# Specialized strategies for resolution of singularities of determinantal ideals

Von der

Fakultät für Mathematik und Naturwissenschaften

der

Carl von Ossietzky Universität Oldenburg

zur Erlangung des Grades und Titels

Doctor rerum naturalium (Dr. rer. nat.)

angenommene Dissertation von

Frau Sabrina Alexandra Gaube

geboren am 13.09.1995 in Peine

Gutachter: Prof. Dr. Anne Frühbis-Krüger
weitere Gutachter: Prof. Dr. Gerhard Pfister
Tag der Disputation: 19.07.2023

# Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichungen, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Peine, den 17. Mai 2023

Sabrina Alexandra Gaube

## Zusammenfassung

Auflösung von Singularitäten spielt eine wichtige Rolle in der Arbeit von algebraischen Geometern seit dem Ende des 19. Jahrhunderts.

In Charakteristik 0 ist bekannt, dass immer eine Auflösung durch eine endliche Folge von Aufblasungen existiert, wie H. Hironaka im Jahr 1964 bewies. In positiver Charakteristik ist dies oberhalb kleinster Dimensionen eines der grundlegendsten offenen Probleme der Algebraischen Geometrie. Lediglich für Spezialfälle, wie bspw. Binomialideale ist dieses Problem bereits gelöst.

Wir werden in dieser Arbeit die Menge der Spezialfälle mit dem nächstschwierigeren Fall, nämlich einer speziellen Klasse determinantieller Ideale, erweitern. Dabei nutzen wir die bisherigen Kenntnisse über die Auflösung von Binomialidealen und den determinantiellen Singularitäten, erzeugt durch generische Matrizen, aus. Diese spezielle Klasse verallgemeinert die determinantiellen Singularitäten generischer Matrizen zum Fall der determinantiellen Singularitäten monomialer und binomialer Matrizen.
Außerdem werden wir einen effizienteren Weg sehen, (schief-)symmetrische generische determinantielle Singularitäten aufzulösen. Dieser Weg wird die (Schief-)Symmetrie grundlegend ausnutzen.

Insgesamt zeigt sich, dass wir, durch eine geeignete Implementation der präsentierten Strategie, die Klasse, der in der Praxis berechenbaren Auflösungen durch geeignete Zentrumswahlen und Ausnutzen struktureller Eigenschaften erweitern können.



## Abstract

The resolution of singularities has played a crucial role in the work of algebraic geometers since the end of the 19th century.

In characteristic 0, it is known that a resolution of singularities by a finite sequence of blow-ups always exists, as H. Hironaka proved in 1964. In positive characteristic, this is one of the most fundamental open problems in algebraic geometry above the smallest dimensions. Only for special cases, such as binomial ideals, has this problem already been solved.

In this thesis, we extend the set of special cases with the next more complex case, namely a special class of determinantal ideals. In doing so, we exploit previous

knowledge of the resolution of binomial ideals and the determinantal singularities of generic matrices. This special class generalizes the determinantal singularities of generic matrices to the next most challenging case: the determinantal singularities of at most binomial type.

Overall, by implementing the presented strategy, we can extend the class of resolutions computable in practice by appropriate center choices and exploiting structural properties.

# Contents

# 1. Introduction

This chapter presents the basic definitions, notations, and preliminaries required in the later chapters and further outlines this thesis' structure.

## 1.1. Motivation

Resolution of singularities plays a crucial role in the work of many algebraic geometers since the end of the 19th century. The problem of resolution of singularities asks whether every algebraic variety or, more generally, every sufficiently nice scheme $X$ has a desingularization, i.e., whether there is a non-singular variety or scheme $W$ with a proper birational map $W \to X$.
Hironaka [54] proved the existence of resolution of singularities for arbitrary dimensional algebraic varieties over fields of characteristic 0 in 1964. This proof consists of more than 200 pages with many technical details. The proof has constructive points, but there are crucial non-constructive points.

