AN ALGORITHMIC APPROACH TO CHEVALLEY’S THEOREM ON IMAGES
OF RATIONAL MORPHISMS BETWEEN AFFINE VARIETIES

MOHAMED BARAKA T AND MARKUS LANGE-HEGERMANN

ABSTRACT. The goal of this paper is to introduce a new constructive geometric proof of the
affine version of Chevalley’s Theorem. This proof is algorithmic and a verbatim implementa-
tion resulted in an efficient code for computing the constructible image of rational maps
between affine varieties. Our approach extends the known descriptions of uniform matrix
product states to $uMPS(2, 2, 5)$.

1. INTRODUCTION

Theorem 1.0 (Chevalley). Let $f : X \to Y$ be rational map of affine varieties and $C$ a
constructible subset of $X$. Then the image $f(C)$ is again constructible.

This paper develops a geometric proof of the above affine version of Chevalley’s The-
orem which is simultaneously a description of an efficient algorithm for computing the
constructible image $f(C)$. The general case of Chevalley’s Theorem reduces to the affine
case.

The central idea of all geometric and algebraic proofs of Chevalley’s Theorem we are
aware of is the use of the following Theorem 1.1 which ensures the existence of a proper
closed subset $D$ of $f(C)$ containing the (constructible) set $\overline{f(C)} \setminus f(C)$ of all points outside
of the image:

$$\overline{f(C)} \setminus f(C) \subseteq D \subsetneq \overline{f(C)}.$$

We call such a subset $D$ a relative boundary hull of $f(C) \subseteq Y$ (cf. Section 2).

Theorem 1.1. Let $f : X \to Y$ be a rational map of affine varieties and $C$ a Zariski-
closed subset of $X$. Then the image $f(C)$ admits a relative boundary hull $D$.

This already suffices to prove the following special form of Chevalley’s Theorem, which
in turn easily implies Theorem 1.0 by a generalized version of Rabinowitsch trick, as shown
in Subsection 3.1.

Theorem 1.2 (Chevalley). Let $f : X \to Y$ be a rational map of affine varieties and $C$ a Zariski-closed subset of $X$. Then the image $f(C)$ is constructible.

All proofs of Theorem 1.2 basically use the same iterative algorithm and only differ in
their choice of the relative boundary hull $D$ of $f(C)$ constructed in the proof of Theorem 1.1.
Indeed, the choice of $D$, in particular the number of its irreducible components and their
degrees have a profound impact on the computational efficiency of the resulting iteration.

2010 Mathematics Subject Classification. 13P10, 13P15, 68W30, 14Q20, 14R20.
Key words and phrases. Constructible images, Chevalley’s Theorem, Gröbner bases, elimination.
This work is a contribution to Project II.1 of SFB-TRR 195 ‘Symbolic Tools in Mathematics and their
Application’ funded by Deutsche Forschungsgemeinschaft (DFG).

1By “image” we always mean the set-theoretic image. The scheme-theoretic image is closed by definition;
it is the vanishing locus of the elimination ideal of the vanishing ideal of the graph of $f$. 

1
The main reduction step in this iterative algorithm relies on the properness of \( D \) as a subset of \( \overline{f(C)} \). More precisely, the projection formula \( f(f^{-1}(D) \cap C) = D \cap f(C) \) yields the equation

\[
(\psi) \quad f(C) = \left( \overline{f(C)} \setminus D \right) \cup (D \cap f(C)) = \left( \overline{f(C)} \setminus D \right) \cup f(f^{-1}(D) \cap C),
\]

which decomposes \( f(C) \) into the (disjoint) union of the \textit{locally closed} subset \( \overline{f(C)} \setminus D \) and the image under \( f \) of the strictly smaller subset \( f^{-1}(D) \cap C \subseteq C \) of points of \( C \) lying over the relative boundary hull \( D \). The termination of the iteration is guaranteed by the Noetherianity of \( X \) or of \( Y \). This proves Theorem 1.2 as a corollary of Theorem 1.1.

In Section 3 we recall how the proof of Theorem 1.1 reduces to the construction of a relative boundary hull of the image of a closed subset \( \Gamma \subseteq \mathbb{A}^n_Y \) under the base-projection \( \pi : \mathbb{A}^n_Y \to Y \).

The algorithmic heart of this paper is Section 4, where a new relative boundary hull of \( \pi(\Gamma) \) is constructed in Algorithm 4. The idea of our construction is to replace \( \Gamma_0 \subseteq \mathbb{A}^n_Y \) by a closed subset \( \Gamma_0 \subseteq \Gamma \), that has the same projection closure \( \pi(\Gamma_0) = \pi(\Gamma) \) as \( \Gamma \), but where the restriction \( \pi|_{\Gamma_0} \) has generically zero-dimensional fibers. The generic zero-dimensionality of the fibers gives rise to the following argument: Let \( \hat{\Gamma}_0 \) denote the closure of \( \Gamma_0 \subseteq \mathbb{A}^n_Y \) in \( \mathbb{P}^n_Y \) and \( H := \mathbb{P}^n_Y \setminus \mathbb{A}^n_Y \) the hyperplane at infinity. Then the closed image \( D \) of the points \( \Gamma_0^\infty := \hat{\Gamma}_0 \cap H \) at infinity under the closed projection \( \hat{\pi} : \mathbb{P}^n_Y \to Y \) is a relative boundary hull of \( \pi(\Gamma) \).

As mentioned above, constructing a relative boundary hull \( D \) is a central step of all previous proofs of Chevalley’s theorem known to us. To simplify the notation one can assume without loss of generality that \( \pi|_{\Gamma} \) is dominant, i.e., that \( \overline{\pi(\Gamma)} = Y \). Most of these approaches construct an open set \( U = Y \setminus D \subseteq Y \) with a property guaranteeing the inclusion \( U \subseteq \pi(\Gamma) \). However, these properties tend to be far too strong, guaranteeing much more than the inhabitedness of the fibers over \( U \). This usually results in an unnecessarily large relative boundary hull of high degree and many irreducible components, slowing down the recursion significantly. One such property is \textit{generic freeness}, where \( \mathcal{O}_{\pi^{-1}(U)} \) is required to be a non-zero free \( \mathcal{O}_U \)-module. The idea of using local freeness was for example exploited in

- [Kem10, Alg. 10.3] by an approach using Gröbner bases,
- [BGLHR12] by an approach based on resultant, or
- Chevalley’s original proof [CC56, Thm. 3].

Similarly, an even more elaborate approach are (canonical) Gröbner covers [MW10], where the preimage \( \pi^{-1}(U) \) can be described by a single Gröbner basis with leading ideal independent of specializations to fibers \( \pi^{-1}(u) \) for \( u \in U \). Alternatively, the \textit{Noether normalization} lemma suggests choosing \( U \) such that \( \pi|_{\pi^{-1}(U)} \) is the composition of a finite surjective morphism \( \pi^{-1}(U) \to \mathbb{A}^n_U \) with the projection \( \mathbb{A}^n_U \to U \), cf., e.g., [GD67, Thm. 1.8.4]. For a more detailed comparison of these and more approaches we refer the reader to Section 5, where we also argue for the efficiency of our algorithm.

\footnote{See Remark 4.12 for a dual point of view.}
The recent approach [HMeS19] is similar to ours in that it specifically targets small relative boundary hulls in a computationally efficient algorithm. It is tailor-made for rational morphisms between projective varieties, whereas our approach is tailor-made for the affine case. For a detailed comparison see Subsection 5.5. The only non-intrinsic step in this and in our approach lies in the reduction $\Gamma \sim \Gamma_0$, i.e., the reduction to the case of (generically) zero-dimensional fibers.

Note that all algorithmic approaches to Chevalley’s Theorem have to start by computing the closure of the image. This relies on elimination and usually remains an expensive first step. Whereas our algorithm theoretically needs primary decompositions, we manage to postpone (or even prevent, cf. Appendix B) primary decompositions; any primary decomposition is computed after passing from $\Gamma$ to $\Gamma_0$ (cf. Subsection 4.1).

An important special case of constructible images are orbits of algebraic group actions $\alpha : Y \times G \to Y$. The orbit $yG$ of $y \in Y$ is the image of the action map $\alpha_y : G \to Y, g \mapsto yg$. Chevalley’s Theorem implies that each such orbit is even locally closed (Corollary 7.2). Group orbits play an important role in the representation theory of semisimple algebraic groups (cf. [Jan04] for nilpotent orbits). Another prominent example is quantum information theory, where group orbits classify entangled states of multiple qubit systems. In Section 7 we show how the special context of algebraic group orbits allows for various improvements in our algorithm (cf. Corollary 7.3 and Proposition 7.6). These improvements speed up computations considerably.

As mentioned above, the Gröbner basis algorithm remains the bottleneck when treating bigger examples. However, we expect that the limits of the Gröbner basis machinery can be pushed much further for $G$-equivariant problems (like the computation of algebraic group orbits) once the $G$-equivariance can be exploited in the algorithm and in the underlying data structures. We will address this in future work.

The structure of the paper is as follows. Section 2 lays the topological framework for constructible sets, which is used in Section 3 to rephrase and prove Chevalley’s Theorem 1.0 in terms of projections. Our relative boundary hull is constructed in Section 4 and compared to other relative boundary hulls in Section 5. In Section 6 we reinterpret (generalizations of) Rabinowitsch’s trick in our setting (cf. Example 6.1). We demonstrate the algorithmic efficiency of our approach on the uniform matrix product states in Example 6.2, significantly improving on the state of the art in [CMS19] by providing an explicit description of $\text{uMPS}(2,2,5)$ and recomputing $\text{uMPS}(2,2,4)$ in seconds without using any additional representation-theoretic arguments. We present a formal Gröbner basis proof for the dimension reduction in Appendix A and show how to completely avoid the primary decomposition in Appendix B. In Appendix C we show how to replace the linear structure of the classical proofs summarized in Algorithm 1 by a more sophisticated branching in Algorithm 15, in which unnecessary branches are removed as early as possible. Such a branched algorithm can easily be parallelized at several places, a planned improvement of our current implementation.

2. CONSTRUCTIBLE SETS

Let $Y$ be a topological space. A subset $A \subseteq Y$ is said to be

- **locally closed** if one of the following equivalent descriptions holds:
  - $A$ is the intersection of an open and a closed subset;
  - $A$ is the difference of two closed subsets;
– $A$ is the difference of two open subsets.
• constructible if it is the finite union of locally closed sets.

We are interested in constructible sets as they describe images of rational maps by Chevalley’s Theorem 1.2.

**Example 2.1.** Let $k$ be a field or even $k = \mathbb{Z}$. The image of the polynomial map

$$
\varphi : \begin{cases}
\mathbb{A}_k^2 \\
(x_1, x_2) \mapsto (b_1, b_2) := (x_1, x_1x_2).
\end{cases}
$$

is the union of $\mathbb{A}_k^2 \setminus \{b_1 = 0\}$ and the origin $\{(0, 0)\}$ (cf. Example 5.7). The image is clearly neither closed, nor open, not even locally closed in the Zariski topology. However, it is constructible.

For a constructible set $A$, we use the following two approximations to the boundary $\partial A$ to inductively approach the intricate details of $\partial A$, leading to efficient algorithms.

**Definition 2.2 (Relative boundary).** The relative boundary $\partial$ of $A \subseteq Y$ is defined by

$$
\partial A := \overline{A \setminus A} = \overline{A \cap (Y \setminus A)} \subseteq \overline{A} \cap \overline{Y \setminus A} =: \partial A.
$$

If $A$ is a nonempty set we call a closed set $D$ a relative boundary hull of $A \subseteq Y$ if

$$
\overline{A} \setminus A \subseteq D \subsetneq \overline{A}.
$$

In other words, $D$ contains $\partial A$ but does not contain $A$. We further define the empty set to be the relative boundary hull of itself.

The notion of a relative boundary hull is introduced in [HMeS19] under the name “frame”.

**Definition 2.3 (Locally closed part).** The locally closed part $LCP(A)$ of an arbitrary subset $A \subseteq Y$ is defined as

$$
LCP(A) := \overline{A \setminus \partial A}.
$$

If $D$ is a relative boundary hull of $A$ then we call $\overline{A} \setminus D$ a locally closed approximation of $A$, the locally closed part being the largest such approximation.

In Example 2.1 the minimal relative boundary hull is the relative boundary $\{b_1 = 0\}$. Relative boundary hulls do not always exist, for example for $A := \mathbb{Q} \subset \mathbb{R} =: Y$ equipped with the natural topology the equality $\partial A = \overline{A}$ holds. However, relative boundary hull, or equivalently locally closed approximations, exist for images of rational maps in the Zariski topology by Theorem 1.1, which will be proved in Section 4.

**Definition 2.4 (Canonical form).** If $A$ is a constructible set in the topological space $Y$, then the locally closed part $LCP(A)$ is the largest locally closed subset of $A$. If moreover $Y$
is Noetherian, then the **canonical form** or **canonical decomposition** of a constructible set is the finite **disjoint** union

\[ A = \text{LCP}(A) \uplus \text{LCP}(A \setminus \text{LCP}(A)) \uplus \cdots. \]

In Example 2.1 the locally closed part is \( \mathbb{A}^2 \setminus \{ b_1 = 0 \} \), and the canonical form is

\[ \mathbb{A}^2 \setminus \{ b_1 = 0 \} \uplus \{ (0,0) \}. \]

**Remark 2.5.** The canonical form is in an obvious sense the “most exhaustive” decomposition of a constructible set into locally closed subsets. It is by definition an **intrinsic** decomposition of a constructible set and as such of general interest independent of the context of this paper. We will therefore defer its algorithmic treatment to a later paper, where we also show its finiteness. We have already implemented all operations of the Boolean algebra of constructible objects in a locale including the computation of the canonical decomposition in the GAP package **Locales** [BKLH19a]. The package **Locales** uses the philosophy of category constructors [Pos19] and relies on the CAP project which makes categories accessible to the computer [GPS18, GP19].

**Remark 2.6.** Let \( A = \biguplus_{i=1}^n A_i \setminus B_i \) be the canonical decomposition of the constructible set \( A \). The tuple \((A_1, B_1, \ldots, A_n, B_n)\) of closed subsets is intrinsically associated to \( A \). Hence, any invariant of closed sets induces a \( 2n \)-tuple of invariants when applied to the above tuple. The dimension, the degree, and the Euler characteristic are such invariants. An even finer invariant which incorporates all these three invariants is the Chern-Schwartz-MacPherson class of an affine or projective variety [Ple09, MB12, Jos15, PB14, Bäc14].

All known image algorithms can a priori only compute relative boundary hulls instead of relative boundaries for the yet unknown image, at least when the morphism has non-zero dimensional fibers. This means that they can a priori only approximate the canonical decomposition by successive locally closed approximations (LCA, cf. Algorithm 4)

\[ A = \text{LCA}(A) \uplus \text{LCA}(A \setminus \text{LCA}(A)) \uplus \cdots. \]

The result is thus a non-exhaustive decomposition of the constructible image\(^3\). More precisely:

The decomposition in Formula (\( \uplus \)) approximates the canonical form by replacing the locally closed part \( \text{LCP}(f(C)) \) by a locally closed approximation \( \text{LCA}(f(C)) := f(C) \setminus D \) of the image \( f(C) \), as described above.

The canonical form can be achieved in a single step in the case of algebraic group orbits, see Corollary 7.2 and Corollary 7.3.

### 3. Notational Preliminaries

In this section we fix the notation for the remaining paper, both geometrically and algebraically. The remaining paper mostly uses the language of algebraic geometry. Additionally, we sometimes use the algebraic language both to specify slight restrictions on the general geometric setting to allow for algorithms and to describe some algorithmic improvements.

To compute the image of a map \( f : X \to Y \) of sets it is sufficient to compute the image of the graph \( \Gamma := \Gamma_f := \{(f(x), x) \mid x \in X\} \subseteq Y \times X \) under the projection map

\(^3\)However, once the image is written in this way, the canonical decomposition can then be computed a posteriori (see Remark 2.5).
\[ \pi : Y \times X \to Y. \] Equally general, one can consider the projection of a subset \( \Gamma \subseteq Y \times X \) which is not necessarily the graph of a map.

In our affine setup we consider Zariski-closed subsets \( X \subseteq \mathbb{A}^n_k \) and \( Y \subseteq \mathbb{A}^m_k \) over a commutative coefficients ring \( k \). We write \( \text{Spec } B \) for either \( \mathbb{A}^m_k \) or \( \mathbb{A}^n_k \). If \( X \) and \( Y \) are closed subsets of the relative affine space \( \mathbb{A}^n_B = \text{Spec } B \times_k \mathbb{A}^n_k = \text{Spec } R \) for \( R := B[x_1, \ldots, x_n] \). The above projection \( \pi \) is then the restriction of the base-projection \( \mathbb{A}^n_B \to \text{Spec } B \) to \( Y \times X \). If \( X \) is reduced resp. irreducible, then so is \( \Gamma_f \). And if \( \Gamma \) is reduced resp. irreducible, then so is \( \pi(\Gamma) \).

As manifested in our use of the \( \text{Spec} \)-notation our topological spaces are the prime spectra of affine \( k \)-algebras. As customary, this is one of the ways to avoid the assumption that \( k \) is an algebraically closed field, which is far too restrictive for our intended applications. However, when our algorithm reduces to the case that \( k \) is a finite field we will occasionally pass to a finite extension of \( k \) in order to guarantee the existence of certain hyperplanes. Still, the image \( \pi(\Gamma) \) will be a constructible subset of the topological space \( \text{Spec } B \), where \( B \) is the original \( k \)-algebra.

Note that non-reduced structures play no role in our setup and we may assume all ideals to be radical, i.e., all affine varieties to be reduced.

