\lambda) \binom{d}{\ell_\lambda}. \tag{C.3}$$

Furthermore,

$$(1 + t + \cdots + t^n)^d = (1 - t^{n+1})^d \, (1 - t)^d = \Big(\sum_{k=0}^{d} (-1)^k \binom{d}{k} \, t^{(n+1)k} \Big) \Big(\sum_{i=0}^{\infty} \binom{d + i - 1}{i} t^i \Big),$$

where t^n appears only when $k = 0$ and $i = n$. In other words,

$$[t^n] \, (1 + t + \cdots + t^n)^d = \binom{n + d - 1}{n}. \tag{C.4}$$

Combining (C.1), (C.3) and (C.4), gives

$$\sum_{\lambda \vdash n, \, \ell_\lambda \geq s} \mathfrak{c}_\lambda \leq d^s \left(\sum_{\lambda \vdash n} h(\lambda) \binom{d}{\ell_\lambda} \right) \leq d^s \binom{n + d - 1}{n}.$$

Similarly, we can prove the inequality $\sum_{\lambda \vdash n, \, \ell_\lambda \geq s} \mathfrak{c}_\lambda \leq n(d + 1)^s \binom{n+d}{n}$.