It took a quarter of a century to fill the non-constructive steps with algorithmic details with the pioneering work of Bierstone and Milman [6], [7], and Villamayor [76] and [77].

Further, the arguments were simplified, for example, by Bravo, Encinas, and Villamayor [16], Cutkosky [26], Encinas Hauser [33], Hauser [51], Kollár [63] and Włodarczyk [79] in order to understand the problems of the proof in positive characteristic.

There exist implementations of a Hironaka-style resolution (more precisely, the simplification by Villamayor [16]). One of them is implemented by Bodnár and Schicho in the computer algebra system maple [13] and another is written in the computer algebra system SINGULAR by Frühbis-Krüger and Pfister (see [37]).

In positive characteristic, it is still an open problem for dimensions higher than 3. The first result for some special cases was shown by Abhyankar [1]. He showed the resolution of singularities of surfaces and the three-dimensional case in characteristic greater than 5 for algebraically closed base fields. Simplified proofs can be found in [27] and [28].

For curves, there are two fundamental strategies by normalization and by blow-ups. In dimension 2, there is, for example, the algorithm of Lipman [65] and the algorithm of Cossart, Jannsen, and Saito [20]. Whereas the first deals with alternating normalization and blow-up in finitely many isolated singular points, the second one is achieved by a canonical sequence of blow-ups (see Section 5.6). Normalizations in combination with blow-ups do not yield an embedded resolution for dimensions bigger than 2 since the normalization destroys the embedding. So it can not be generalized for a higher-dimensional case. That is why we do not want to consider the algorithm of Lipman here. The strategy of Cossart, Jannsen, and Saito terminates for arbitrary excellent schemes of dimension at most 2 in positive and also mixed characteristic (see also [35]). For surfaces, this strategy completely dispenses with the use of coefficient ideals.

While it is formulated for arbitrary dimension, it is unknown if the approach of Cossart, Jannsen, and Saito terminates for higher dimensions and whether this property holds then, too, in particular not in characteristic 0.
In dimension 3, there are some results on weak resolution of singularities. Zariski [82] proved the weak resolution of singularities for threefolds over a field $K$ of characteristic 0. Cossart and Piltant [22] prove the existence of a birational and global resolution in dimension 3. Originally, the base field was assumed to be differentially finite over a perfect field. In newer articles this requirement has been dropped [24] and [23]. This result is not constructively given through an algorithm. It is unclear, if the resolution is achieved purely by blow-ups in regular centers.

Furthermore, some algorithms yield a resolution of arbitrary characteristic and dimensions for a restricted class of singularities – namely binomial or toric singularities. Many arguments become much easier when dealing with the restricted class of binomial ideals. The problem becomes more combinatorial in this case, whereas the general problem is more geometric. With these ideas Blanco (see [9] and [10], see also Section 5.4) could apply the ideas of Bierstone and Milman and Bravo, Encinas, and Villamayor to arbitrary binomial singularities. There is also an implementation by Blanco and Pfister [12] in the computer algebra system SINGULAR. This implementation was a proof-of-concept implementation. Unfortunately, the structure of the implementation is not modular enough to use it as a blackbox. We only need a submethod which calculates the center and updates every data for next blow-up. That is neither given in RESBINOMIAL.LIB nor in RESOLVE.LIB. So the author reimplements the version for resolution of binomial ideals in characteristic zero. More information can be found in the Appendix in Section A.3.

Furthermore, there is an algorithm of Blanco, and Encinas [11] for toric varieties, which is more or less the same algorithm with a small change of the invariant, so it does not depend on a choice of a Gröbner basis, and there is the algorithm of Bierstone and Milman [8] for toric singularities.
A weaker condition of the structure is a determinantal singularity for which a generating matrix has monomial entries. If the entries are generic, the first result is by [75] and was later generalized in [71]. See also Section 6.1 for details.
Dropping the generality, we arrive at the goal of this thesis. The goal of this thesis is to combine these different approaches to get a resolution strategy for a special class of determinantal singularities. For a determinantal singularity in this class, there exists a generating matrix with, at most binomial entries, and we will see that in characteristic 0, this leads us to a resolution strategy of arbitrary determinantal singularities. Furthermore, we will see that we are not restricted to binomials but to entries for which there exists a constructive simultaneous local monomialization procedure.