Our goal is an algorithm, hence we assume that the rings \( k \), \( B \), and \( R \) allow for Gröbner basis algorithms. One general framework to allow for Gröbner bases is sketched in Subsection 3.2, which allows for a wide range of rings \( k \) like affine algebras over a constructive field or \( \mathbb{Z} \). We also refer to Subsection 3.2 for a discussion about rings over which our algorithm is possible, even though these rings do not allow Gröbner bases. We treat Gröbner bases as black boxes, in particular we are agnostic to term orders, except for the occasional need of elimination orders, and do not comment on possible optimization of Gröbner bases for input of certain types. For computational efficiency we also assume the existence of a primary decomposition algorithm, even though this is not strictly necessary (cf. Appendix B).

If
\[
\begin{align*}
\pi : Y \times X & \to Y, \\
\left( x_1, \ldots, x_n \right) & \mapsto \left( b_i := (f_i(x_1, \ldots, x_n)) \right)
\end{align*}
\]
is a rational map\(^4\) (or a polynomial map for \( q_i \equiv 1 \)), then its graph \( \Gamma_f \) is given by the ideal
\[
I := \langle q_i(x_1, \ldots, x_n)b_i - p_i(x_1, \ldots, x_n) \mid i = 1, \ldots, m \rangle \subseteq R.
\]

Encoding maps in projections we can reformulate Theorem 1.1 and Theorem 1.2 as follows:

**Theorem 3.1.** \( \pi(\Gamma) \) admits a relative boundary hull, or equivalently, a locally closed approximation.

**Theorem 3.2** (Chevalley). \( \pi(\Gamma) \) is a constructible set.

The proof of Theorem 3.2 is again a reformulation of the proof of Theorem 1.2 given in the Introduction. We present it in form of the following algorithm, which we express geometrically as Algorithm 1 and, with corresponding lines, algebraically as Algorithm 2.

---

\(^4\)For rational maps we assume \( k \) to be a field.
Note that the used sub-algorithm `LocallyClosedApproximationOfProjection` depends on the approach used to compute the relative boundary hull:

- Our new approach is expressed geometrically as Algorithm 4 and algebraically as Algorithm 5, where the respective algorithms have matching line numbers to indicated the correspondence between geometry and algebra.
- Algorithm 6 [Kem10, Alg. 10.3] uses Gröbner bases to construct a relative boundary hull $D$ where $\Gamma$ has free fibers on the complement $U = \overline{\pi(\Gamma)} \setminus D$.

**Algorithm 1:** Outer induction computing a constructible projection (geometric)

```plaintext
Input: A closed subset $\Gamma \subseteq \mathbb{A}^n_B$ with $\pi: \mathbb{A}^n_B \to \text{Spec } B$
Output: $\pi(\Gamma) = C$ as a finite disjoint union of locally closed subsets of $\text{Spec } B$

ConstructibleProjection ($\Gamma$)

1. $C := \emptyset$
2. while $\Gamma \neq \emptyset$ do
   3. $A, D := \text{LocallyClosedApproximationOfProjection}(\Gamma)$  // $A := \pi(\Gamma)$
   4. $C := C \uplus (A \setminus D)$
   5. $\Gamma := \Gamma \cap \pi^{-1}(D)$
3. return $C$
```

**Algorithm 2:** Outer induction computing a constructible projection (algebraic)

```plaintext
Input: $I \subseteq B[x_1, \ldots, x_n]$ with $\Gamma = V(I) \subseteq \mathbb{A}^n_B$ with $\pi: \mathbb{A}^n_B \to \text{Spec } B$
Output: A finite list $C := ((I_1, J_1), (I_2, J_2), \ldots)$ of pairs of ideals in $B$ such that $\pi(\Gamma) = \bigcup (V(I_i) \setminus V(J_i))$

ConstructibleProjection ($I$)

1. $C := ()$
2. while $1 \notin I$ do
   3. $I_A, I_D := \text{LocallyClosedApproximationOfProjection}(I)$
   4. Add($C, (I_A, I_D)$)
   5. $I := I + \langle I_D \rangle_{B[x_1, \ldots, x_n]}$
3. return $C$
```

**Remark 3.3.** All ideals appearing in the algebraic versions of the algorithms (Algorithms 2, 5, and 6) are given by a finite set of generators. This is by specification for the input ideals and by construction for all ideals computed within the algorithms. It is well-known how to use Gröbner bases and syzygies in order to translate operations like ideal membership test, ideal sum, and ideal quotient into operations involving the corresponding finite sets of generators.

**Remark 3.4.** In practice it is extremely useful to exploit known decompositions of the relative boundary hull $D$, and occasionally even to compute a complete decomposition in irreducibles components. To this end, the simple linear structure of iteratively constructing $C$ in Algorithm 1 will be replaced by the directed graph structure in Algorithm 15 which we will develop in Appendix C.

The more general version of Chevalley’s Theorem 1.0 can now be easily and constructively derived from Theorem 3.2:
3.1. Proof of Theorem 1.0. If \( \Gamma \) is constructible as the finite union of locally closed subsets \( \Gamma_i \), then \( \pi(\Gamma) = \bigcup \pi(\Gamma_i) \). Using a generalized version of Rabinowitsch’s trick, each locally closed \( \Gamma_i \subseteq Z = \bar{Y} \times X \) can replaced by a so-called Rabinowitsch cover which is a closed subset in \( \Gamma_i^{\text{r}} \subseteq Z \times A^1 \xrightarrow{\pi'} Z \) with \( \pi'(\Gamma_i^{\text{r}}) = \Gamma_i \) (cf. Equation (1) in Example 6.1). The image \( \pi(\Gamma_i) \) can now be computed as the image of \( \Gamma_i^{\text{r}} \) under the composed projection \( \pi \circ \pi' : Y \times X \times A^1 \to Y \).

\[ \square \]

3.2. Computability of rings and Gröbner bases. In this paper, we will explicitly or implicitly assume that the three rings \( k, B := k[b_1, \ldots, b_m]/\mathcal{I}(Y) \), and \( R := B[x_1, \ldots, x_n] \) allow for Gröbner bases. This will be necessary to compute elimination ideals in Algorithm 5 or in various places to decide ideal membership, compute ideal quotients and saturations.

Gröbner bases exist over many rings. One common construction assumes that \( k \) is a ring with effective coset representatives [AL94, §4.3], i.e., if for every ideal \( J \subseteq k \) in the commutative coefficients ring \( k \) we can determine a set \( T \) of coset representatives of \( k/J \), such that for every \( a \in k \), \( k + J = t + J \). Many rings have this property, e.g., constructive fields and \( \mathbb{Z} \). Furthermore, if \( k \) has effective coset representatives, then affine rings over \( k \) have effective coset representatives as well. Hence, if \( k \) has effective coset representatives, then there exist Gröbner bases in \( B := k[b_1, \ldots, b_m]/\mathcal{I}(Y) \) and therefore also in \( R := B[x_1, \ldots, x_n] \). In Appendix A we also require the decomposition of \( k \) into a finite product of domains \( k = k_1 \times \cdots \times k_r \) to be constructive.

Theoretically, we could even apply our algorithm over rings without Gröbner bases, as it suffices to assume computability for the rings \( k, B, R \). We call a (unital) commutative ring \( R \) computable [BLH11a] if there exists an algorithm to solve a linear system over \( R \), i.e., to find an (affine) generating set of all \( X \) with \( b = XA \) for given matrices \( A \) and \( b \) over \( R \). Of course, any ring with Gröbner bases is computable, but also their residue class rings and certain localizations thereof [BLH11a, BLH11b, Pos18]. Computability obviously allows to decide ideal membership (a particular solution of a linear system) and to compute ideal quotients and saturations (solutions of homogeneous systems), but also to compute elimination ideals [BLHP17].

According to our experience a primary decomposition algorithm is beneficial for computational efficiency, but otherwise not necessary (cf. Appendix B).

4. Existence of locally closed approximations and relative boundary hulls

In this section, we present our alternative proof of Theorem 3.1, i.e., the existence of locally closed approximations of images of projections or, equivalently, of relative boundary hulls. We first present a sketch of a purely geometric algorithm for our construction and then explain the algorithm in detail in Subsections 4.1 and 4.2.

Idea 1: Take the projective closure of each fiber. To understand the origin of the superfluous points \( \pi(\Gamma) \setminus \pi(\Gamma) \), we “add the points at infinity”, i.e., we replace \( \Gamma \subseteq A^n_B \subseteq \mathbb{P}^n_B \) (\( n > 0 \)) by its Zariski closure \( \hat{\Gamma} \) in \( \mathbb{P}^n_B \) and replace \( \pi : A^n_B \to \text{Spec } B \) by the extended projection \( \hat{\pi} : \mathbb{P}^n_B \to \text{Spec } B \). We call \( \hat{\Gamma} \subseteq \mathbb{P}^n_B \) the projective closure of \( \Gamma \). On the one hand \( \hat{\pi} \) is a closed morphism by the main theorem of elimination theory, hence the image of the closed subset \( \hat{\Gamma} \) is closed. One the other hand \( \hat{\pi}(\hat{\Gamma}) = \pi(\Gamma) = \pi(\Gamma) \) due to the continuity of
and the fact that $\hat{\Gamma}$ is the closure of $\Gamma$ as a subset of $\mathbb{P}_B^n$. It immediately follows that

\[
\big(\ast\big) \quad \hat{\pi}(\Gamma) \setminus \pi(\Gamma) \subseteq \hat{\pi}(\Gamma \setminus \Gamma) \subseteq \hat{\pi}(\Gamma),
\]

where $\Gamma^\infty := \hat{\Gamma} \setminus \Gamma = \hat{\Gamma} \cap H$ is the closed set of points of $\hat{\Gamma}$ at infinity and $H := \mathbb{P}_B^n \setminus A^n_B$ is the hyperplane at infinity.

**Example 4.1.** Consider the hyperbola $\Gamma = \Gamma_0 := \mathcal{V}(bx - 1)$ in $A^1_B$ over $B = \mathbb{Q}[b]$. The image under the projection $\pi : \text{Spec}(A^1_B) \rightarrow \text{Spec}(B)$ is $\text{Spec}(B) \setminus \{0\}$. When approaching the missing point $0 \in \pi(\hat{\Gamma}) \setminus \pi(\Gamma)$ in the base, the hyperbola goes to “infinity” in the fiber.

**Idea 2: Approximate the image in its closure by computing a relative boundary hull.** Our goal is to determine $\pi(\Gamma)$ as the difference of the right and the left hand side in Equation (\ast). In the optimal case where the first inclusion is an equality, the image is the locally closed subset $\pi(\Gamma) = \hat{\pi}(\Gamma) \setminus \hat{\pi}(\Gamma^\infty)$. This includes $\hat{\pi}(\Gamma^\infty) = \emptyset$ as special instance where the image is closed. In case the first inclusion in (\ast) is strict then an approximation of the superfluous points $\hat{\pi}(\Gamma) \setminus \pi(\Gamma)$ would require the second inclusion to be strict as well. In other words, in all cases we need $\hat{\pi}(\Gamma^\infty)$ to be a relative boundary hull of $\pi(\Gamma)$.

It turns out that Idea 2 is not enough to compute a relative boundary hull, since $\hat{\pi}(\Gamma^\infty)$ might be all of $\hat{\pi}(\Gamma)$.

**Example 4.2.** Reconsider the hyperbola from Example 4.1 in one fiber-dimension higher, i.e., $\Gamma = \mathcal{V}((bx_1 - 1)) \subseteq A^2_B$. In this case every nonempty fiber of $\Gamma$ along the projection $\pi : (b, x_1, x_2) \mapsto b$ is 1-dimensional and the projective closure $\hat{\Gamma}$ contains for each nonempty fiber a new point at infinity (cf. Example 4.9 for a detailed computation). It follows that the second inclusion in (\ast) is an equality and $\hat{\pi}(\Gamma^\infty)$ is not a relative boundary hull for $\pi(\Gamma)$.

**Idea 3: Make the fibers zero-dimensional.** To find a suitable relative boundary hull in Example 4.2 we need to recover the setup of the original hyperbola, in which the dimensions of the nonempty fibers were (generically) 0-dimensional. This can be achieved by replacing $\Gamma$ with the intersection $\Gamma_0 := \Gamma \cap L$, where $L \subseteq A^n_B$ is an affine subspace of appropriate dimension such that the restriction of $\pi$ to $\Gamma_0$ has generically 0-dimensional fibers but still $\pi(\Gamma_0) = \pi(\Gamma)$, set-theoretically\(^6\). Again denoting by $\Gamma^\infty_0 := \hat{\Gamma}_0 \cap H$, the above guarantees the strict inclusion

\[
\hat{\pi}(\Gamma) \setminus \pi(\Gamma) \subseteq \hat{\pi}(\Gamma^\infty_0) \subsetneq \hat{\pi}(\Gamma).
\]

\(^5\)The construction of this relative boundary hull is closely connected to our results in [BLHP17], which show that saturations and eliminations are equivalent in a certain sense.

\(^6\)The equality is not a scheme-theoretic equality.
and \( \hat{\pi}(\Gamma_0^\infty) \) is now a relative boundary hull of the constructible set \( \pi(\Gamma) \). In other words

\[
\overline{\pi(\Gamma_0)} \setminus \hat{\pi}(\Gamma_0^\infty)
\]

is a locally closed approximation of \( \pi(\Gamma) \). It can be shown that one can choose \( L \) to be constant over the base \( \text{Spec} \, B \).

**Idea 4: Make \( \Gamma \) irreducible.** The affine subspace \( L \) as described above does not necessarily exist if \( \Gamma \) has two components of different dimensions. In that case, we decompose \( \Gamma = \Gamma_1 \cup \ldots \cup \Gamma_c \) into its irreducible components and intersect each component \( \Gamma_i \) independently with an \( L_i \). For computational efficiency, we try to prevent computing a primary decomposition, if possible, by using heuristics (cf. Subsection 4.1). A complete avoidance of primary decompositions is possible (cf. Appendix B), but not algorithmically prudent.

### 4.1. Achieving zero-dimensional fibers.

In this subsection, we explicitly describe an algorithm to replace the closed set \( \Gamma \subseteq \mathbb{A}^n_B \) by a suitable closed subset having locally zero-dimensional fibers under the projection \( \pi : \mathbb{A}^n_B \rightarrow \text{Spec} \, B \), cf. Algorithm 3. More specifically, we compute

1. a closed subset \( \Gamma_0 \subseteq \Gamma \), such that
2. \( \Gamma_0 \) has locally 0-dimensional \( \pi \)-fibers, i.e., over an open set of a component of \( \overline{\pi(\Gamma)} \), but still
3. \( \pi(\Gamma_0) = \pi(\Gamma) \), i.e., without altering the projection closure.

The construction of \( \Gamma_0 \) works as follows. Assume \( \Gamma \subseteq \mathbb{A}^n_B \) irreducible. Denote by \( d = \dim(\Gamma) - \dim(\pi(\Gamma)) \), the generic dimension of the fibers of \( \pi|_{\Gamma} \). For a generic affine subspace \( L \subseteq \mathbb{A}^n_B \) of dimension \( n - d \) define \( \Gamma_0 := \Gamma \cap L \). For a generic \( p \in \pi(\Gamma) \) it holds that \( \dim(\pi^{-1}_|_{\Gamma_0}(\{p\})) = 0 \), since the codimensions of \( L \) and \( \pi^{-1}_|_{\Gamma}(\{p\}) \) sum up to \( n \). A formal proof that such an \( L \) can be chosen generically is given in Appendix A. The proof is based on Gröbner bases.

In practice, we not do intersect \( \Gamma \) with a high codimension \( n - d \) subspace \( L \), but rather we intersect \( \Gamma \) iteratively with \( n - d \) hyperplanes. The choice of these hyperplanes is a Las Vegas algorithm, as we can check whether the choice of the hyperplane is suitable w.r.t. assumption (iii), i.e., not reducing the image. For this Las Vegas algorithm, any choice of hyperplane on a Zariski open and dense set is suitable (cf. Appendix A).

**Remark 4.3.** Furthermore, the above construction assumed \( \Gamma \) irreducible, i.e., if \( \Gamma \) is reducible we should compute its irreducible components and treat those independently. However, Algorithm 3 uses two improvements for computational efficiency:

- it applies heuristics before (and often instead of) the primary decomposition, and
- we do not compute the primary decomposition\(^7\) of \( \Gamma \), but of its intersection with hyperplanes (cf. \( \Gamma_0^\prime \) in line 15 of Algorithm 3).

Only a generic hyperplane \( E \) is suitable for an intersection with \( \Gamma \) such that the three assumptions above can be achieved. If the intersection \( \Gamma \cap E \) is not of smaller dimension than \( \Gamma \), then we need to try a new hyperplane \( E \). If the intersection leads to a smaller image closure, i.e., \( \overline{\pi(\Gamma \cap E)} \subseteq \overline{\pi(\Gamma)} \), the hyperplane \( E \) might lead to a decomposition of \( \Gamma \). The first kind of such a decomposition is that the decreasing image leads to a split in the base,\(^7\)

---

\(^7\)This speed-up conforms to theoretical considerations: degree bounds for Gröbner bases are double exponential in its codimension [MR13], hence reducing this codimension should drastically reduce the computational complexity.
i.e., $\pi(\Gamma \cap E)$ is the union of components of $\pi(\Gamma)$, cf. Example 4.4. A second kind of such a decomposition is that the preimage of the image closure leads to a split of $\Gamma$, i.e., $\pi^{-1}(\pi(\Gamma \cap E)) \cap \Gamma$ is the union of components of $\Gamma$, cf. Example 4.5. Finally, if $E$ does not lead to such a decomposition, we discard it as unsuitable, cf. Example 4.6.

Example 4.4. Consider $\Gamma = \{b_1 x = 0, b_1 b_2 = 0\} \subseteq A^1_{k[b_1, b_2]}$ of dimension 2. Taking a hyperplane $E = \{x = 1\}$ leads to $\Gamma \cap E = \{b_1 = 0, x = 1\}$ of dimension 1. However, $\gamma := \pi(\Gamma) = \{b_1 b_2 = 0\}$, whereas $\gamma_1 := \pi(\Gamma \cap E) = \{b_1 = 0\}$, hence we decrease the image contrary to assumption (iii). Luckily, $\gamma_2 := \gamma \setminus \gamma_1$ leads to a decomposition of the image, which induces the decomposition $\Gamma = \Gamma_1 \cup \Gamma_2$ for $\Gamma_1 = \pi^{-1}(\gamma_1) \cap \Gamma$.