In this thesis, we first give an overview of the preliminaries (Section 1.2) and the goal (Section 1.3). In Chapter 2, we give the most important definitions and fundamental constructions in the resolution of singularities. So we first illustrate the idea of resolution of singularities with the so-called polyhedral game of Hironaka (Section 2.1), which gives an intuition on the construction of the blow-up (Section 2.2). Afterwards, we can define the task of resolution of singularities (Section 2.3) and different measures that are suitable to measure improvements during the process of resolution of singularities (Section 2.4).
The next chapter (Chapter 3) presents the most accessible resolution strategies for curves and surfaces.
Then, in Chapter 4, we are able to formulate a simplified local version of the main idea of resolution of determinantal singularities.

Afterwards, in Chapter 5, we illustrates the idea of a general Hironaka-style resolution and the main ideas of the algorithm of Villamayor. Then we restrict the considered singularities to the class of singularities which are generated by binomial ideals and discuss the algorithm of Blanco and Encinas. Section 5.5 discusses the algorithm of Hu for the resolution of simple arrangements, and finally we formulate the algorithm of Cossart, Jannsen and Saito in Section 5.6.

Then we can present the approach in the generic determinantal case, the

symmetric generic determinantal case, and the skewsymmetric generic determinantal case in Chapter 6.

In Chapter 7, we go back to binomial hypersurfaces and compare a strategy of local monomialization of a binomial considering a complexity analysis of different choices of centers.

Everything comes together in Chapter 8, where the main result is presented by the main algorithm (Construction 8.2.2).

Chapter 9 gives a summary and formulates future work that is still to be done.

In the Appendix, one can find the main ideas in implementations of the presented strategies in Chapter A. Since the algorithm of Villamayor has an implementation in the computer algebra system Singular, we give the main ideas of the implementation of the author of the algorithms of Cossart, Jannsen, and Saito (dimension-free) (Section A.2), Hu (Section A.4), the generic determinantal case in arbitrary characteristic and the non-generic determinantal case of at most binomial type in characteristic 0 (Section A.5). This main algorithm only needs the choice of center of [12] or [37]. Both implementations have not the neccessary structure, so the author also rewrote the algorithm of Blanco (Section A.3), and her main ideas can be read in the Appendix, too.

Furthermore, Chapter B gives a brief illustration of the given complexity results concerning the number of considered charts of the different methods, which yield a comparison of the several implementations of the author and given implementations in Singular.

Unit tests of the implementation of the main algorithm are described in Chapter C.

## 1.2. Preliminaries

Before we specify our goal, we have to discuss the preliminaries in this section.

Throughout this thesis, we fix $K$ to be an algebraically closed field and we use multiindex notation. We write $\underline{x}^A = x_1^{A_1} \cdots x_n^{A_n}$, for $A = (A_1, \ldots, A_n) \in \mathbb{Z}_{\geq 0}^n$ and $|A| := \sum_{i=1}^n A_i$. Furthermore, given a polynomial $f \in K[\underline{x}]$, we write $D_+(f) = D(f) = \mathbb{A}_K^n \backslash V(f) = \{p \in \mathbb{A}_K^n \mid f(p) \neq 0\}$ for the principal open set defined by $f$, where $V(f)$ denotes the vanishing locus of $f$. And we assume all schemes to be irreducible and reduced excellent noetherian schemes.

In order to formulate the goal of this thesis, we have to define the property of

being singular.

**Definition 1.2.1.** Let $X$ be an irreducible reduced excellent noetherian scheme.

- A point $x \in X$ is called regular (or non-singular), if the local ring $(\mathcal{O}_{X,x}, \mathfrak{m}_{X,x})$ is regular, which means, that

$$\dim_{k(x)} \mathfrak{m}_x/\mathfrak{m}_x^2 = \dim \mathcal{O}_{X,x},$$

  where $k(x) = \mathcal{O}_{X,x}/\mathfrak{m}_x$ denotes the residue field.

- A point $x \in X$ is called singular, if it is not regular.

- $X$ is called regular (or non-singular), if every $x \in X$ is regular.

- $X$ is called singular, if there exists an $x \in X$ which is singular.

Note, that for perfect fields regular is equivalent to the term of being smooth.

**Definition 1.2.2** ([26, Definition 2.6])**.** Let X be a variety of dimension $s \in \mathbb{N}$ over a field $K$ and let $P \in X$. Furthermore, let $U = \mathrm{Spec}(R)$ be an affine neighborhood of P such that $R \cong K[x_1, \ldots, x_n]/I$ with $I = \langle f_1, \ldots, f_m \rangle$. Then $X$ is smooth over $K$ at $P$, if the jacobian matrix of $I$ with respect to $\underline{x}$ has rank $n - s$ at $P$.

**Remark 1.2.3.** Let $X$ be a variety over a perfect field $K$ and let $p \in X$. Then $X$ is regular at $p$ if and only if $X$ is smooth at $p$.
If $K$ is not a perfect field, regularity implies smoothness but not vice versa.

If we want to consider resolutions of a singular scheme $X$, we need to say what is meant by this term. The main goal is to find a scheme $\tilde{X}$ which has the same properties as $X$ outside its singular locus, which we want to denote with $\mathrm{Sing}(X)$. $\tilde{X}$ should also be regular. More generally:

**Definition 1.2.4.** An embedded resolution of singularities of a given scheme $X \subset Z$ with $Z$ regular, is a proper, birational morphism $\pi : Z' \to Z$, for some regular scheme $Z'$, such that

(a) the reduced strict transform $X'_{\mathrm{red}}$ of $X$ in $Z'$ is regular,

(b) $\pi$ is an isomorphism outside of the singular locus of $X_{\mathrm{red}}$, i.e.,

$$\pi^{-1}(Z \setminus \mathrm{Sing}(X_{\mathrm{red}})) \cong Z \setminus \mathrm{Sing}(X_{\mathrm{red}}),$$

(c) $\pi^{-1}(\mathrm{Sing}(X_{\mathrm{red}}))$ is a simple normal crossings divisor, i.e., all irreducible components are regular and they intersect transversally, which intersects $X'_{\mathrm{red}}$ transversally.

The property of being a simple normal crossing divisor says that the intersection behaviour should be as simple as possible.

**Definition 1.2.5** ([16, Definition 2.1])**.** Let $W$ be a regular scheme and let $Y_1, \ldots, Y_k \subset W$ be a set of closed subschemes. $Y_1 \cup \cdots \cup Y_k$ are said to have normal crossings at a point $x \in W$ if there exists a regular system of parameters $\{x_1, \ldots, x_d\} \subset \mathcal{O}_{W,x}$ such that for each $i \in \{1, \ldots, k\}$ either $\mathcal{I}(Y_i)_x = \mathcal{O}_{W,x}$ or $\mathcal{I}(Y_i)_x = \langle x_{i_1}, \ldots, x_{i_s} \rangle$ for some $x_{i_1}, \ldots, x_{i_s} \in \{x_1, \ldots, x_d\}$.
We say that $Y_1 \cup \cdots \cup Y_k$ have normal crossings in $W$ if they have normal crossings at any point of $W$.

More definitions of several 'levels' of resolution of singularities are given in Section 2.3.1.

We want to use the approach of singularities generated by binomial ideals, so we state the definition of a binomial ideal here.

**Definition 1.2.6.** An ideal $I$ is called binomial ideal if there exists a set of generators $\{f_1, \ldots, f_r\}$ such that each $f_i = \underline{x}^{A_i} - \lambda_i \underline{x}^{B_i} \in K[\underline{x}]$ is a binomial, where $\lambda_i \in K^\times$ and $\underline{x}^{A_i} = x_1^{A_{i_1}} \cdots x_n^{A_{i_n}}$ for $A_i = (A_{i_1}, \ldots, A_{i_n}) \in \mathbb{Z}^n_{\geq 0}$ for all $1 \leq i \leq r$.