Example 4.5. Consider $\Gamma = \{bx = 0\} \subseteq A^1_{k[b]}$ of dimension 1. Taking a random hyperplane $E = \{x = a\}$ for $0 \neq a \in k$ leads to $\Gamma \cap E = \{x = a, b = 0\}$ of dimension 0. However, $\pi(\Gamma) = \text{Spec } k[b]$, whereas $\pi(\Gamma \cap E) = \{b = 0\} \subseteq \text{Spec } k[b]$, hence we decrease the image contrary to assumption (iii). Luckily, $\Gamma_1 := \pi^{-1}(\pi(\Gamma \cap E)) \cap \Gamma = \{b = 0\} \subseteq A^1_{k[b]}$ is a component of $\Gamma = \Gamma_1 \cup \Gamma_2$ for $\Gamma_2 = \Gamma \setminus \Gamma_1 = \{x = 0\}$.

Example 4.6. Consider $\Gamma = \{x_1 - b = 0\} \subseteq A^2_{k[b]}$ of dimension 2. Taking a random hyperplane $E = \{x_1 = a\}$ for $a \in k$ leads to $\Gamma \cap E = \{x_1 = b = a\}$ of dimension 1. However, $\pi(\Gamma) = \text{Spec } k[b]$, whereas $\pi(\Gamma \cap E) = \{b = a\} \subseteq \text{Spec } k[b]$, hence we decrease the image contrary to assumption (iii). Here, we just took an unlucky hyperplane, but the hyperplane $E = \{x_2 = 0\}$ works.

---

8For reasons of computationally efficiency, we do not use this second decomposition, but it underlies the complete avoidance of primary decompositions in Appendix B.

9This example shows that we would need a full decomposition into irreducible factors, an equidimensional decomposition does not suffice.
We formalize the above approach in Algorithm 3 under the assumption that the coefficients ring \( k \) is an infinite domain and that the composition \( \Gamma \to \text{Spec} \ B \to \text{Spec} \ k \) is dominant. The general case will be reduced to this one in Appendix A.

**Algorithm 3:** ZeroDimensionalFibers

**Input:** A closed subset \( \Gamma \subseteq \mathbb{A}^n_B \to \text{Spec} \ B \). We assume \( k \) to be an infinite domain and that the composition \( \Gamma \to \text{Spec} \ B \to \text{Spec} \ k \) is dominant.

**Output:** A closed affine subset \( \Gamma' \subseteq \Gamma \) and a (possibly empty) list of additional closed affine subsets \([\Gamma_1', \ldots, \Gamma_e']\) defining a closed affine subset \( \Gamma_0 := \Gamma' \cup \Gamma_1' \cup \ldots \cup \Gamma_e' \subseteq \Gamma \), such that (ii) and (iii) are satisfied; more specifically, the fibers in (ii) are locally zero-dimensional over the subset \( \Gamma_0 \).

ZeroDimensionalFibers \((\Gamma)\)

1. \( \Gamma_0' := \Gamma \) // candidate \( \Gamma_0 \), dimension to be decreased below
2. \( s := 1 \) // step counter
3. \( \ell := [] \) // list of additional components
4. while \( \dim(\Gamma_0') - \dim(\pi(\Gamma_0')) > 0 \) do
5. \( E := \text{RandomHyperplane}(\mathbb{A}^n_B \times_{\text{Spec} \ k} \text{Spec} \ B) \) // cf. Remark 4.7
6. \( \Gamma_0' := \Gamma_0' \cap E \) // intersect with hyperplane
7. if \( \dim(\Gamma_0') < \dim(\Gamma_0) \) then // intersection decreases dimension...
8. if \( \pi(\Gamma_0') \subseteq \pi(\Gamma_0) \) then // ...without reducing the image
9. \( \Gamma_0 := \Gamma_0' \) // found smaller \( \Gamma_0 \)
10. elif \( s > n \) then // split w.r.t. reduced image, avoid early
11. /* try splitting \( \Gamma_0' \) by splitting the base space: */
12. \( \Delta := \pi^{-1}(\pi(\Gamma_0')) \) \( \backslash \pi(\Gamma_0) \) // “complement” of the reduced image
13. if \( \pi(\Gamma_0) \not\subseteq \Delta \) then // if the image reduces nontrivially
14. \( \Gamma_0' := \pi^{-1}(\pi(\Gamma_0')) \cap \Gamma \) // continue with one part
15. \( \ell := \text{Add}(\ell, \pi^{-1}(\Delta) \cap \Gamma) \) // store the other half
16. elif \( s > 4n \) then // very expensive, avoid early
17. /* find components: */
18. \( P := \text{Decomposition}(\Gamma_0') \) // in irreducibles
19. \( \Gamma_0' := P_1 \) // continue with first component
20. \( \ell := \ell \cup P \setminus \{\Gamma_0'\} \) // store remaining components
21. \( s := s + 1 \) // increase number of steps
22. return \([\Gamma_0', \ell]\)

Finding an affine subset with small coefficients is of vital importance for computational efficiency: “complicated” hyperplanes not only impede the current step, but might also lead to “complicated” relative boundary hulls which massively hamper subsequent steps.

**Remark 4.7.** An efficient implementation can choose the hyperplanes in line 5 by the following heuristic approach. First of all, take the affine hyperplane \( E \) to be constant over the base \( \text{Spec} \ B \). The first \( n \) hyperplanes we test are \( \{x_i = 0\}, 1 \leq i \leq n \). Then, try several
hyperplanes of the form \( \{ x_i - a \} \) for “small” elements \( a \in k \). Only if this fails, one could entertain hyperplanes of the form \( \{ x_i - a_1 x_j - a_0 \} \) for “small” elements \( a_0, a_1 \in k \). Note that for finite fields \( k \), it might be necessary to take coefficients \( a_i \) from finite field extensions. We refer the reader to Appendix A for a proof (using Gröbner bases) of the generic existence of such hyperplanes.

We employ the (potentially partial) decomposition \( \Gamma_0 := \Gamma'_0 \cup \Gamma'_1 \cup \ldots \cup \Gamma'_e \subseteq \Gamma \) created by Algorithm 3 to continue with each component \( \Gamma_i \) independently, cf. Remark 4.11 for details.

4.2. Computing the relative boundary hull of the projection. As described in Subsection 4.1, we begin by replacing \( \Gamma \) by

(i) a closed subset \( \Gamma_0 \subseteq \Gamma \), such that
(ii) \( \Gamma_0 \) has locally 0-dimensional \( \pi \)-fibers, but still
(iii) \( \pi(\Gamma_0) = \pi(\Gamma) \).

The next geometric step is to take the projective closure \( \hat{\Gamma}_0 \subseteq \mathbb{P}^n_B \) of \( \Gamma_0 \subseteq \mathbb{A}^n_B \). Now define \( \Gamma_\infty^0 := \hat{\Gamma}_0 \cap H \), the points of \( \Gamma_0 \) at infinity, where \( H := \mathbb{P}^n_B \setminus \mathbb{A}^n_B \) is the hyperplane at infinity. The image \( \hat{\pi}(\Gamma_\infty^0) \) of \( \Gamma_\infty^0 \) under the extended projection \( \hat{\pi} : \mathbb{P}^n_B \to \text{Spec} \, B \) is the desired relative boundary hull.

**Algorithm 4:** Locally closed approximation of projection (geometric)

\begin{itemize}
    \item **Input:** A closed subset \( \Gamma \subseteq \mathbb{A}^n_B \xrightarrow{\pi} \text{Spec} \, B \)
    \item **Output:** The closure \( \pi(\Gamma) \) of the projection together with a relative boundary hull \( D \).
\end{itemize}

\begin{algorithm}
\begin{algorithmic}[0]
    \State \( \Gamma_0 := \text{ZeroDimensionalFibers}(\Gamma) \)
    \State Compute \( \pi(\Gamma_0) \)
    \State Compute \( \hat{\pi}(\Gamma_\infty^0) \)
    \State return \( \pi(\Gamma_0), \hat{\pi}(\Gamma_\infty^0) \)
\end{algorithmic}
\end{algorithm}

We implemented the above geometric algorithm in the GAP package ZariskiFrames [BKLH19b]. It gets automatically translated upon execution into an algebraic algorithm (Algorithm 5) by using the standard dictionary between geometry and algebra, including the following “optimization”:

In order to compute \( \Gamma_\infty^0 := \hat{\Gamma}_0 \cap H \) we would first need to compute the homogenization \( I_0^h \) of the vanishing ideal \( I_0 \) of \( \Gamma_0 \) w.r.t. a new indeterminate \( x_0 \) and then intersect the vanishing locus of \( I_0^h \) with the hyperplane \( \{ x_0 = 0 \} \). We can do the composition of these two steps in an optimized way: just take the monomials of maximal degree (in \( x_1, \ldots, x_n \)) from the polynomials in \( I_0 \). Formally, we denote

\[
    f^{\text{maxdeg}} = \sum_{i \in \mathbb{Z}_{\geq 0}^n, |i| = d} q_i x^i
\]

for \( f = \sum_{i \in \mathbb{Z}_{\geq 0}^n} q_i x^i \in R \), \( q_i \in B \), and \( d = \max_{q_i \neq 0} |i| \) and compute

\[
    I_0^{\text{maxdeg}} := (f^{\text{maxdeg}} \mid f \in I_0) = (f^{\text{maxdeg}} \mid f \in G_0) = \mathcal{I}(\hat{\Gamma}_0 \cap H),
\]
where \( G_0 \) is a Gröbner basis of \( I_0 \) w.r.t. a block elimination ordering with \( x_i \gg B \) for all \( i \). It follows that

\[
I_0^{\text{maxdeg}} = (I_0^h)_{|x_0=0}.
\]

We cannot compute the projection of \( \Gamma \) down to \( \text{Spec} \ B \) by substituting all \( x_i \)'s by zero in \( I_0^{\text{maxdeg}} \), since \( (I_0^{\text{maxdeg}})_{|x_0=\ldots=x_n=0} = (I_0)_{|x_0=\ldots=x_n=0} = I_0 \cap B = I \cap B \), where the second and last equality are [BLHP17, Cor. 2.9]. We rectify this by saturating \( I_0^{\text{maxdeg}} \) w.r.t. the ideal \( \langle x_1, \ldots, x_n \rangle \), which corresponds to the irrelevant locus of the hyperplane at infinity. The vanishing locus of the ideal

\[
I_{\text{rbh}}^\infty := (I_0^{\text{maxdeg}} : \langle x_1, \ldots, x_n \rangle^\infty)_{|x_1=\ldots=x_n=0}
\]
is now a relative boundary hull of the image of \( \pi|_{\Gamma_0} \).

**Algorithm 5:** Locally closed approximation of projection (algebraic)

**Input:** An ideal \( I \) defining an irreducible closed subset \( \Gamma := V(I) \subseteq \mathbb{A}_B^n = \text{Spec} \ R \)

**Output:** An ideal \( I_B \subseteq B \) defining the closure of the projection \( \pi(\Gamma) := V(I_B) \subseteq \text{Spec} \ B \) and an ideal \( I_{\text{rbh}} \subseteq B \) defining a relative boundary hull \( D := V(I_{\text{rbh}}) \) of \( \pi(\Gamma) \).

**LocallyClosedApproximationOfProjection** \((I)\)

0  \hspace{1cm} \text{I}_0 := \text{ZeroDimensionalFibers}(I) /* The following Gröbner basis \( G_0 \) gets computed in the algebraic version of Algorithm 3 (ZeroDimensionalFibers). */

1  \hspace{1cm} G_0 := \text{GröbnerBasis}_{x_i \gg B}(I_0)
   \hspace{1cm} I_B := (G_0 \cap B) = I_0 \cap B = I \cap B

2  \hspace{1cm} I_0^{\text{maxdeg}} := (G_0^{\text{maxdeg}}) := (f^{\text{maxdeg}} | f \in G_0)
   \hspace{1cm} I_{\text{rbh}}^{\infty} := (I_0^{\text{maxdeg}} : \langle x_1, \ldots, x_n \rangle^\infty)_{|x_1=\ldots=x_n=0}

3  \hspace{1cm} \text{return } I_B, I_{\text{rbh}}^{\infty}

The line numbering in Algorithm 5 corresponds to that in Algorithm 4.

**Example 4.8.** Consider again the hyperbola \( \Gamma = \Gamma_0 := \mathcal{V}(bx - 1) \) in \( \mathbb{A}_B^1 \) over \( B = \mathbb{Q}[b] \) from Example 4.1. The monomials of maximal degree are \( I_0^{\text{maxdeg}} = \langle bx \rangle \). This ideal has two components \( \langle x \rangle \) and \( \langle b \rangle \). The projection of the first component \( \langle x \rangle \) corresponds to \( \pi(\Gamma) \) and is removed by the saturation w.r.t. \( x \). The projection \( \langle b \rangle \) of the second component \( \langle b \rangle \) defines a relative boundary hull, even the relative boundary, of the image.

**Example 4.9.** Consider again the hypersurface \( \Gamma := \mathcal{V}(bx_1 - 1) \) in the two-dimensional affine space \( \mathbb{A}_B^2 \) with coordinate ring \( B[x_1, x_2] \) over \( B = \mathbb{Q}[b] \) from Example 4.2. The variety of the ideal \( I_0^{\text{maxdeg}} = \langle bx_1 \rangle \) has all of \( \text{Spec} \ B \) as image under the projection, even after saturation w.r.t. \( \langle x_1, x_2 \rangle \). Intersecting \( \Gamma \) with a generic affine subspace \( x_2 = ax_1 \) with \( a \in k \) yields \( \Gamma_0 = \mathcal{V}(bx_1 - 1, ax_1 - x_2) \), which is up to isomorphism the situation of Example 4.8.

**Example 4.10.** Consider the hypersurface \( \Gamma := \mathcal{V}(b \cdot (x^2 + 1) - x) \) in the one-dimensional affine space \( \mathbb{A}_B^1 \) with coordinate ring \( B[x] \) over \( B = \mathbb{Q}[b] \). The relative boundary hull of \( \pi(\Gamma) \) is given by the variety of the ideal \( \langle bx^2 \rangle : \langle x \rangle^\infty = \langle b \rangle \). Computing further,

\(^{10}\) A single ideal quotient might not be sufficient, as \( \mathcal{V}(bx_1^2 - 1) \) in \( \mathbb{A}_B^1 \) over \( B = \mathbb{Q}[b] \) shows.
\[ \pi((\pi^{-1}(\mathcal{V}(b))) \cap \Gamma) = \mathcal{V}(b). \] Hence, \( \pi(\Gamma) = (\text{Spec } B \setminus \mathcal{V}(b)) \cup \mathcal{V}(b) = \text{Spec } B. \) So, even though the first relative boundary hull is non-trivial, \( \pi_{|\Gamma} \) is surjective.

**Remark 4.11.** The ideal \( I_0 \) returned by **ZeroDimensionalFibers** can be returned as several components. The additional components are returned in the list \( \ell \) in Algorithm 3. In theory, all these components should be intersected to continue. In practice, we just continue with the main component \( \Gamma' \), for which the condition on zero-dimensional fibers was ensured, and treat the remaining components \( \Gamma'_i \) in separate cases. This has a major advantage in computational efficiency, as we can work with smaller components, at the cost of losing disjointness. Disjointness can be restored by additional computations as in Remark 2.5.

**Remark 4.12.** Let \( f : X \rightarrow Y \) be a rational map between affine varieties with positive dimensional fibers. An, in some sense dual, idea to decrease the (generic) fiber dimension of \( f \) is to find a factorization of \( f \)

\[
\begin{array}{ccc}
X & \xrightarrow{\varepsilon} & Z \\
& \downarrow{f} & \downarrow{\tilde{f}} \\
Y & & 
\end{array}
\]

where \( \varepsilon \) is surjective. It follows that \( f \) and \( \tilde{f} \) have the same image and: the higher the generic fiber dimension of \( \varepsilon \), the lower that of \( \tilde{f} \). And once \( \tilde{f} \) has even 0-dimensional fibers (at least generically) Line 0 of Algorithm 5 becomes redundant (cf. Example 6.2). Achieving an epi-mono factorization would be optimal (cf. Subsection 7.2).

5. **Further Approaches to Relative Boundary Hulls**

All constructive proofs of Chevalley’s Theorem known to us use locally closed approximations via relative boundary hulls. We compare our approach to some proofs, many of which have been made algorithmic, and argue for greater efficiency of our algorithm.

5.1. **Comparing the computational efficiency of approaches to Chevalley’s theorem.**

All approaches for the image of a polynomial map make use of Algorithm 1. In particular, these algorithms only differ in their respective subalgorithms which compute the relative boundary hull. To compare the efficiency of these algorithms, we consider the number of iterations in Algorithm 1 and the computation time in each iteration.

The number of iterations in Algorithm 1 is influenced by the codimension of the relative boundary hull \( D \) of \( \pi(\Gamma) \). Generically, the input \( \Gamma := \pi^{-1}(D) \) (line 5 of subsequent steps in Algorithm 1) has the same codimension as \( D \) has in \( \pi(\Gamma) \). Since Algorithm 1 terminates as soon as \( \Gamma \) is empty, a larger codimension of \( D \) is favorable. If \( \pi(\Gamma) \) is irreducible, \( D \) is at least of, but rarely bigger than, codimension one. A relative boundary hull \( D \) with multiple components usually results in a reducible subsequent \( \pi(\Gamma) \). In particular, a relative boundary hull with many components often leads to additional iterations in Algorithm 1.
In each iteration, the computation time of the relative boundary hull is strongly dominated by computing the closure of the image (e.g. by computing the Gröbner basis of line 1 in Algorithm 5). This is true for our and all other algorithmically relevant approaches to the image of a polynomial map. The computation time of this closure in subsequent iterations strongly depends on the degree of relative boundary hulls, as e.g. higher degrees make Gröbner bases significantly slower. To further decrease computation time in later iterations, our algorithm retains sparsity in its input, e.g. by choosing hyperplanes in a sparse way (Remark 4.7), postponing computations after the intersection with (sparse) hyperplanes (Remark 4.3), and by decompositions.