## 1.3. Goal

We aim to generalize the approaches in the resolution of binomial ideals and the resolution of the generic matrix case. We want to combine them with some approaches, e.g., the resolution of arrangements of hypersurfaces (see Section 5.5).

We want to generalize these approaches to get an algorithm that resolves determinantal singularities.

The next step is to consider a special case of determinantal singularities, namely the determinantal singularities, which are generated by matrices with, at most binomial entries to prove the correctness and the terminancy.

**Definition 1.3.1** (determinantal singularity of at most binomial type)**.** Let $K$ be an algebraically closed field and let $M = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,m} \end{pmatrix} \in (K[\underline{x}])^{n \times m}$,

where $f_{i,j} = \underline{x}^{C_{i,j}}(\underline{x}^{A_{i,j}} - \lambda_{i,j}\underline{x}^{B_{i,j}})$ are binomials.

Let $I_t$ be the ideal generated by the $t$-minors of $M$, for all $1 \leq t \leq \min\{n, m\}$. Then $I_t$ is called a determinantal singularity of at most binomial type.

The main question is, how to resolve the singularities of $V(I_t)$, where $I_t$ is an ideal like in Definition 1.3.1.
We are in the tension field between the completely open problem in positive characteristic and the binomial case, which is completely solved. We want to move the boundary of actually computable resolution of singularities. We will see the following results:

**Main Theorem 1.** *In arbitrary characteristic the algorithm in Construction 8.2.2 provides a resolution of arbitrary determinantal singularities of at most binomial type.*

**Main Theorem 2.** *In characteristic* 0 *the algorithm in Construction 8.2.2 provides a resolution of arbitrary determinantal singularities.*

**Main Theorem 3.** *If there is a constructive procedure which principalizes the entries and establishes normal crossings (and a covering) to the entries of a matrix that generates the determinantal singularity, then the algorithm in Construction 8.2.2 provides a resolution.*

In order to reduce complexity, we will also restrict to (skew-)symmetric generic determinantal singularities and see the following results:

**Main Theorem 4.** *Let $m, \ell \in \mathbb{Z}_+$ with $2\ell \leq m$, let $R_0$ be a regular ring with $\operatorname{char}(R_0) \neq 2$, and let $A_m := (x_{i,j})_{i,j}$ be the generic skew-symmetric $m \times m$ matrix with entries in $R_0[x_{i,j} | 1 \leq i \leq j \leq m]$, i.e., $x_{i,j} = -x_{j,i}$ for all $i, j \in \{1, \ldots, m\}$.*

*The following sequence of blowing ups is an embedded resolution of singularities for the generic skew-symmetric determinantal singularity $Y^{\text{skew}}_{m,2\ell} \subset Z_{\text{skew}}$,*

$$Z_{\text{skew}} =: Z_0 \xleftarrow{\pi_1} Z_1 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_{\ell-1}} Z_{\ell-1},$$

*where $\pi_i$ is the blowing up with center the strict transform of $(Y^{\text{skew}}_{m,2i})_{\text{red}}$ in $Z_{i-1}$, for $1 \leq j \leq \ell - 1$.*

**Main Theorem 5.** *Let $m, r \in \mathbb{Z}_+$ with $r \leq m$, let $R_0$ be a regular ring, and let $B_m := (x_{i,j})_{i,j}$ be the generic symmetric $m \times m$ matrix with entries in $R_0[x_{i,j} | 1 \leq i \leq j \leq m]$, i.e., $x_{i,j} = x_{j,i}$ for all $i, j \in \{1, \ldots, m\}$.*

*The following sequence of blowing ups is an embedded resolution of singularities for the generic symmetric determinantal singularity* $Y^{\mathrm{sym}}_{m,r} \subset Z_{\mathrm{sym}}$*,*

$$Z_{\mathrm{sym}} =: Z_1 \xleftarrow{\pi_1} Z_2 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_{r-1}} Z_r$$

*where* $\pi_j$ *is the blowing up with center the strict transform of* $Y^{\mathrm{sym}}_{m,j}$ *in* $Z_j$*, for* $1 \leq j \leq r-1$*.*

Why should we consider these determinantial singularities?

- It is interesting because of being an open problem for characteristic $p > 0$ and dimension $> 3$.
- We may get a more efficient resolution than a general Hironaka-style resolution in characteristic 0.
- We can reduce it to a binomial resolution and the resolution of a determinantal singularity generated by a generic matrix, which is solved in arbitrary characteristic.

# 2. Theoretical background

In this chapter, we will discuss the main constructions that are used for the resolution of singularities. Therefore we firstly discuss the construction of the blow-up (Section 2.2). The idea is to blow-up the most singular points and take advantage of that blow-ups are proper birational morphisms which are isomorphisms outside the center of the blow-up. So we can improve the singular points without changing the regular points. We will illustrate in Section 2.1 via Hironaka's polyhedral game the construction of blow-ups, which only should give an intuition of how blow-ups come into play at the resolution of singularities.
Then (Section 2.3), we talk about resolution of singularities itself. We give a short overview of several resolution problems in Section 2.3.1. The data which has to be stored during the calculation of the resolution is stored in basic objects or idealistic exponents. We discuss them in Section 2.3.2 and Section 2.3.3. At the end of this chapter, in Section 2.4, we discuss how to measure singular points. We define the multiplicity in Section 2.4.1 and focus on the order, and the refined order of [35] in Section 2.4.2.

## 2.1. Hironaka's polyhedral game

In this section, we introduce the polyhedral game of Hironaka in the version that Hauser stated in [51, page 324]. This illustrates one way how singularities will be resolved. One round in the game is analogous looking at the origin of one chart ($j$-chart) expression of a blow-up with chosen center $J$.

Let us formulate the polyhedral game. Let $A$ be a set of points in $\mathbb{N}^n$ and let $N$ be the positive convex hull in $\mathbb{R}^n$, $N = \text{conv}(A) + \mathbb{R}^n_{\geq 0}$. Figure 2.1 illustrates the situation.

The game has two players, $\mathcal{P}_1$ and $\mathcal{P}_2$. Player $\mathcal{P}_1$ starts by choosing a non-empty subset $J \subset \{1, \ldots, n\}$ and afterwards player $\mathcal{P}_2$ picks a number $j \in J$.
After the players have made their choices the set $A$ is replaced by the set $A'$ obtained from $A$ by substituting the $j$-th component of the vectors $\alpha$ in $A$ by the

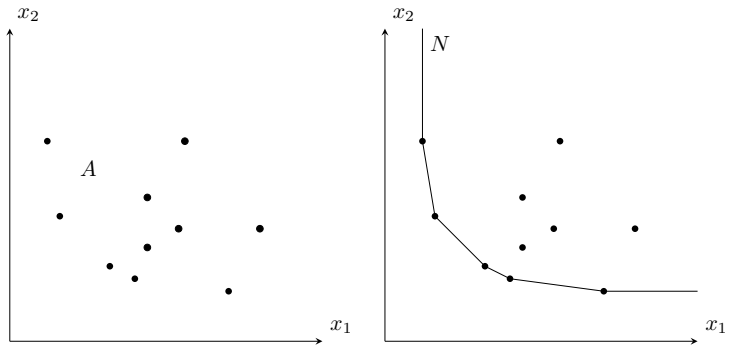



Figure 2.1.: Convex hull of a set $A$ of points

sum of the components $\alpha_i$ with $i \in J$, so $\alpha_j \mapsto \alpha'_j = \sum_{i \in J} \alpha_i$ holds.
The other components remain the same, so $\alpha_k = \alpha_{k'}$ for every $k \neq j$. And we set $N' = \text{conv}(A') + \mathbb{R}^n_{\geq 0}$.
After this round, the next round starts again with $N'$ instead of $N$, and players $\mathcal{P}_1$ and $\mathcal{P}_2$ have to make the analogous choices as in the round before. This procedure is repeated.