Decompositions of the sets $\Gamma$ (or $\Gamma_0$) increase the number of iterations, which slightly increases the computation time, but they strongly decrease the size of the considered $\Gamma$ in the iteration. The slight increase in the number of iterations can –to a certain extent– be controlled by Appendix C. We always use a decomposition if we know one. As primary decompositions are expensive to compute, we employ heuristics (cf. Remark 4.3, Appendix B, and lines 10 to 17 in Algorithm 3).

Summing up, the efficiency of a proof of Chevalley’s theorem can mostly be read off the degree of its relative boundary hulls.

Many approaches construct a relative boundary hull $D \subseteq Y$ of the image of a (w.l.o.g.) dominant morphism $f : X \rightarrow Y$ by imposing additional strong properties on the open complement $U = Y \setminus D = f(X) \setminus D$, which as a by-product ensures $U \subseteq f(X)$. This deviates the focus from constructing a small relative boundary hull $D$. Instead of ensuring such properties for $U$, our algorithm focuses on constructing points potentially outside of the image, by finding points coming from infinity. Furthermore, our approach is intrinsic in the case of zero-dimensional fibers, whereas previous approaches depend on a choice $\alpha$ (e.g., coordinate system, term order, . . . ), so that the relative boundary hull $D = D(\alpha)$ could be replaced by the smaller $\bigcap_\alpha D(\alpha)$. And even if one succeeds in intersecting over different choices, the additional properties imposed on $U$ force the degree of the relative boundary hull $D$ to be larger than the one produced by our approach (for an explicit comparison see Remark 5.3 and the examples following it). This in turn induces more costly subsequent computations and more iterations in Algorithm 1.

The following remark describes an extreme case of ensuring stronger properties.

**Remark 5.1.** Whereas the outer induction of Chevalley’s proof in Algorithm 1 seems necessary to compute the image of a rational map, some constructions of a relative boundary hull rely on an additional induction on the dimension in order to ensure the generic zero-dimensionality of the fibers of $\Gamma$ and its projections in each step. To this end the base-projection $\pi : A^n_B \rightarrow \text{Spec } B$ is factored into $n$ successive projections $A^n_B \rightarrow A^{n-1}_B \rightarrow \cdots \rightarrow A^1_B \rightarrow A^0_B = \text{Spec } B$. Such an induction on the dimension is for example necessary for all resultant based approaches that need to interpret $A^n_B$ as $A^{n}_{B_{l-1}}$. This adds major computational drawbacks. Already the first projection $A^n_B \rightarrow A^{n-1}_B$ of a closed $\Gamma \subseteq A^n_B$ is in general a constructible subset of $A^{n-1}_B$, introducing relative boundary hull so “big” that even using decompositions often fails due to a combinatorial explosion of the number of cases.

**5.2. Generic freeness approaches.** Generic freeness is at the heart of two constructions of relative boundary hulls: via Gröbner bases and via resultants. A special case of the generic freeness lemma can be stated as follows:
Theorem 5.2 (Generic freeness lemma for affine rings). Let \( R = B[x_1, \ldots, x_n] \) for a commutative Noetherian domain \( B \). Then, for any finitely presented \( R \)-module \( M \) there exists an \( s \in B \) such that the localization \( M_s \) is a free \( B_s \)-module.

Consider \( \Gamma \subseteq \mathbb{A}_B^n \). As mentioned above, we may assume without loss of generality that the projection \( \pi : \mathbb{A}_B^n \to \text{Spec } B \) is dominant when restricted to \( \Gamma \). Then, the module \( M = R/I(\Gamma) \) is free over \( \text{Spec}(B) \setminus \mathcal{V}(s) \), for \( s \in B \) as in the generic freeness lemma. As \( M \) is additionally non-zero due to the dominance, \( \text{Spec}(B) \setminus \mathcal{V}(s) \) is part of the image of \( \Gamma \) under \( \pi \). Hence, \( \mathcal{V}(s) \) is a relative boundary hull.

Local freeness of positive rank is a rather strong property, which is only needed to guarantee non-zero fibers. This leads to computational inefficiencies due to the polynomial \( s \) often being of a very high degree, as not only the points not in the image are described by \( s \), but also the (usually much bigger) locus of non-freeness.

5.2.1. A Gröbner basis approach to generic freeness. Generic freeness is the idea of [Kem10, Alg. 10.3], where Gröbner basis structures over \( R = B[x_1, \ldots, x_n] \) are used to construct a “Gröbner relative boundary hull”, see Algorithm 6 for details. This approach computes the reduced Gröbner basis \( G := \text{GröbnerBasis}_{x_i \gg B}(I) \) w.r.t. an elimination ordering \( x_i \gg B \). Now, a relative boundary hull is given by \( \mathcal{V}(s) \) for

\[
s = \prod_{g \in (G \setminus B)} \text{LC}_B(g),
\]

where \( \text{LC}_B(g) \) denotes the leading coefficient as a polynomial in \( B \) of the polynomial \( g \) when considered as a polynomial in the \( x_i \)'s w.r.t. the chosen monomial order \( x_i \gg B \) in the Gröbner basis calculation. This approach yields a new algorithmic proof of generic freeness for rings with Gröbner bases. Note that this approach even needs to make two choices: the choice of coordinates and the choice of term orders. A special case of this approach for \( B \) a principal ideal domain (specifically \( B = \mathbb{Z} \)) is given in [BV03, Thm. 3.5].

Algorithm 6: Locally closed approximation of projection via freeness (algebraic)

**Input:** An ideal \( I \) defining an irreducible closed subset \( \Gamma := V(I) \subseteq \mathbb{A}_B^n = \text{Spec } R \) where \( R := B[x_1, \ldots, x_n] \).

**Output:** An ideal \( I_B \leq B \) defining the closure of the projection \( \pi(\Gamma) := V(I_B) \subseteq \text{Spec } B \) and an ideal \( I_{\text{rbh}} \leq B \) defining a relative boundary hull \( D := V(I_{\text{rbh}}) \) of \( \pi(\Gamma) \).

**LocallyClosedApproximationOfProjection (I)**

1. \( G := \text{GröbnerBasis}_{x_i \gg B}(I) \)
2. \( I_B := \langle G \cap B \rangle = I \cap B \)
3. \( I_{\text{rbh}} := \langle \prod_{g \in (G \setminus B)} \text{LC}_B(g) \rangle + I_B = \left( \prod_{g \in (G \setminus B)} \langle \text{LC}_B(g) \rangle \right) + I_B \)

The line numbering in Algorithm 6 corresponds to that in Algorithm 5. This relative boundary hull has three obvious advantages:

- its computation does not require generically zero-dimensional fibers as in Line 0 of Algorithm 5;
- it is easy to compute since we need the elimination Gröbner basis anyway for the computation of the ideal \( I_B \) with vanishing locus \( \pi(\Gamma) \);
• the factorization $\prod_{g \in (G \setminus B)} \LC_B(g)$ in Line 2 induces an obvious decomposition of the relative boundary hull (see below).

However, and apart from the non-intrinsic choices necessary for this construction, the resulting relative boundary hull is of equal dimension as the one proposed in Algorithm 5 of this paper but tends to be of much higher degree (and hence much larger). This can be seen as follows:

**Remark 5.3.** Algorithm 6 becomes directly comparable to our new Algorithm 5 when the ambient fiber dimension $n = 1$, i.e., when $R = B[t]$. In that case Line 0 of Algorithm 5 becomes obsolete and Line 2 of Algorithm 5 simplifies to

$$I_{rbh}^\infty \coloneqq I_B + \langle \LC_B(g) \mid g \in (G \setminus B) \rangle = I_B + \sum_{g \in (G \setminus B)} \langle \LC_B(g) \rangle.$$ 

In particular, as $I_{rbh}^\infty = I_B + \sum_{g \in (G \setminus B)} \langle \LC_B(g) \rangle \supseteq I_B + \prod_{g \in (G \setminus B)} \langle \LC_B(g) \rangle = I_{rbh}$, we get the inclusion

$$D^\infty \coloneqq V(I_{rbh}^\infty) \subseteq V(I_{rbh}) =: D$$

for the two relative boundary hulls $D^\infty$ and $D$. Furthermore, the degree of $I_{rbh}$ increases rapidly with the number of Gröbner basis elements in $G \setminus B$.

**Remark 5.4.** To improve Algorithm 6 one can use more computations to replace $I_{rbh}$ with its radical $\sqrt{I_{rbh}}$ at Line 2 in the obvious way: Denote by $M$ the set of (normalized) irreducible factors of all leading coefficients $\LC_B(g)$ for $g \in (G \setminus B)$. Then $\prod_{c \in M} c$ is the square-free part of $\prod_{g \in (G \setminus B)} \LC_B(g)$. Further denote by $N$ be the set of minimal associated primes of all ideals $\langle c, I_B \rangle$. Then $\sqrt{I_{rbh}} = \bigcap_{p \in N} p$. Still, this improvement does not decrease the degree of $D = V(I_{rbh})$ as a variety (i.e., reduced scheme).

The next examples will use the notation of Remark 5.4 and show that the relative boundary hull $D$ produced by Algorithm 6 can be a highly reducible variety of a very high degree in comparison with $D^\infty$. We verify the strict inclusion $D^\infty \subsetneq D$.

**Example 5.5.** Consider the rational curves

$$\varphi_m : \mathbb{A}^1_q \to \mathbb{A}^m_q, t \mapsto (b_i := t^i(t^{2i} + 1) \mid i = 1, \ldots, m),$$

and the vanishing ideal $I_m = \langle b_i - t^i(t^{2i} + 1) \mid i = 1, \ldots, m \rangle$ of the graph of $\varphi_m$. It is obvious that the image of $\varphi_m$ is closed. Let $M_m$ the set of irreducible factors of leading coefficients of $I_m$ in the notation of Remark 5.4. Explicit computations show that $|M_m| = m + 2$ for $m = 2, \ldots, 10$ and $(\deg c \mid c \in M_m) = (2, 1, \ldots, 1)$. One can verify for $m = 1, \ldots, 10$ that already $\langle M_m \rangle = \langle c \mid c \in M_m \rangle \subseteq I_{rbh}^\infty$ is the unit ideal, hence $D^\infty = V(I_{rbh}^\infty) = \emptyset$, and the iteration in Algorithm 1 stops with an empty relative boundary hull, verifying that im $\varphi_m$ is closed. In contrast, the range of degrees of the radical ideals of $\langle c, I_B \rangle$ for $c \in M_m$ and the sum of all degrees can be read of the following table:

| $m$ | 2 | 3 | 4 | 5 |
|-----|---|---|---|---|
| $\min_{c \in M_m} \deg \sqrt{\langle c, I_B \rangle}$ | 2 | 3 | 6 | 7 |
| $\max_{c \in M_m} \deg \sqrt{\langle c, I_B \rangle}$ | 5 | 15 | 15 | 27 |
| $\sum_{c \in M_m} \deg \sqrt{\langle c, I_B \rangle}$ | 13 | 35 | 55 | 87 |

Starting from $m = 6$ at least one of the ideals $\langle c, I_B \rangle$ gets so complicated, that computing its radical with SINGULAR does not terminate in less than an hour. This is the reason why the
above table stops at \( m = 5 \), since we are not even able to compute all degrees in a sensible time.

**Example 5.6.** Considering \( \text{uMPS}(2, 2, 4) \) from Example 6.2 the cardinality of \( M \) is 156 and the sum of degrees of all \( c \in M \) is 632. The degrees of the radical ideals \( \sqrt{\langle c, I_B \rangle} \) for \( c \in M \) range between 6 and 60 with 3792 being the sum of all degrees. **SINGULAR** failed to compute the minimal associated primes of some of these 156 ideals. In stark contrast, the ideal \( I_{\text{rbh}}^\infty \) is a principal prime ideal generated in degree 6 and one can computationally verify the (very) strict inclusion \( D^\infty = V(I_{\text{rbh}}^\infty) \subsetneq V(I_{\text{rbh}}) = D \).

Whereas Algorithm 15 (the refined version of Algorithm 1) in conjunction with Algorithm 5 terminates within seconds for \( \text{uMPS}(2, 2, 4) \) (cf. Example 6.2), already the computation of the first relative boundary hull using Algorithm 6 fails to terminate in less than a day due to the above mentioned degree explosion.

Luckily, the smaller example \( \text{uMPS}(2, 2, 3) \) terminates also in conjunction with Algorithm 6 after a few seconds, but uses much more intermediate steps as shown in the notebook [BLH20b]. A more impressive jump in the number of intermediate steps when using Algorithm 6 is demonstrated in the notebook [BLH20a].

5.2.2. Resultant based methods to generic freeness. Resultant based methods are conceptually simple, which makes them a prime candidate for a theoretically motivated proof of Chevalley’s Theorem, cf. [Har13, Thm. 3.16, Lem. 3.17] or [CC56, Thm. 3] together with [Car56, unnumbered lemma] or [Sta19, Theorem 00FE] or [Har77, Exerc. II.3.19]. The case distinctions necessary to compute an image are usually already implemented [MM99, Wan03, LMMX05], in one case even by the second author [BGLHR12, BLH14].

Resultant based methods replace the projection \( \pi : A^n_B \to \text{Spec} B \) from an \( n \)-dimensional relative affine space \( A^n_B \) to a composition of \( n \) chosen successive projections \( \pi_i : A^n_B \to A^{i-1}_B, n \geq i \geq 1 \) of a one-dimensional affine spaces. Each of these \( n \) projections \( \pi_i \) is described by a univariate polynomial \( p_i \) (considered as equation or inequation), which can be computed via resultant methods. If \( p_i \) is an equation, its leading coefficients and its discriminant then yields (a superset of) the non-free locus in \( A^{i-1}_B \), whereas \( \Gamma \) is free over the remaining (generic) subset of \( \pi_i(\Gamma) \). Of course, we need to recursively consider \( A^{i-1}_B \), which leads to recursively more and more case distinctions.

While Gröbner bases and resultants have comparable degree bounds in the case of two bivariate polynomials [BL10, Vil18, Laz83, Laz85, Buc83], resultants have a significantly worse complexity for larger systems. One reason for this is that resultant based methods emulate the lexicographical term order of Gröbner bases when using the iterated projections, whereas our algorithm is content with any block elimination order. Furthermore, resultant based methods show behavior close to the worst case bounds, whereas Gröbner bases are often able to exploit sparsity in the inputs. As a consequence, resultant based methods compute the closure of the image of a morphism significantly slower than Gröbner bases. Since computing this closure is the bottleneck of our algorithm, resultant based methods often fail to even compute this preliminary step for computing the image of a morphism. For example, the computations of Example 5.5 for \( m = 4 \) do not terminate after a CPU-hour with

- **Triangularize** (Maple package RegularChains [CGL+07]),
- **AlgebraicThomasDecomposition** (Maple package AlgebraicThomas [BGLHR12]),
- **PolynomialMapImage** (Maple package RegularChains [CGL+07]).
• Comprehensive (Maple package AlgebraicThomas [BGLHR12]).

In contrast, Algorithm 1 in conjunction with Algorithm 4 terminates in less than 10 seconds for $m = 15$.

The recursive approach of resultants yields an algorithmic, albeit technically complicated, proof of generic freeness for affine $k$-algebras and suffers from all the drawbacks induced by the successive projections as discussed in Remark 5.1, in addition to choosing an ordered coordinate system. In particular, this approach tends to induce huge relative boundary hulls.

5.3. **Images via Gröbner covers.** Consider $k$ a field. A Gröbner cover is a stratification of $\text{Spec } B$ into locally closed sets $C_i$, such that $\pi^{-1}(C_i) \cap \Gamma$ can be described by a single Gröbner basis with leading ideal independent of specialization of the $b_j$’s to elements of the algebraic closure $\overline{k}$ [MW10]. In particular, the coordinate ring of $\pi^{-1}(C_i) \cap \Gamma$ is free over the coordinate ring of $C_i$ for all $i$, hence we have a much stronger condition than non-empty fibers. The construction of a Gröbner cover is algorithmic, i.e., Gröbner covers are another constructive approach to generic freeness. Gröbner covers gives us the image of a projection on a silver platter: the image $\pi(\Gamma)$ is the union of all locally closed sets $C_i$, where the corresponding Gröbner basis is not $\{1\}$. The computational cost of Gröbner covers is high, as it demands the existence of a uniform Gröbner basis, a much stronger property than only non-zero fibers or generic freeness. Furthermore, coordinates and term orders need to be chosen.

5.4. **Less effective approaches.** There are additional proofs of Chevalley’s Theorem, which are in principle constructive, but have not been suggested as algorithmic.

Some proofs build on the Noether normalization to ensure the existence of preimages outside of a hypersurface of codimension 1 (cf. [GD67, Thm. 1.8.4, Lem. 1.8.5.1], [Bou72, V.§3.1.Cor. 3], [GW10, Lem. 10.18]). This does probably not yield an efficient algorithm, as the Noether normalization in itself is usually rather expensive, leads to a computationally costly base change that removes the sparsity in the input, the additional property of a Noether normalization is rather specific, and hence the resulting hypersurfaces is usually big. The Noether normalization is in some sense dual to our approach: the dimension of the fibers is brought down to zero, not by intersecting with hyperplanes, but by enlarging the base.

An old version of Vakil’s lecture notes [Vak05, 8.4.2] also constructs the relative boundary hull via the projection of points added at infinity. However, similar to resultant based methods, the general setting $\pi : \mathbb{A}^n_B \to \mathbb{A}^0_B = \text{Spec } B$ is reduced to $n$ successive projections $\pi_i : \mathbb{A}^i_B \to \mathbb{A}^{i-1}_B$, $n \geq i \geq 1$ and the projective closure is only taken in this reduced case. However, this approach suffers from the properties described in Remark 5.1. We expect this succession of projections to unnecessarily enlarge the relative boundary hull.

5.5. **The approach of Harris, Michalek, and Sertöz.** Our approach is closely related to [HMeS19]: both make the map more well-behaved (closed resp. everywhere defined) by extending their domains (by infinity resp. an exceptional divisor of a blowup) and get a relative boundary hull as the image of the extension of the domain. An additional similarity is that both approaches need generically zero-dimensional fibers of the map. This necessitates the only sources of choices: the affine subspaces with which we intersect. The main differences lies in the respective settings: our paper works in an affine setting, whereas [HMeS19] works in a projective setting.
These two settings are special cases of one another, where the affine setting is reduced to the projective setting (cf. [HMeS19, §2.1]) and the projective setting can be reduced to the affine setting via

- the stratification $\mathbb{P}^n = \mathbb{A}^n \cup \ldots \cup \mathbb{A}^0$.

- extending a map

$$
f : \begin{cases} 
\text{Spec } k[x_1, \ldots, x_n] &\to \text{Spec } k[b_1, \ldots, b_m], \\
(x_1, \ldots, x_n) &\mapsto (b_i) := (f_i(x_1, \ldots, x_n))
\end{cases}
$$

to

$$
\tilde{f} : \begin{cases} 
\text{Spec } k[x_1, \ldots, x_n, s, t]/(st - 1) &\to \text{Spec } k[b_1, \ldots, b_m], \\
(x_1, \ldots, x_n, s, t) &\mapsto (b_i) := (s \cdot f_i(x_1, \ldots, x_n))
\end{cases}
$$

to include the operation of the one-dimensional torus over $k$, and

- removing the irrelevant locus with vanishing ideal $\langle b_1, \ldots, b_m \rangle$.

In our experience, each of these two algorithms works best when applied in their intended setting, as pressing them into the other framework leads to unnecessary inefficiencies.

**Example 5.7.** Consider the rational map

$$
f : \mathbb{A}^2 \to \mathbb{A}^2 : (x_1, x_2) \mapsto (b_1, b_2) := (x_1, x_1 x_2)
$$

with all of $\mathbb{A}^2$ as closure of the image. The fibers are already generically zero-dimensional.

The approach from this paper considers the graph $\{b_1 - x_1, b_2 - x_1 x_2\} \subseteq \mathbb{A}^2 \times \mathbb{A}^2$. We get a relative boundary hull via

$$
(\langle b_1 - x_1, b_2 - x_1 x_2 \rangle_{\maxdeg}, \langle x_1, x_2 \rangle_{\infty})_{x_1=x_2=0} = \langle b_1 x_2, x_1 \rangle_{x_1=x_2=0} = \langle x_1, b_1 \rangle_{x_1=x_2=0} = \langle b_1 \rangle.
$$

The approach from [HMeS19] considers $f$ extended, but retaining the same image, to

$$
f' : \mathbb{P}^2 \to \mathbb{P}^2 : (x_0 : x_1 : x_2) \mapsto (x_0^4 : x_0^3 x_1 : x_0^2 x_1 x_2).
$$

The blowup $\text{BL}_{x_0} \mathbb{P}^2$ is defined as subset of $\mathbb{P}^2 \times \mathbb{P}^2$ by the $\mathbb{Z}^2$-homogeneous ideal

$$
\langle b_0 - t x_0^4, b_1 - t x_0^3 x_1, b_2 - t x_0^2 x_1 x_2 \rangle \cap \langle k[x_0, x_1, x_2, b_0, b_1, b_2] = \langle b_2 x_0 - b_1 x_2, b_1 x_0 - b_0 x_1 \rangle
$$

in $k[x_0, x_1, x_2, b_0, b_1, b_2]$ with degrees $(1, 0)$ for the $x_i$’s and $(0, 1)$ for the $b_i$’s. The construction of the exceptional divisor $\mathcal{E}$ of this blowup takes the homogeneous ideal

$$
\delta := \left( \langle 0 \rangle + \langle b_2 x_0 - b_1 x_2, b_1 x_0 - b_0 x_1 \rangle \right) \langle x_0^4, x_0^3 x_1, x_0^2 x_1 x_2 \rangle_{\infty} + \langle x_0^4, x_0^3 x_1, x_0^2 x_1 x_2 \rangle
$$

$$
= \langle b_2 x_0 - b_1 x_2, b_1 x_0 - b_0 x_1, x_0^4, x_0^3 x_1, x_0^2 x_1 x_2 \rangle
$$

\[^{11}\text{Beware: the formulas (when interpreted verbatim) only cover the case of homogeneous polynomial maps.}\]

\[^{12}\text{The important special case is } k^* \text{ for a field } k.\]

\[^{13}\text{In the computation of the blowup one can avoid eliminating the auxiliary variable } t \text{ by computing a row-syzygy matrix } s = \begin{pmatrix} 0 & -x_2 & x_0 \\ -x_1 & 0 & x_0 \end{pmatrix} \text{ of } \begin{pmatrix} x_0^3 x_1^2 \\ x_0^2 x_1 x_2 \end{pmatrix} \text{ and then multiply } s \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} b_2 x_0 - b_1 x_2, b_1 x_0 - b_0 x_1 \end{pmatrix}.\]
and removes irrelevant components
\[ ε := (δ : ⟨x_0, x_1, x_2⟩^∞) : ⟨b_0, b_1, b_2⟩^∞ = ⟨b_2x_0 - b_1x_2, b_1x_0 - b_0x_1, x_0^2x_1x_2, x_0^3x_1, b_0x_0^2x_1, x_0^4, b_0^3b_1^2x_1, b_0^3b_1⟩ \]

The image of \( E \) in \( \mathbb{P}^1 \) under the map induced by \( φ' \) is then given by the ideal
\[ ε \cap k[b_0, b_1, b_2] = ⟨b_0^3b_1⟩. \]

This is a relative boundary hull. This is basically the same relative boundary hull as \( ⟨b_1⟩ \) in our approach, just with additional factor \( b_0 \) to remove points at infinity and powers introduced by the projective modeling.

The approach from [HMeS19] needs polynomial maps, whereas our approach can flexibly switch between maps (the general case) and projections (which do not duplicate the indeterminates of the domain).

6. EXAMPLES

Example 6.1 (Rabinowitsch trick revisited). Let \( B \) be an affine algebra, \( J ⊆ B \), and \( p ∈ B \). Consider the locally closed set \( ∆ := \mathcal{V}(J) \setminus \mathcal{V}(p) = \mathcal{V}(J) \setminus \mathcal{V}(⟨J, p⟩) \subseteq \text{Spec} B \). Rabinowitsch described this locally closed set \( ∆ \) as the image of the closed set
\[ Γ := ∆^{\text{rab}} := \mathcal{V}(⟨J, tp − 1⟩) ⊆ \mathbb{A}^1_B = \text{Spec} B[t] \]
under the natural projection \( (π : \text{Spec} B[t] → \text{Spec} B) = \text{Spec}(B → B[t]) \). We call \( Γ := ∆^{\text{rab}} \) the Rabinowitsch cover of \( ∆ \).

Let \( G \) be a Gröbner basis of \( J \) (w.r.t. some global monomial order on \( B \)). Then \( G_0 := G ∪ \{tp − 1\} \) is a Gröbner basis of \( I_0 := ⟨J, tp − 1⟩ \) w.r.t. a block elimination order \( t ≫ B \) on \( B[t] \). Indeed, since the fibers of \( π|Γ \) are singletons or empty, Algorithm 5 verifies that the closure of the projection is given by the ideal \( I_B := ⟨G_0 ∩ B⟩ = ⟨G⟩ = J \) and the relative boundary hull by the ideal
\[ I_{\text{rbh}} := \left(⟨G_0^{\text{maxdeg}}⟩ : t^∞ \right)_{t=0} = \left(⟨G⟩ : t^∞ \right)_{t=0} + \left(⟨tp⟩ : t^∞ \right)_{t=0} = ⟨J, p⟩. \]

Finally, Algorithm 1 states that this is already the entire projection since the ideal \( I + ⟨I_{\text{rbh}}⟩ \) is the unit ideal.

The same argument applies to the iterated Rabinowitsch trick where the locally closed set \( ∆ := \mathcal{V}(J) \setminus \mathcal{V}(p_1 \cdots p_n) = \mathcal{V}(J) \setminus \mathcal{V}(p_1) \setminus \cdots \setminus \mathcal{V}(p_n) \) is isomorphic to the closed Rabinowitsch cover
\[ Γ := \mathcal{V}(⟨J, tp_1 - 1, \ldots, tp_n - 1⟩) ⊆ \mathbb{A}^1_B = \text{Spec} B[t_1, \ldots, t_n] \]
under the natural projection \( π := \text{Spec}(B → B[t_1, \ldots, t_n]) \).

It is well-known that Rabinowitsch trick may also be generalized to general locally closed subsets \( ∆ := \mathcal{V}(J) \setminus \mathcal{V}(⟨q_1, \ldots, q_r⟩) \). This set is obviously the projection of the closed Rabinowitsch cover\textsuperscript{14}
\[ (1) \quad Γ := ∆^{\text{rab}} := \bigcup_{i=1}^r \mathcal{V}(⟨J, tq_i - 1⟩) ⊆ \mathbb{A}^1_B \]

\textsuperscript{14}This was important in the proof of Theorem 1.0 in Subsection 3.1.
under the projection $\pi := \text{Spec}(B \hookrightarrow B[t])$. Although the projection $\pi|_1$ will not be injective in general, it will still have 0-dimensional fibers over its image.

**Example 6.2** (Uniform matrix product states). For a field $k$ and $D, d, N \in \mathbb{Z}_{>0}$ consider the map

$$T_{D,d,N} : \begin{cases} (k^{D \times D})^d & \rightarrow \text{Cyc}^N(k^d) \\ (M_0, \ldots, M_{d-1}) & \mapsto \sum_{0 \leq i_1, \ldots, i_N \leq d-1} \text{tr}(M_{i_1} \cdots M_{i_N}) e_{i_1} \otimes \cdots \otimes e_{i_N}, \end{cases}$$

where $\text{Cyc}^N(k^d) \leq (k^d)^{\otimes N}$ is the subspace of cyclically symmetric tensors. A cyclically symmetric tensor which lies in the image $\text{uMPS}(D, d, N)$ of $T_{D,d,N}$ is called a **uniform $D$-matrix product state**.

Following Remark 4.12 it is desirable to find a factorization of $T_{D,d,N}$

$$(k^{D \times D})^d \xrightarrow{T_{D,d,N}} \text{Cyc}^N(k^d) \xrightarrow{\pi_{D,d}} \text{Z}_{D,d} \xrightarrow{\Gamma_{D,d,N}} \text{T}_{D,d,N},$$

where $\pi_{D,d}$ is surjective with high dimensional fibers.

A candidate for such a space $Z_{D,d}$ is the spectrum of the so-called trace algebra $C_{D,d}$ which is generated by traces of products $\text{tr}(M_{i_1} \cdots M_{i_d})$ and where $M_0, \ldots, M_{d-1}$ are general $d \times d$ matrices over $k$. The trace algebra $C_{D,d}$ is finitely generated by invariant-theoretic arguments. Sibirskii has among other things showed in [Sib68] that $C_{2,2}$ is freely generated by the five traces $s_i := \text{tr}(M_i)$, $s_{ij} := \text{tr}(M_i M_j)$ for $i, j \in \{0, 1\}$. This means that $Z_{2,2} \cong k^5$ and $T_{2,2,N} : (k^{2 \times 2})^2 \rightarrow \text{Cyc}^N(k^2)$ can be replaced by $\Gamma_{2,2,N} : k^5 \rightarrow \text{Cyc}^N(k^2)$ with generic fiber dimension 0 for $N \geq 4$. For more details and background information see [CMS19].

$$T_{2,2,4} : \begin{bmatrix} \mathbb{Q}^8 \\ \mathbb{Q}^2 \end{bmatrix} \rightarrow \mathbb{Q}^6 \cong \text{Cyc}^4(\mathbb{Q}^2) \begin{bmatrix} x_{0000} \\ x_{0001} \\ x_{0011} \\ x_{1001} \\ x_{1011} \\ x_{1111} \end{bmatrix} = \begin{bmatrix} 0 \quad 0 \\ 1 \quad 1 \\ 0 \quad 1 \\ 1 \quad 0 \\ 1 \quad 0 \\ 0 \quad 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} s_1 s_2 + s_0 s_4 + \frac{1}{4} s_2^2 \\ -\frac{1}{8} s_1 s_3 + \frac{1}{2} s_0 s_4 + \frac{1}{4} s_2 s_3 + \frac{1}{2} s_0 s_2 s_4 + \frac{1}{2} s_1 s_4 \\ -\frac{1}{8} s_1 s_2 + \frac{1}{2} s_1 s_3 + \frac{1}{2} s_0 s_4 + \frac{1}{2} s_0 s_2 s_4 + \frac{1}{4} s_1 s_4 \\ -\frac{1}{3} s_2 s_3 + \frac{1}{2} s_1 s_4 + \frac{1}{4} s_0 s_4 + \frac{1}{2} s_0 s_2 s_4 \\ -\frac{1}{3} s_2 s_1 + \frac{1}{2} s_1 s_4 + \frac{1}{4} s_0 s_4 + \frac{1}{2} s_0 s_2 s_4 \end{bmatrix}.$$
where
\[
J := 2^6 \sqrt{11} - 12 x_{0011} x_{1111} x_{0111} + 16 x_{001} x_{0111} x_{1111} + 4 x_{0000} x_{0111} x_{1111} - 8 x_{0000} x_{0001} x_{011} - 8 \text{MOHAMED BARAKA T AND MARKUS LANGE-HEGERMANN}
\]
\[
\text{Our implementation ConstructibleImage finished in less than 15 seconds (using SINGULAR's Gröbner engine in the background [DGPS19] on an Intel Xeon E5-2687W v4). We stopped}
\]
1. totalImage (MACAULAY2 package TotalImage [HMeS19])
2. grobcov (SINGULAR package grobcov.lib [MW10])
3. PolynomialMapImage (Maple package RegularChains [CGL+07])
4. Comprehensive (Maple package AlgebraicThomas [BGLHR12])

after 24 hours.

The image uMPS(2, 2, 5) of \(T_{2,2,5}\) is computed by the command ConstructibleImage in around 6 minutes (using SINGULAR's Gröbner engine in the background [DGPS19]). The result is the (not locally closed but) constructible set

\[\text{uMPS}(2, 2, 5) = \left( V(\tilde{I}) \setminus V(\tilde{P}) \right) \cup V(\tilde{J}),\]

where the output is too big to reproduce here (see the notebook [BLH19b]).

\[\text{The image was computed in [CMS19, Appendix B] with the help of an ad hoc representation theoretic argument.} \]
7. Algebraic group actions

The approach presented in this paper is well-suited to compute orbits of affine algebraic group actions \( \alpha : Y \times G \to Y \), where \( G \) is an affine algebraic group \( G \) and \( Y \) an affine variety. The computation of the \( G \)-orbit of an element \( y \in Y \) is a special case of our setting as the orbit \( yG \) is nothing but the image of the orbit morphism

\[
\alpha_y : G \to Y : g \mapsto yg.
\]

**Proposition 7.1.** A \( G \)-invariant relative boundary hull of an orbit is the relative boundary of the orbit.

**Proof.** If a \( G \)-invariant relative boundary hull contains a point of the orbit then it must contain the entire orbit, contradicting the definition of a relative boundary hull. \( \square \)

Using the existence of relative boundary hulls from Theorem 1.1 and Proposition 7.1 one can easily reprove the following proposition for \( Y \) and \( G \) both affine over \( \text{Spec} \, \mathbb{Z} \).

**Corollary 7.2** ([Bor91, Proposition in Section I.1.8]). The orbit of an affine algebraic group action is locally closed.

**Proof.** The orbit \( yG = \alpha_y(G) \) is the image of a polynomial map and hence admits a relative boundary hull \( D \). It follows that \( Dg \) is also a relative boundary hull for all \( g \in G \). Hence, \( \bigcap_{g \in G} Dg \) is a \( G \)-invariant relative boundary hull of \( yG \) and the relative boundary of \( yG \) by Proposition 7.1. It follows that \( yG = \overline{yG} \setminus \bigcap_{g \in G} Dg \) is locally closed (written in canonical form). \( \square \)

The proof shows how to avoid the iteration in Algorithm 1:

**Corollary 7.3.** Let \( D \) be a relative boundary hull of \( yG \). Then there exists finitely many group elements \( g_1, \ldots, g_\ell \) such that \( yG = \overline{yG} \setminus (D \cap Dg_1 \cap \ldots \cap Dg_\ell) \). In particular, the iteration in Algorithm 4 can be avoided, i.e., we only need to apply Algorithm 4 once.

**Proof.** The Noetherianity of \( Y \) shows that the intersection defining the relative boundary \( \bigcap_{g \in G} Dg \) in the previous proof is finite and can be computed by choosing random elements \( g_1, \ldots, g_\ell \) until \( \alpha_y^{-1}(D \cap Dg_1 \cap \ldots \cap Dg_\ell) \) is empty. \( \square \)

We consider an instructive example.