Player $\mathcal{P}_1$ wins if the polyhedron $N$ has become an orthant $N = \alpha + \mathbb{R}^n_{\geq 0}$ for some $\alpha \in \mathbb{N}^n$ after finitely many steps. If this is not the case, player $\mathcal{P}_2$ wins.

The interesting question in this game is whether player $\mathcal{P}_1$ always possesses a winning strategy.

We have a look at the game with 2 variables. So $n = 2$ holds. If $\mathcal{P}_1$ always chooses $J = \{1\}$ or $J = \{2\}$, the transformation is the identity map, so $N' = N$ and player $\mathcal{P}_1$ loses the game. So player $\mathcal{P}_1$ has to choose $J = \{1, 2\}$ and player $\mathcal{P}_2$ can choose $j = 1$ or $j = 2$.
In Figure 2.2 we see the transformation of $N$, if player $\mathcal{P}_2$ chooses $j = 1$.

If player $\mathcal{P}_2$ chooses $j = 2$ afterwards, the vertices move vertically and yield a polyhedron $N''$ which is in this case already a quadrant (see Figure 2.3).

Now we have given a rough intuition of how blow-ups works. The technical details behind the idea can be read in Section 2.2.

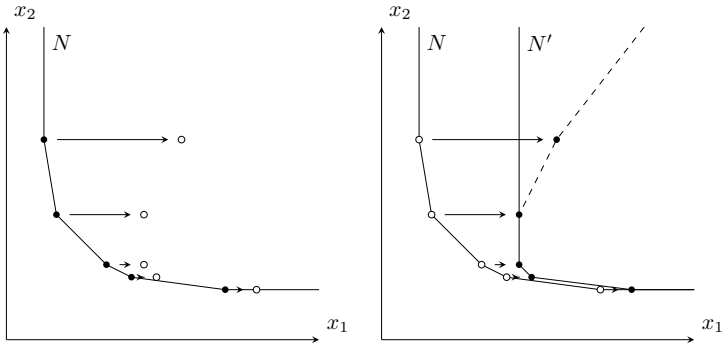



Figure 2.2.: Transform of the polyhedron $N$

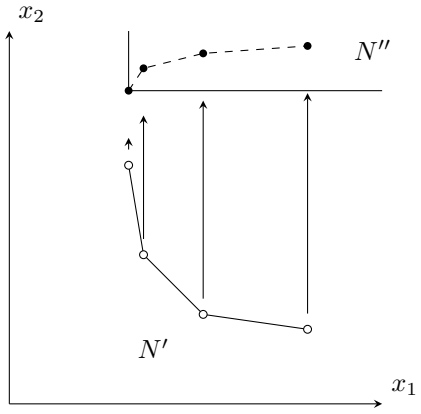



Figure 2.3.: Transform of the polyhedron $N'$

## 2.2. Blow-up

In Chapter 3 and Chapter 5, we will discuss some of the most known algorithmic strategies for the resolution of singularities. All strategies discussed in this thesis deal with blow-ups, which we introduce below: The resolution of singularities with these blow-ups illustrated by the Hironaka polyhedral game in Section 2.1.

**Definition 2.2.1** ([49, page 163])**.** Let $W$ be a scheme, $Y \subset W$ a subscheme associated by the coherent ideal sheaf $\mathcal{J}$. The blow-up of $W$ with center $Y$ is defined by

$$\pi\colon \tilde{W} := \mathrm{Proj}(\bigoplus_{d\geq 0} \mathcal{J}^d) \to W.$$

**Remark 2.2.2.** Each blow-up is uniquely defined by its center $Y$.
The center is the locus of $W$ above which the map $\pi$ is not an isomorphism.