**Example 7.4** (Nilpotent orbit of type \( A_1 \)). Consider the special linear group \( G = \text{SL}_2 \) acting via conjugation on the affine space of \( 2 \times 2 \)-matrices \( Y = \text{Mat}^{2 \times 2} \cong \mathbb{A}^4 \). We are interested in the orbit of \( y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \), i.e. in the set all matrices similar to this Jordan block (independent of the characteristic). Hence, consider the orbit morphism

\[
\alpha_y : G \to Y : \begin{bmatrix} g_{1,1} & g_{1,2} \\ g_{2,1} & g_{2,2} \end{bmatrix} \mapsto \begin{bmatrix} g_{2,2} & -g_{1,2} \\ -g_{2,1} & g_{1,1} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g_{1,1} & g_{1,2} \\ g_{2,1} & g_{2,2} \end{bmatrix} = \begin{bmatrix} g_{2,1}g_{2,2} & g_{2,2}^2 \\ -g_{2,1}^2 & -g_{2,1}g_{2,2} \end{bmatrix}.
\]

The graph of the orbit morphism \( \alpha_y \) is given by

\[
I = \left\langle \begin{array}{c} g_{2,1}g_{2,2} - b_{1,1}, \\ g_{2,2}^2 - b_{1,2}, \\ -g_{2,1}^2 - b_{2,1}, \\ -g_{2,1}g_{2,2} - b_{2,2}, \\ g_{1,1}g_{2,2} - g_{1,2}g_{2,1} - 1 \end{array} \right\rangle \subseteq B[g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}]
\]
for \( B = \mathbb{Z}[b_{1,1}, b_{1,2}, b_{2,1}, b_{2,2}] \). This yields the closure of the orbit 
\[
I \cap B = (b_{1,1} + b_{2,2}, b_{1,1}b_{2,2} - b_{1,2}b_{2,1}),
\]
i.e., the matrices with zero trace and zero determinant.

We make the dimension of a generic fiber zero-dimensional by intersecting with \( \{g_{1,1} = 0\} \). This does not change the image closure; \( \langle I, g_{1,1} \rangle \cap B = I \cap B \). The corresponding relative boundary hull \( D_1 = \{b_{1,1} = b_{2,1} = b_{2,2} = 0\} \) is too big, as \( y \) is contained in the orbit \( yG \), but also in \( D_1 \). The action of \( \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \) \( \in G \) yields a second relative boundary hull \( D_2 = D_1g = \{b_{1,1} = b_{1,2} = b_{2,2} = 0\} \). The intersection
\[
D = D_1 \cap D_2 = \{b_{1,1} = b_{1,2} = b_{2,1} = b_{2,2} = 0\}
\]
is the minimal relative boundary hull, i.e., the relative boundary, and the orbit is locally closed:
\[
yG = \{b_{1,1} + b_{2,2} = b_{1,1}b_{2,2} - b_{1,2}b_{2,1} = 0\} \setminus \{b_{1,1} = b_{1,2} = b_{2,1} = b_{2,2} = 0\}.
\]

For injective orbit morphisms, i.e., for principal orbits we can say more.

**Proposition 7.5.** _If the orbit morphism is injective, then Algorithm 3 which passes from \( \Gamma \sim \Gamma_0 \) can be skipped in Algorithm 4 and the latter yields the relative boundary of the orbit. In particular, Algorithm 1 terminates after a single call of Algorithm 4._

**Proof.** Since the orbit morphism is injective, the dimension of any non-empty fiber is zero and Algorithm 3 is not needed, i.e., \( \Gamma_0 = \Gamma \). Now, Algorithm 4 is devoid of any choices and works intrinsically with \( G \)-invariant inputs. Thereby also its output, the relative boundary hull, is \( G \)-equivariant and therefore is the relative boundary of the orbit by Proposition 7.1. \( \square \)

### 7.1. Another heuristic for computing a generic affine subspace.

The algorithmic approach of this paper can be improved for orbits of irreducible groups: Reduce the dimension of the fiber to zero by intersecting with a complement of the embedded tangent space of the stabilizer at the identity element (=complement of the Lie algebra of the stabilizer).

Recall that orbits of groups are homogeneous and smooth, and the non-empty fibers under
\[
\alpha_y : G \to Y : g \mapsto yg
\]
for \( y \in Y \) are isomorphic [Bor91, Proposition in Section I.1.8]. Hence, we compute the unique fiber dimension locally via linear algebra at \( (y, 1_G) \in \Gamma \subseteq Y \times G \), a point in the graph \( \Gamma \) of \( \alpha_y \). Then, again via linear algebra, we compute in the tangent space \( T_{(y,1_G)}\Gamma \) at \( (y, 1_G) \) a complement \( \tilde{L} \) to the tangent space \( T_{(y,1_G)}F \leq T_{(y,1_G)}\Gamma \) of the fiber at \( F := \pi^{-1}(y) \). The tangent space \( T_{(y,1_G)}F \) along the fiber is computed by considering only derivatives in the direction of the group \( G \). For the hyperplane \( L := (y, 1_G) + \tilde{L} \), the set \( \Gamma_0 := \Gamma \cap L \) satisfies conditions (i), (ii), and (iii) from Subsection 4.1. Interpreting \( L \subseteq G \) yields almost

---

\( ^{16} \) \( G \) and \( Y \) are both defined over \( \mathbb{Z} \).
a system of representatives of $G_y \backslash G$ (some classes outside an open dense set are met more or less than once). We formalize this in an algebraic language in Algorithm 7.

**Algorithm 7: IntersectWithGenericAffineSubspace**

**Input:** The graph $\Gamma := \Gamma_{A_y} \subseteq Y \times G$ of the action homomorphism $\alpha_y : G \to Y$, $g \mapsto gy$ of an irreducible algebraic group $G$ on an algebraic variety $Y$.

We consider $G$ resp. $Y$ given by their coordinate rings $k[x_1, \ldots, x_n]/J_G$ resp. $k[b_1, \ldots, b_m]/J_Y$, the graph $\Gamma$ by its vanishing ideal

$$I = \langle f_1, \ldots, f_t \rangle \subseteq k[b_1, \ldots, b_m, x_1, \ldots, x_n]/(J_G + J_Y),$$

and the points $1_G \in G$ resp. $y \in Y$ by coordinates $(\xi_1, \ldots, \xi_n) \in k^n$ resp. $(\beta_1, \ldots, \beta_m) \in k^m$.

**Output:** a closed affine subset $\Gamma_0 \subseteq \Gamma$ given by an ideal $I_0 \subseteq k[b_1, \ldots, b_m, x_1, \ldots, x_n]/(J_G + J_Y)$ such that Item (i), Item (ii), and Item (iii) from Subsection 4.1 are satisfied.

**IntersectWithGenericAffineSubspace** $(f_1, \ldots, f_t, \xi_1, \ldots, \xi_n, \beta_1, \ldots, \beta_m)$

1. $J^1_{x: \xi, \beta} := \left( \frac{\partial (f_i)_{y=b_j}}{\partial x_i} \right)_{x_i=\xi_i} \in k^{t \times n}$ // Jacobian in fiber direction at $1_G$
2. $\gamma := \text{REF} \left( J^1_{x: \xi, \beta} \right)$ // compute the row echelon form
3. Define $N \subseteq \{1, \ldots, n\}$ as the set of column positions of $\gamma$ without pivots
4. $E := \langle x_i - \xi_i \mid i \in N \rangle$ // define the subspace $L \subseteq \mathbb{A}^n_B$
5. $I_0 := I + E$ // intersect $\Gamma \cap L = \Gamma_0 = V(I_0)$
6. Return $I_0$

Summing up:

**Proposition 7.6.** When applying Algorithm 4 to compute the locally closed projection of $\Gamma := \Gamma_{A_y}$ we can replace Algorithm 3 by Algorithm 7 which only needs derivatives\(^{17}\) and the Gaussian algorithm over a field. In particular, the elimination and primary decomposition needed in Algorithm 3 to compute $\pi(\Gamma)$ can be avoided and the orbit closure $\overline{Gy}$ can be computed as the image closure of $\pi(\Gamma_0) = \overline{\pi(\Gamma)}$ with an elimination involving the usually smaller $\Gamma_0 \subseteq \Gamma$.

**Remark 7.7.** In principle, this approach of reducing the fiber dimension to zero works for a general $\Gamma$ by considering the tangent space along the fibers of a (smooth) point in $\Gamma$. However, we face several difficulties not existing in the case of group orbits. First, unlike the case of group orbits, there is no guarantee that the tangent space along the fiber at the chosen smooth point is of generic fiber dimension, hence one a priori needs to compute the generic fiber dimension, usually via elimination. Second, in general, we usually do not have a smooth point at hand and even though constructing such an ideal is possible, it is non-trivial and necessary in each step; in case of group orbits we can simply take $(y, 1_G)$ and we are done in one step. Third, the $\xi_i$'s defining the linear subspace $E = \langle x_i - \xi_i \mid i \in N \rangle$ tend to be more complicated than in the group orbit case (there the $\xi_i$'s, representing the identity $1_G$ are usually zero or one). This problem gets more pronounced in later steps of the algorithm.

\(^{17}\)Computing $J^1_{x: \xi, \beta} := \left( \frac{\partial (f_i)_{y=b_j}}{\partial x_i} \right)_{x_i=\xi_i}$ is more efficient than computing $J^1_{x: \xi, \beta} := \left( \frac{\partial f_i}{\partial y} \right)_{x_i=\xi_i, b_j=\beta_j}$.
Example 7.8 (Nilpotent cone in type $A_1$, cf. Example 7.4, revised). Consider again the special linear group $G = SL_2$ acting via conjugation on the affine space of $2 \times 2$-matrices $Y = \text{Mat}_2^{2 \times 2} \cong \mathbb{A}^4_2$ and the orbit of $y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. The graph of the action morphism $\alpha_y$ is given by

$$I = \begin{bmatrix} g_{2,1}g_{2,2} - b_{1,1}, & g_{2,2}^2 - b_{1,2}, \\ -g_{2,1}^2 - b_{2,1}, & -g_{2,1}g_{2,2} - b_{2,2}, \\ g_{1,1}g_{2,2} - g_{1,2}g_{2,1} - 1 \end{bmatrix} \subseteq B[g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}]$$

and we know that $(y, 1_G) = (\beta, \xi) = ((0, 1, 0, 0); (1, 0, 0, 1)) \in \Gamma = V(I)$. The Jacobian

$$J^f_{x;\xi,\beta} \coloneqq \left. \frac{\partial(f_{ab})|_{\beta,\xi}}{\partial x_i} \right|_{x_i = \xi_i} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

suggests $L = \{g_{1,2} = 0\}$. This yields a cheaper way to compute the closure of the orbit

$$I \cap B = \{I, g_{1,2}\} \cap B = \langle b_{1,1} + b_{2,2}, b_{1,1}b_{2,2} - b_{1,2}b_{2,1} \rangle$$

along with a relative boundary hull.

7.2. Epi-mono decomposition. Following Remark 4.12 an ideal approach for computing the orbit would be to use the epi-mono decomposition $G \twoheadrightarrow G_y \backslash G \rightarrow Y$ of $\alpha_y$, where $G_y$ is the stabilizer of $y \in Y$. Note that $G_y \backslash G$ is in general quasi-projective and not affine. Then $\gamma G = \text{im } \alpha_y = \text{im } \iota_y$ and the monic $\iota_y$ has trivially zero dimensional fibers. This approach has several advantages:

- The description of $G_y \backslash G$ as a preparatory step is independent from the space $Y$ which in applications tends to be of much larger dimension than the group $G$.
- The fibers are singletons, so the initial step replacing $\Gamma \twoheadrightarrow \Gamma_\iota$ is obsolete. This removes the arbitrariness in choosing hyperplanes in our algorithm.
- Since $\iota_y$ is $G$-equivariant the relative boundary hull will automatically be $G$-equivariant and hence will coincide with the relative boundary (and we are done without subsequent “invariantization”).

We will pursue the algebraic compilation of this approach in future work. The challenge will be to compute the monic $\iota_y$ given $\alpha_y$ without the explicit pre-computation of the rational invariants of the action of $G_y$ on $G$ by multiplication from the left.

7.3. Finitely many orbits. Examples 7.4 and 7.8 considered a single orbit. Its closure contained another single orbit (consisting of the zero matrix). Of course, in general, a group operation can have infinitely many orbits\(^{18}\) and even the closure of an orbit can contain infinitely many orbits\(^{19}\). Below we mention some algorithmic benefits applicable when a $G$-space partitions into finitely many orbits.

Assume the action admits finitely many orbits and let $\{y_1, \ldots, y_e\}$ be a set of representatives of $G$-orbits. Then we can compute the closure $y_iG$, defined by an ideal $J_i$, of any orbit

\(^{18}\)e.g., the action of $\text{SL}_2$ via conjugation on $2 \times 2$-matrices over an infinite field.

\(^{19}\)e.g., the right action of $\text{GL}_2$ on $2 \times 2$-matrices over an infinite field yields infinitely many column reduced echelon forms as representatives of orbits.
AN ALGORITHMIC APPROACH TO CHEVALLEY’S THEOREM

via elimination. Note that Proposition 7.6 is applicable for each of these eliminations, with stronger gains in efficiency for smaller orbits. Determining the containment of the closures of the orbits in one another is an ideal membership test: $y_iG \supseteq y_jG$ iff $J_i \subseteq J_j$. Hence, the description of any such orbit $y_iG$ as a locally closed set can be given by the difference

$$y_iG = y_iG \setminus \bigcup_{y_jG \supseteq y_iG} y_jG$$

of its closure $y_iG$ and the closures of all (maximal) orbits contained in it. The containment of the closures induces defines the finite stratification of the $G$-space by its orbits. The nilpotent cone of a semisimple algebraic group $G$ is such space.

Another important class of group operations on varieties with finitely many orbits\(^{20}\) are the torus operations on normal affine toric varieties (cf. [CLS11]). The combinatorial description of toric varieties via the orbit-cone-correspondence is enough to classify orbits. It also provides a distinguished representative of each orbit. Furthermore, for the non-maximal orbits there is an explicit description of an epi-mono decomposition of the orbit morphism (cf. Subsection 7.2). For the convenience of the reader, we have summarized the relevant results of toric varieties in Appendix D.

**Example 7.9.** We consider the non-smooth cone $\sigma = \text{Cone}(e_1, e_2, e_1 + e_3, e_2 + e_3) \subset N_{\mathbb{R}} = \mathbb{R}^3$ with standard lattice $N = \mathbb{Z}e_1 \oplus \mathbb{Z}e_2 \oplus \mathbb{Z}e_3$. The dual cone $\sigma^\vee$ is generated by the Hilbert basis $\mathcal{H} = \{e_1, e_2, e_3, e_1 + e_2 - e_3\}$, which defines the orbit morphism

$$\alpha_{(1,1,1,1)}^{\mathcal{H}} : \begin{cases} \text{Spec } \mathbb{Z}[t_1, t_2, t_3, s]/(t_1t_2t_3s - 1) \to \text{Spec } \mathbb{Z}[b_1, b_2, b_3, b_4], \\ (t_1, t_2, t_3) \mapsto (b_1, b_2, b_3, b_4) := (t_1, t_2, t_3, t_1t_2t_3^{-1} := t_1^2t_2^2s). \end{cases}$$

Since the orbit morphism $\alpha_{(1,1,1,1)}^{\mathcal{H}}$ is injective, our algorithm computes the dense torus orbit $O(\{0\})$ in one step (cf. Proposition 7.5)

$$O(\{0\}) = V(b_1b_2 - b_3b_4) \setminus (V(b_1, b_4) \cup V(b_2, b_4) \cup V(b_1, b_3) \cup V(b_2, b_3)),$$

which we expect from the theory. Indeed, the orbit-cone-correspondence yields for each of the four rays $\rho$ of $\sigma$ the distinguished point $p_\rho$ on the corresponding orbit $O(\rho)$ and the equations of the closure of the orbit.

| ray generator $u_\rho$ | $p_\rho = (b_1, b_2, b_3, b_4) \in \{0, 1\}^4 \cap O(\rho)$ | orbit closure $\overline{O(\rho)}$ |
|------------------------|---------------------------------|----------------------|
| $e_1$ | $(0, 1, 1, 0)$ | $V(b_1, b_4)$ |
| $e_2$ | $(1, 0, 1, 0)$ | $V(b_2, b_4)$ |
| $e_1 + e_3$ | $(0, 1, 0, 1)$ | $V(b_1, b_3)$ |
| $e_2 + e_3$ | $(1, 0, 0, 1)$ | $V(b_2, b_3)$ |

The 3-dimensional dense orbit $O(\{0\})$ and the four 2-dimensional orbits are only two layers of the stratification of the toric variety $\overline{O(\{0\})}$ into orbits. The facets defined by two neighboring rays induce four 1-dimensional orbits and the cone $\sigma$ induces a singleton orbit.

The action of the torus on the remaining orbits is given by the orbit morphisms

$$\alpha_{(a_1, a_2, a_3, a_4)}^{\mathcal{H}} : \begin{cases} \text{Spec } \mathbb{Z}[t_1, t_2, t_3, s]/(t_1t_2t_3s - 1) \to \text{Spec } \mathbb{Z}[b_1, b_2, b_3, b_4], \\ (t_1, t_2, t_3) \mapsto (b_1, b_2, b_3, b_4) := (a_1t_1, a_2t_2, a_3t_3, a_4t_4t_3^{-1}). \end{cases}$$

\(^{20}\)The only connected algebraic groups $G$ guaranteed to always produce finitely many orbits for any operation with a dense orbit are either a torus or a product of a torus and $G_a$, cf. [Pop17, Theorem 2].
The kernel of these actions/maps can be determined combinatorial, e.g., the kernel of $\omega^{\#}_{(0,1,1,0)}$ is generated by $t_1$. This yields an epi-mono decomposition of $\omega^{\#}_{(0,1,1,0)}$ and the two-dimensional quotient torus acts via the monic $t_{(0,1,1,0)} : (t_2, t_3) \mapsto (0, t_2, t_3, 0)$.

**ACKNOWLEDGMENTS**

We would like to thank Florian Heiderich for several discussions on algebraic groups and Sebastian Posur for a careful read and many improvement suggestions. Our thanks also goes to Sebastian Gutsche and Tom Kuhmichel who contributed to the infrastructure of the software we use. Last but not least we thank the anonymous referees for their helpful comments and suggestions that contributed to the improvement of this paper.

**APPENDIX A. GRÖBNER BASIS PROOF FOR THE DIMENSION REDUCTION**

We give a proof, based on Gröbner basis theory, which constructs the affine subspace $L$ necessary to reduce the dimension of fibers (cf. Subsection 4.1 and the next proposition). We assume the coefficients ring $k$ to have effective coset representatives (cf. Subsection 3.2). As mentioned above, we can always assume $k$ to be reduced. Furthermore, we can also always assume $k$ to be a domain, otherwise it is a product of finitely many domains $k = k_1 \times \cdots \times k_r$ and we can split the projection into $r$ different ones $\text{Spec} k_1 \times \text{Spec} k \to \text{Spec}(k_1 \otimes_k B)$.

**Proposition A.1.** Let $k$ be an infinite domain, $B = k[b_1, \ldots, b_n]$, and $\Gamma \subseteq k_B^n$ be a nonempty irreducible affine subset of generic fiber dimension $d$ along $\pi^{\#}$. Let $I = \mathcal{I}(\Gamma) \triangleleft B[x_1, \ldots, x_n]$ be the vanishing ideal of $\Gamma = V(I) \subseteq k_B^n$. If $k \cap I = \langle 0 \rangle$, then for each $d' \leq d$ there exists an affine subspace $L \subseteq k_B^n$ of dimension $\dim_B L = n - d'$ such that $\Gamma_0 := L \cap \Gamma$ satisfies (cf. Subsection 4.1)

- the fibers of $\Gamma_0$ along the projection $\pi^{\#}_{\Gamma_0}$ are generically $(d - d')$-dimensional;
- $\overline{\pi(\Gamma)} = \overline{\pi(\Gamma_0)}$.

Moreover, $L$ can be chosen constant along the fibers, i.e., there exists an $(n - d')$-dimensional affine subspace $L' \subseteq k_B^n$ such that $L := L' \times_k \text{Spec} B \subseteq k_B^n = k_B^n \times_k \text{Spec} B$. Finally, the subspace $L'$ can be generically chosen.

**Proof.** Define $K := \text{Frac}(B/(I \cap B))$ to be the field of fractions of the domain $B/(I \cap B)$. Due to our assumption it is a field extension of $k$. Since $k$ has effective coset representatives then so does $B$. Hence, one can decide equality of elements in $B/(I \cap B)$ and $K$ is a constructive field, in particular, again with decidable equality of elements.

Consider the extended ideal $I^e = \langle I \rangle \triangleleft K[x_1, \ldots, x_n]$. After possible renaming of the indeterminates we can assume that $x_1, \ldots, x_{d'}$ is an independent set modulo $I^e$. Compute a comprehensive Gröbner basis $G$ of $I^e \triangleleft K[x_1, \ldots, x_{d'}][x_{d'+1}, \ldots, x_n]$ and set $q := \prod_{p \in G} \text{LC}(p) \in K[x_1, \ldots, x_{d'}] \setminus \{0\}$. Now choose a specialization $\sigma : K[x_1, \ldots, x_{d'}] \to K, x_i \mapsto a_i$ such that $a_i \in k \subseteq K$ and $\sigma(q) = q(a_1, \ldots, a_{d'}) \neq 0$. Since $G$ is a comprehensive Gröbner basis it follows that the $\sigma$-specialization $G_\sigma \subseteq K[x_{d'+1}, \ldots, x_n]$ is a Gröbner basis of the $\sigma$-specialization $J$ of $I^e$ to $K[x_{d'+1}, \ldots, x_n]$. Since $q(a_1, \ldots, a_{d'}) \neq 0$ the Gröbner basis $G_\sigma$ has no constant polynomials, in particular, the ideal $J$ is not the unit ideal in $K[x_{d'+1}, \ldots, x_n]$ and $\dim J = d - d'$ due to the independence of $x_1, \ldots, x_{d'}$. Without loss of generality we can assume all leading coefficients of $G_\sigma$ to be 1. The product $q'$ of the denominators of the remaining coefficients is an element of $B \setminus I$. It follows that $G_\sigma$ is

\footnote{Note that $d' = \dim I^e$.}
a Gröbner basis with normalized leading coefficients over the localization \( (B/(I \cap B))_{q'} \).

Now we are done with \( L' := V((x_1 - a_1, \ldots, x_d - a_d)) \subseteq \mathbb{A}_k^d \), since a dense subset of \( \pi(\Gamma) \) has a preimage in \( \Gamma \cap L' \) under \( \pi \). □

Otherwise, if \( k \cap I \neq \langle 0 \rangle \) then we replace \( k \) by \( k/(k \cap I) \), \( B \) by \( B/(k \cap I)_B \), and \( R \) by \( R/(k \cap I)_R \). By the discussion at the beginning of this Section we can again assume \( k \) to be a domain. If \( k \) is finite, it is a finite field and we consider a finite field extension \( \bar{k}/k \) over which the linear space constructed in the proof of Proposition A.1 over the algebraic closure \( \bar{k} \) can be realized (see Remark 4.7). Geometrically this means that we are considering the image of \( \bar{\Gamma} := \text{Spec} \bar{k} \times_{\text{Spec} k} \Gamma \) under the composed projection \( \bar{\pi} : \mathbb{A}_B^n \to \text{Spec} \bar{B} \to \text{Spec} B \), where \( \bar{B} := \bar{k} \otimes_k B \). Note that \( \bar{\pi}(\bar{\Gamma}) = \pi(\Gamma) \) and that \( \text{Spec} \bar{B} \subset \text{Spec} B[t] \) is itself affine over \( \text{Spec} B \) with zero dimensional fibers. This finishes the proof of generic existence of the hyperplanes needed in Algorithm 3 and its variant (Algorithm 8) below.

**Appendix B. Get rid of the primary decomposition**

To bring the dimension of the fibers to zero, in particular to apply Appendix A correctly, we need irreducible varieties. Subsection 4.1 applies the primary decomposition to ensure irreducibility, but only after trying some heuristics. In this section, we present a stronger heuristic approach that makes the primary decompositions redundant. Despite theoretical interest, this approach does not seem faster than computing a primary decomposition of the smaller \( \Gamma_0 \) (cf. line 15 of Algorithm 3; note that computing a decomposition of the entire \( \Gamma \) is much more expensive).
The stronger heuristic uses a hyperplane, which reduces the dimension of the image to construct a split in total space, as in Example 4.5.

**Algorithm 8: ZeroDimensionalFibersNoDecomposition**

**Input:** A closed subset $\Gamma \subseteq \mathbb{A}_B^n \xrightarrow{\pi} \text{Spec } B$. We assume $k$ to be an infinite domain and that the composition $\Gamma \to \text{Spec } B \to \text{Spec } k$ is dominant.

**Output:** A closed affine subset $\Gamma'_0 \subseteq \Gamma$ and a (possibly empty) list of additional closed affine subsets $[\Gamma'_1, \ldots, \Gamma'_e]$ defining a closed affine subset $\Gamma_0 := \Gamma'_0 \cup \Gamma'_1 \cup \ldots \cup \Gamma'_e \subseteq \Gamma$, such that (iii) and (ii) are satisfied; more specifically, the fibers in (ii) are locally zero-dimensional over the subset $\Gamma'_0$ of $\Gamma_0$.

**ZeroDimensionalFibersNoDecomposition** ($I$)

```plaintext
1 $\Gamma'_0 := \Gamma$ // candidate $\Gamma_0$, dimension to be decreased below
2 $s := 1$ // step counter
3 $\ell := []$ // list of additional components

/* decrease the dimension in the fiber, till it is zero */
4 while $\dim(\Gamma'_0) - \dim(\pi(\Gamma'_0)) > 0$ do
5     $E := \text{RandomHyperplane}(\mathbb{A}_k^n) \times \text{Spec } k \to \text{Spec } B$ // cf. Remark 4.7
6         // intersect with hyperplane
7     if $\dim(\Gamma''_0) < \dim(\Gamma'_0)$ then // intersection decreases dimension...
8         if $\pi(\Gamma'_0) \subseteq \pi(\Gamma''_0)$ then // ...without reducing the image
9             $\Gamma'_0 := \Gamma''_0$ // found smaller $\Gamma'_0$
10         else $s > n$ then // split w.r.t. reduced image, avoid early
11             /* try splitting $\Gamma'_0$ by splitting the base space */
12             $\Delta := \pi(\Gamma'_0) \setminus \pi(\Gamma''_0)$ // “complement” of the reduced image
13             if $\pi(\Gamma'_0) \not\subseteq \Delta$ then // if the image reduces nontrivially
14                 $\Gamma'_0 := \pi^{-1}\left(\pi(\Gamma'_0) \cap \Gamma\right)$ // continue with one part
15                 $\ell := \text{Add}(\ell, \pi^{-1}(\Delta) \cap \Gamma)$ // store the other half
16             else $s > 4n$ then // very expensive, avoid early
17                 $\Gamma'_1 := \pi^{-1}\left(\pi(\Gamma'_0) \cap \Gamma\right)$ // preimage of $\pi(\Gamma'_0)$
18                 if $\Gamma \not\subseteq \Gamma'_1$ then // if this preimage is not the entire $\Gamma$
19                     $\Gamma'_2 := \Gamma \setminus \Gamma'_1$ // “complement” of the preimage
20                     if $\Gamma \not\subseteq \Gamma'_2$ then
21                         $\Gamma'_0 := \Gamma'_1$ // continue with one part
22                         $\ell := \text{Add}(\ell, \Gamma'_1)$ // store the other half
23         $s := s + 1$ // increase number of steps
24 return $[\Gamma'_0, \ell]$
```

Generically, we expect the following behavior in Algorithm 8: Intersecting with a generic hyperplane decreases the dimension of each component of $\Gamma$. The decrease in dimension generically happens without decreasing the image, as long as the dimension of the fiber is positive over a dense set of the image. Once we can no longer reduce the dimension of the
fiber, the image of certain components generically needs to shrink. In this case, \( \Gamma \) generically splits into two components, similar to Example 4.5 (or, more rarely, but computationally less expensive, similar to Example 4.4).

**APPENDIX C. IMPROVING THE ITERATION IN ALGORITHM 1**

As mentioned in Remark 3.4 it is often advantageous to exploit known decompositions of the relative boundary hulls or even to compute decompositions in irreducible components. As a consequence, Algorithm 15 below will then follow a directed graph structure rather than the linear structure in Algorithm 1. To achieve this flow in Algorithm 15 we loop over a bookkeeping bipartite directed graph data structure which we will now define. The poset \((Z, \succeq)\) in the following definition will specialize to the poset of closed reduced subsets of \(\text{Spec } B\) in Algorithm 1.

**Definition C.1.** A (bookkeeping) bipartite directed graph data structure \( c \) consists of

- a mutable finite set \( \mathcal{A}(c) \) of positive nodes,
- a mutable finite set \( \mathcal{D}(c) \) of negative nodes, and
- a finite FIFO \( \mathcal{P}(c) \) for so-called pre-nodes (with push and pop operations).

where:

- A **positive node** \( A = (\tilde{A}, \text{parents}(A), \text{children}(A)) \in \mathcal{A}(c) \) consists of an underlying object \( \tilde{A} \in Z \) together with two mutable subsets
  \[ \text{parents}(A), \text{children}(A) \subseteq \mathcal{D}(c) \]
  satisfying:
  \[ \tilde{D} \succeq \tilde{A} \succeq \tilde{D}' \quad \forall D \in \text{parents}(A), D' \in \text{children}(A). \]
  Furthermore, require that the map \( \tilde{\cdot} : \mathcal{A}(c) \to Z, A \mapsto \tilde{A} \) is injective.

- A **negative node** \( D = (\tilde{D}, \text{parents}(D), \text{children}(D)) \in \mathcal{D}(c) \) consists of an underlying object \( \tilde{D} \in Z \) together with two mutable subsets
  \[ \text{parents}(D), \text{children}(D) \subseteq \mathcal{A}(c) \]
  satisfying:
  \[ \tilde{A} \succeq \tilde{D} \succeq \tilde{A}' \quad \forall A \in \text{parents}(D), A' \in \text{children}(D). \]
  Furthermore, require that the map \( \tilde{\cdot} : \mathcal{D}(c) \to Z, D \mapsto \tilde{D} \) is injective.

- A **pre-node** in \( \mathcal{P}(c) \) is a triple \( (D, \ell, \Gamma) \), where \( D \in \mathcal{D}(c) \), \( \ell \) a nonnegative integer (called level), and \( \Gamma \) an arbitrary object with no further specification.

- Let \( A \in \mathcal{A}(c) \) be a positive node and \( D \in \mathcal{D}(c) \) a negative node. Then \( D \in \text{parents}(A) \) iff \( A \in \text{children}(D) \) and \( A \in \text{parents}(D) \) iff \( D \in \text{children}(A) \).

- For each negative node \( D \in \mathcal{D}(c) \) we have \( \text{parents}(D) \neq \emptyset \).

The bipartite directed graph structure is defined by the parents-children relationship.

We wrote several algorithms to manipulate \( c \). The following algorithm will be used in the main Algorithm 15 to attach the locally closed output \( \tilde{A} \setminus (\tilde{D}_1 \cup \cdots \cup \tilde{D}_a) \) of Locally-ClosedApproximationOfProjection(\( \Gamma \cap \pi^{-1}(\tilde{D}) \)) (Algorithm 4) to a pre-existing negative
node $D \in \mathcal{D}(c)$. The closure $\widetilde{A}$ of the projection will be attached as a positive node (with underlying object $\widetilde{A}$) and the $\widetilde{D}_i$'s as negative nodes (with underlying objects $\widetilde{D}_i$).

**Algorithm 9:** Attach positive node & corresponding negative nodes to negative node

**Input:** A bipartite directed graph data structure $c$, a negative node $D \in \mathcal{D}(c)$, an integer $\ell \geq 0$, an object $\widetilde{A} \in \mathbb{Z}$, objects $\{\widetilde{D}_1, \ldots, \widetilde{D}_a\} \subset \mathbb{Z}$, and an object $\Gamma$

**Output:** Nothing, a side effect on $c$

1. if $\exists B \in \mathcal{A}(c) : \widetilde{A} = \widetilde{B}$ then
   2. $A := B$
   else
   3. Add $A := (\widetilde{A}, \emptyset, \emptyset)$ to $\mathcal{A}(c)$
   4. Add $D$ to parents($A$)
   5. for $i = 1, \ldots, \ell$ do
      6. if $\exists D \in \mathcal{D}(c) : \widetilde{D}_i = \widetilde{D}$ then
         7. $D_i := D$
         8. Create a pre-node $(D_i, \ell + 1, \Gamma)$ and push it to $\mathcal{P}(c)$
      else
         9. Add $D_i := (\widetilde{D}_i, \emptyset, \emptyset)$ to $\mathcal{D}(c)$
         10. Add $A$ to parents($D_i$)
         11. Add $D_i$ to children($A$)

The following algorithm checks whether the FIFO of pre-nodes of the bipartite directed graph data structure $c$ has been exhausted. It will be used as the while-loop condition in Algorithm 15:

**Algorithm 10:** Check if the FIFO of pre-nodes of $c$ is exhausted

**Input:** A bipartite directed graph data structure $c$

**Output:** true or false

1. return $\text{IsEmpty}(\mathcal{P}(c))$

The next algorithm is used in Algorithm 15 to extract the oldest pre-node in the bipartite directed graph data structure $c$.

**Algorithm 11:** Get the oldest pre-node of $c$

**Input:** A bipartite directed graph data structure $c$

**Output:** A decorated pre-node $(D_i, \ell, \Gamma)$

1. return $\text{Pop}(\mathcal{P}(c))$, i.e., return the oldest element in $\mathcal{P}(c)$ and delete it from $\mathcal{P}(c)$

Algorithm 15 will use the following procedure to check if all pre-nodes in $c$ of a certain level have been processed in order to trigger the squashing of $c$. 
Algorithm 12: The minimum level among the pre-nodes of $c$

\textbf{Input:} A bipartite directed graph data structure $c$
\textbf{Output:} A non-negative integer

**MinimalLevelOfPreNodes** ($c$)

1. \textbf{return} $\min \{ \ell \mid (D, \ell, \Gamma) \in \mathcal{P}(c) \}$

After all pre-nodes of a certain level have been processed the bipartite directed graph data structure $c$ can be squashed as follows:

- remove all pairs $(D, A) \in D(c) \times A(c)$ from $c$ once $D$ is the only parent of $A$ and $A$ is the only child of $D$ and $A = D$;
- reassign the parents-children relationship in $c$;
- for all positive nodes $A \in A(c)$ remove all obsolete children $D \in \text{children}(A)$, i.e., remove all those children of $A$ which are smaller than other children of $A$ with respect to $\succeq$.

Algorithm 13: Squash the bipartite directed graph data structure $c$

\textbf{Input:} A bipartite directed graph data structure $c$
\textbf{Output:} Nothing, a side effect on the bipartite directed graph data structure $c$

**Squash** ($c$)

1. for $A \in A(c)$ do
2. \hspace{1em} if parents($A$) = $\{D\}$ and children($D$) = $\{A\}$ and $\tilde{A} = \tilde{D}$ then
3. \hspace{2em} for $A' \in \text{parents}(D)$ do
4. \hspace{3em} Add the elements of children($A$) to children($A'$)
5. \hspace{2em} for $D' \in \text{children}(A)$ do
6. \hspace{3em} Add the elements of parents($D$) to parents($D'$)
7. \hspace{2em} remove $A, D$ from $c$
8. for $A \in A(c)$ do
9. \hspace{1em} for $D \in \text{children}(A)$ do
10. \hspace{2em} if $\exists D' \in \text{children}(A) \setminus \{D\}$ with $\tilde{D'} \succeq \tilde{D}$ then
11. \hspace{3em} remove $D$ from children($A$)
12. \hspace{3em} remove $A$ from parents($D$)
13. \hspace{2em} if parents($D$) = $\emptyset$ then
14. \hspace{3em} remove $D$ from $D(c)$

\textsuperscript{a}i.e., remove $A$ from $A(c)$ and from parents($D'$) and children($D'$) of all $D' \in D(c)$ and remove $D$ from $D(c)$ and from parents($A'$) and children($A'$) of all $A' \in A(c)$.

The squash operation can be repeated until $A(c)$ and $D(c)$ do not decrease any further.

We graphically demonstrate the first loop of squash, which removed a negative (light) red node and its child, a positive (light) green node. Parents are drawn left of their children.
This diagram shows how the first loop of Squash brings more negative nodes to a single layer, such that the second loop can be more productive in removing negative nodes. Typically, the negative nodes originally appearing in an earlier generation are $\succeq$-bigger than those appearing in later generations. Hence, some of these $\succeq$-smaller nodes can be removed by the second loop in Squash, as the next diagram shows.

The following last helper algorithm is used by Algorithm 15 to convert the bipartite directed graph data structure $c$ into the corresponding constructible set $C$, as a disjoint union of the multiple differences $L = \overline{A} \setminus \overline{D}_1 \setminus \ldots \setminus \overline{D}_a = \overline{A} \setminus (\overline{D}_1 \cup \ldots \cup \overline{D}_a)$. Since we do not assume $(Z, \succeq)$ to be a Boolean algebra we understand multiple differences and their unions as formal expressions. In our application they evaluate to elements in the Boolean algebra of constructible sets.

Algorithm 14: The constructible set defined by $c$

Input: A bipartite directed graph data structure $c$

Output: A constructible set $C$ as disjoint union of multiple differences

AsUnionOfMultipleDifferences ($c$)

1. $C := \emptyset$
2. for $A \in \mathcal{A}(c)$ do
3.     $L := \overline{A}$
4.     for $D \in \text{children}(A)$ do
5.         $L := L \setminus \overline{D}$
6.     $C := C \uplus L$

return $C$

// $L = \overline{A} \setminus \overline{D}_1 \setminus \ldots \setminus \overline{D}_a$

We can now formulate the promised substitute of Algorithm 1:
Algorithm 15: Compute constructible projection (geometric)

Input: A closed subset $\Gamma \subseteq \mathbb{A}^n_B \xrightarrow{\pi} \text{Spec } B$

Output: $\pi(\Gamma) = C$ as a finite disjoint union of locally closed subsets of $\text{Spec } B$

ConstructibleProjection($\Gamma$)

1. Initialize a bipartite directed graph data structure $c$ with a single pre-node $(\text{Spec } B, 0, \Gamma)$ for the background poset $(\mathbb{Z}, \succeq)$ being the poset of closed reduced subsets of $\text{Spec } B$

2. while not IsDone($c$) do // as long as there are pre-nodes to process
   3. $(D, \ell, \Gamma) := \text{Pop}(c)$ // get the oldest pre-node
   4. $\Gamma := \Gamma \cap \pi^{-1}(\tilde{D})$ // compute the preimage of $\tilde{D}$ in $\Gamma$
   5. $\tilde{A}, \{\tilde{D}_1, \ldots, \tilde{D}_a\} := \text{LocallyClosedApproximationOfProjection}(\Gamma)$
   6. if $\tilde{A} \neq \emptyset$ then
      7. $\text{Attach}(c, D, \ell, \tilde{A}, \{\tilde{D}_1, \ldots, \tilde{D}_a\}, \Gamma)$
   8. if MinimalLevelOfPreNodes($c$) > $\ell$ then // once level $\ell$ is exhausted
      9. $\text{Squash}(c)$

10. return AsUnionOfMultipleDifferences($c$)

Note that $\text{LocallyClosedApproximationOfProjection}$ might additionally return multiple components $[\Gamma'_1, \ldots, \Gamma'_e]$ (cf. Remark 4.11), which we then add immediately after line 5 at the beginning of $P(c)$ as new pre-nodes $(D, \ell, \Gamma_i)$.

Remark C.2. We now compare our use of the bipartite directed graph data structure $c$ with the constructible tree used in [HMeS19]. Whereas they clean up the constructible tree “at the end of the loop” to get rid of obsolete nodes, we squash the data structure after each level has been completed in order to avoid computing at least some of the obsolete nodes. Otherwise, the first and second loop in the squash Algorithm 13 correspond to the first and second cleaning operation in [HMeS19, Section 3.1.1], respectively. A further simplification is that we do not need to delete negative nodes “and all of its descendants” since the negative nodes we delete in the second loop will not have any descendants when the level in which they are computed is completed.

Remark C.3. Algorithm 15 can be parallelized within a single level $\ell$. The procedure $\text{Attach}$ is a blocking operation and the procedure $\text{Squash}$ should be performed in single-threaded mode. In particular, the expensive procedure $\text{LocallyClosedApproximationOfProjection}(\Gamma)$ can be called for multiple $\Gamma$’s within a fixed level $\ell$.

Appendix D. A Primer on Toric Varieties

Let $N \cong \mathbb{Z}^n$ be a lattice and $T_N := N \otimes \mathbb{Z} \times k^* \cong T_k^n$ the corresponding torus over the algebraically closed field $k$ of characteristic 0 (with $N$ being the lattice of one-parameter subgroups of $T_N$). Further let $\sigma \subseteq N_k$ be a strongly convex rational polyhedral cone, $\sigma^\vee \subseteq M_k$ its dual cone, where $M := N^\vee := \text{Hom}_{\mathbb{Z}}(N, \mathbb{Z})$ is the character lattice of $T_N$. A Hilbert basis $\mathcal{H} := \{m_1, \ldots, m_s\} \subseteq M$ of the associated saturated affine semigroup
The normal affine toric variety $U^\sigma := \sigma^\vee \cap M$ describes the action morphism

$$\alpha^{\not\sigma} : \begin{cases} \mathbb{A}^n_k \times T_N & \hookrightarrow \mathbb{A}^n_k, \\ (\underline{a}, \underline{t}) = (a_1, \ldots, a_s, t_1, \ldots, t_n) & \mapsto (b_1, \ldots, b_s) := (a_1 t^{m_1}, \ldots, a_s t^{m_s}). \end{cases}$$

of the torus $T_N$ on $\mathbb{A}^n_k$. The principal orbit $O(\{0\}) := p_0 \cdot T_N$ of the distinguished point $p_0 := (1, \ldots, 1) \in \mathbb{A}^n_k$ under the torus action can be described by the image of

$$\alpha^{\not\sigma}_{(1, \ldots, 1)} : \begin{cases} \Spec k[\underline{t}^\pm_1, \ldots, \underline{t}^\pm_n] := T^\bullet \cong T_N & \hookrightarrow \mathbb{A}^n_k := \Spec k[b_1, \ldots, b_s], \\ \underline{t} = (t_1, \ldots, t_n) & \mapsto (b_1, \ldots, b_s) := (\underline{t}^{m_1}, \ldots, \underline{t}^{m_s}). \end{cases}$$

The normal affine toric variety $U^\sigma \cong \Spec k[S^\sigma]$ is defined as the orbit closure $V(\{0\}) = O(\{0\})$ in $\mathbb{A}^n_k$. The principal orbit $O(\{0\})$ corresponds to the unique 0-dimensional facet $\{0\}$ of the strongly convex $\sigma$. More generally, the orbit-cone-correspondence states that each $d$-dimensional facet $\tau \prec \sigma$ corresponds to a distinguished point $p_\tau$ and its $(n - d)$-dimensional orbit $O(\tau) := p_\tau \cdot T_N$ with orbit closure $V(\tau) := O(\tau)$ in $\mathbb{A}^n_k$. The coordinates of $p_\tau$ are either 0 or 1. The $i$-th coordinate of $p_\tau$ is 1 iff $\tau \prec N_\mathbb{R}$ is perpendicular to $m_i \in M_\mathbb{R}$. The orbit $O(\tau)$ is a principal homogeneous space for the factor group $T_{N(\tau)} := N(\tau) \otimes \mathbb{Z}^* k^*$ of the full torus $T_N$, for the lattice $N(\tau) := N(\tau \cap N)$. In particular, each of these orbits admits an epi-mono factorization of the orbit morphism via combinatorial means.

Furthermore, $V(\tau) = \bigcup_{\nu \succeq \tau} O(\nu)$, inducing a stratification of the orbits. In particular, the locally closed principal torus orbit $O(\{0\})$ admits the description

$$O(\{0\}) = V(\{0\}) \setminus \bigcup_{\rho \in \text{Rays}(\sigma)} V(\rho).$$

In other words, the relative boundary of the locally closed orbit is $\partial O(\{0\}) = \bigcup_{\rho \in \text{Rays}(\sigma)} V(\rho)$.

\section*{References}

[AL94] William W. Adams and Philippe Loustaunau, \textit{An Introduction to Gröbner Bases}, Graduate Studies in Mathematics, American Mathematical Society, 1994. MR 1287608 (95g:13025)

[Bäc14] T. Bächler, \textit{Counting solutions of algebraic systems via triangular decomposition}, Ph.D. thesis, RWTH Aachen, Germany, 2014.

[BGLHR12] Thomas Bächler, Vladimir Gerdt, Markus Lange-Hegermann, and Daniel Robertz, \textit{Algorithmic Thomas decomposition of algebraic and differential systems}, J. Symbolic Comput. 47 (2012), no. 10, 1233–1266. (arXiv:1108.0817). MR 2926124

[BKLH19a] Mohamed Barakat, Tom Kuhmichel, and Markus Lange-Hegermann, \textit{Locales – Locales, frames, coframes, lattices of locally closed subsets, and lattices of constructible sets}, 2018–2019, (https://homalg-project.github.io/pkg/Locales).

[BKLH19b] Mohamed Barakat, Tom Kuhmichel, and Markus Lange-Hegermann, \textit{ZariskiFrames – (Co)frames/Locales of Zariski closed/open subsets of affine, projective, or toric varieties}, 2018–2019, (https://homalg-project.github.io/pkg/ZariskiFrames).

[BL10] Yuval Bistritz and Alexander Lifshitz, \textit{Bounds for resultants of univariate and bivariate polynomials}, Linear Algebra and its Applications 432 (2010), no. 8, 1995–2005.

[BLH11a] Mohamed Barakat and Markus Lange-Hegermann, \textit{An axiomatic setup for algorithmic homological algebra and an alternative approach to localization}, J. Algebra Appl. 10 (2011), no. 2, 269–293. MR 2795737 (2012f:18022)

[BLH11b] Mohamed Barakat and Markus Lange-Hegermann, \textit{LocalizeRingForHomalg: localize commutative rings at maximal ideals}, ACM Commun. Comput. Algebra 44 (2011), 201–204.
AN ALGORITHMIC APPROACH TO CHEVALLY’S THEOREM

[BLH14] Thomas Bächler and Markus Lange-Hegermann, *Algebraic Thomas and Differential Thomas: Thomas decomposition for algebraic and differential systems*, 2008-2014, (http://www.mathb.rwth-aachen.de/thomasdecomposition/).

[BLH19a] Mohamed Barakat and Markus Lange-Hegermann, *uMPS(2,2,4)*, 2019, (https://homalg-project.github.io/nb/uMPS224).

[BLH19b] Mohamed Barakat and Markus Lange-Hegermann, *uMPS(2,2,5)*, 2019, (https://homalg-project.github.io/nb/uMPS225).

[BLH20a] Mohamed Barakat and Markus Lange-Hegermann, *Cavender-Farriss-Neyman*, 2020, (https://homalg-project.github.io/nb/CFN_reduced).

[BLH20b] Mohamed Barakat and Markus Lange-Hegermann, *uMPS(2,2,3)*, 2020, (https://homalg-project.github.io/nb/uMPS223).

[BLHP17] Mohamed Barakat, Markus Lange-Hegermann, and Sebastian Posur, *Elimination via saturation*, (arXiv:1707.00925), 2017.

[Bor91] Armand Borel, *Linear algebraic groups*, second ed., Graduate Texts in Mathematics, vol. 126, Springer-Verlag, New York, 1991. MR 1102012.

[Bou72] N. Bourbaki, *Commutative algebra*, Elements of Mathematics, Hermann, Paris, 1972.

[Buc83] Bruno Buchberger, *A note on the complexity of constructing gröbner-bases*, European Conference on Computer Algebra, Springer, 1983, pp. 137–145.

[BV03] Rosemary Baines and Peter Vámos, *An algorithm to compute the set of characteristics of a system of polynomial equations over the integers*, J. Symbolic Comput. 35 (2003), no. 3, 269–279. MR 1962795.

[Car56] Henri Cartan, *Variétés algébriques affines*, Séminaire Henri Cartan 8 (1955-1956), 12 p. (fr), Talk no. 3.

[CC56] Claude Chevalley and Henri Cartan, *Schémas normaux : morphismes ; ensembles constructibles*, Séminaire Henri Cartan 8 (1955-1956), 10 p. (fr), Talk no. 7.

[CGL+07] Changbo Chen, Oleg Golubitsky, François Lemaire, Marc Moreno Maza, and Wei Pan, *Comprehensive Triangular Decomposition*, Computer Algebra in Scientific Computing (Berlin, Heidelberg) (Victor G. Ganzha, Ernst W. Mayr, and Evgenii V. Vorozhtsov, eds.), Lecture Notes in Computer Science, vol. 4770, Springer Berlin Heidelberg, 2007, pp. 73–101.

[CLS11] David A. Cox, John B. Little, and Henry K. Schenck, *Toric varieties*, Graduate Studies in Mathematics, vol. 124, American Mathematical Society, Providence, RI, 2011. MR 2810322.

[CMS19] Adam Czapliński, Mateusz Michałek, and Tim Seynnaeve, *Uniform matrix product states from an algebraic geometry’s point of view*, (arXiv:1904.07563), 2019.

[DGPS19] Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann, *SINGULAR 4-1-2 — A computer algebra system for polynomial computations*, http://www.singular.uni-kl.de, 2019.

[GD67] Alexandre Grothendieck and Jean Dieudonné, *Eléments de géométrie algébrique IV*, Publications Mathématiques, vol. 20, 24, 28, 32, Institute des Hautes Études Scientifiques., 1964-1967.

[GP19] Sebastian Gutsche and Sebastian Posur, *CAP: categories, algorithms, programming*, Computeralgebra-Rundbrief, 64, 14–17, March, 2019, (https://fachgruppe-computeralgebra.de/data/CA-Rundbrief/car64.pdf), pp. 14–17.

[GPS18] Sebastian Gutsche, Sebastian Posur, and Øystein Skartsøterhagen, *On the syntax and semantics of CAP*, In: O. Hasan, M. Pfeiffer, G. D. Reis (eds.): Proceedings of the Workshop Computer Algebra in the Age of Types, Hagenberg, Austria, 17-Aug-2018, published at http://ceur-ws.org/Vol-2307/, 2018.

[GW10] Ulrich Görtz and Torsten Wedhorn, *Algebraic geometry*, Springer, 2010.

[Har77] Robin Hartshorne, *Algebraic geometry*, Springer-Verlag, New York, 1977, Graduate Texts in Mathematics, No. 52. MR MR0463157 (57 #3116).

[Har13] Joe Harris, *Algebraic geometry: a first course*, vol. 133, Springer Science & Business Media, 2013.
Corey Harris, Mateusz Michał ek, and Emre Can Sertöz, *Computing images of polynomial maps*, Adv. Comput. Math. 45 (2019), no. 5-6, 2845–2865, (arXiv:1801.00827). MR 4047019

Jens Carsten Jantzen, *Nilpotent orbits in representation theory*, Lie theory, Progr. Math., vol. 228, Birkhäuser Boston, Boston, MA, 2004, pp. 1–211. MR 2042689

Christine Jost, *Computing characteristic classes and the topological Euler characteristic of complex projective schemes*, J. Softw. Algebra Geom. 7 (2015), 31–39. MR 3484270

Gregor Kemper, *A course in commutative algebra*, vol. 256, Springer Science & Business Media, 2010.

Daniel Lazard, *Gröbner bases, gaussian elimination and resolution of systems of algebraic equations*, European Conference on Computer Algebra, Springer, 1983, pp. 146–156.

F. Lemaire, M. Moreno Maza, and Y. Xie, *The RegularChains library in MAPLE*, SIGSAM Bull. 39 (2005), no. 3, 96–97.

Miguel A. Marco-Buzunáriz, *A polynomial generalization of the Euler characteristic for algebraic sets*, J. Singul. 4 (2012), 114–130, With an appendix by J. V. Rennemo. MR 3044490

M. Moreno Maza, *On triangular decompositions of algebraic varieties*, Tech. report, presented at the MEGA-2000 Conference, 1999.

Ernst W Mayr and Stephan Ritscher, *Dimension-dependent bounds for gröbner bases of polynomial ideals*, Journal of Symbolic Computation 49 (2013), 78–94.

Antonio Montes and Michael Wibmer, *Gröbner bases for polynomial systems with parameters*, J. Symbolic Comput. 45 (2010), no. 12, 1391–1425. MR 2733386

W. Plesken and T. Bächler, *Counting polynomials for linear codes, hyperplane arrangements, and matroids*, Documenta Math. 19 (2014), no. 9, 247–284.

W. Plesken, *Counting solutions of polynomial systems via iterated fibrations*, Arch. Math. (Basel) 92 (2009), no. 1, 44–56. MR MR2471987 (2010a:13051)

Vladimir L. Popov, *Algebraic groups whose orbit closures contain only finitely many orbits*, (arXiv:1707.06914), 2017.

Sebastian Posur, *Linear systems over localizations of rings*, Arch. Math. (Basel) 111 (2018), no. 1, 23–32. MR 3816974

Sebastian Posur, *Methods of constructive category theory*, (arXiv:1908.04132), (to appear in Contemporary Mathematics, AMS), Aug. 2019.

K. S. Sibirskii, *Algebraic invariants for a set of matrices*, Siberian Mathematical Journal 9 (1968), no. 1, 115–124.

The Stacks Project Authors, *stacks project*, http://stacks.math.columbia.edu, 2019.

Ravi Vakil, *The rising sea: Foundations of algebraic geometry*, 2010/11/05.

Gilles Villard, *On computing the resultant of generic bivariate polynomials*, Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, 2018, pp. 391–398.

D. Wang, *epsilon: description, help pages and examples of use*, 2003, (http://wang.cc4cm.org/epsilon/). 19