**Remark 2.2.3** ([51, page 332])**.** The role of a blow-up is to untie the singularities of some $X \subset W$ by looking at its inverse image $X'$ in $W'$. The scheme $W'$ offers $X'$ more 'space' to spread out than $W$ since $W'$ lives inside a higher dimensional ambient space. The main idea is to repeat this process until after (hopefully) finitely many blow-ups the final inverse image $\tilde{X}$ of $X$ has been resolved.

**Remark 2.2.4.** Let $R_0$ be a regular ring and let $Z = \mathrm{Spec}(R_0[t_1, \ldots, t_N]) \cong \mathbb{A}^N_{R_0}$. Consider the blowing up $\pi\colon B\ell_D(Z) \to Z$ of $Z$ with center $D = \mathrm{Spec}(R_0[t_1, \ldots, t_N]/\langle t_i \mid i \in I\rangle)$, for some $I \subseteq \{1, \ldots, N\}$. Globally, $B\ell_D(Z)$ is described by the Proj-construction as

$$B\ell_D(Z) = \mathrm{Proj}(R_0[t_1, \ldots, t_N][T_i \mid i \in I]/\langle t_iT_j - t_jT_i \mid i, j \in I\rangle),$$

where $(T_i \mid i \in I)$ are projective coordinates. In particular, $B\ell_D(Z)$ is covered by the affine charts $D_+(T_i)$ with $i \in I$, where $D_+(T_i)$ denotes the standard open set of points, where $T_i$ is invertible. We call $D_+(T_i)$ also the *$T_i$-chart* of the blowing up.

Fix $i \in I$. In the $T_i$-chart, the relation $t_iT_j - t_jT_i = 0$ can be rewritten as

$$t_j = t_i\frac{T_j}{T_i}, \quad \text{for } j \in I \setminus \{i\}.$$

This shows that the $B\ell_D(Z) \cap D_+(T_i)$ is isomorphic to

$$\mathrm{Spec}(R_0[t_k, t_i, \frac{T_j}{T_i} \mid k \notin I, j \in I \setminus \{i\}]) \cong \mathbb{A}^N_{R_0}.$$

Often the abbreviation $t'_j := \frac{T_j}{T_j}$ is used and by setting $t'_i := t_i$ and $t'_k := t_k$ for $k \notin I$, it is said that the variables of the $T_i$-chart are $(t'_1, \ldots, t'_N)$, where the relation to the

variables $(t_1, \ldots, t_N)$ before the blowing up is described by

$$t_j = \begin{cases} t'_i t'_j, & \text{if } j \in I \notin \{i\} \\ t'_j, & \text{otherwise.} \end{cases}$$

Note that the exceptional divisor of the blowing up $\pi$, i.e., the preimage of $D$ along $\pi$ (the locus where $\pi$ is not an isomorphism), is given by the divisor $\operatorname{div}(t'_i)$ in the $T_i$-chart.

For example, in the determinantal setting (Section 6 and Section 8), we have $(t_1, \ldots, t_N) = (x_{i,j} \mid i, j \in \{1, \ldots, m\})$ and we speak of the $X_{1,2}$-chart, or similar expressions. Besides reflecting the origin of the variables $(x_{i,j})$ coming from a matrix structure, the index set $\{1, \ldots, m\}^2$ has no impact or deeper meaning.

When resolving, e.g., single binomials, it is more efficient if we work combinatorical, i.e., on the exponents instead of storing whole polynomial structures in the computer. The polynomial structure is more complex than a simple list. So we give here an equivalent characterization of the blow-up in the case of canonical centers, e.g., if we resolve a single binomial (Section 7) or in the case of resolution of a binomial scheme (Section 5.4) as a map

$$B\ell_D(\mathbb{A}^n_K) \to \mathbb{A}^n_K.$$

**Definition 2.2.5.** The blow-up $\pi\colon B\ell_D(\mathbb{A}^n_K) \to \mathbb{A}^n_K$ in a regular center $D = V(x_i \mid i \in I)$, for some $I \subseteq \{1, \ldots, n\}$ is covered by the standard charts

$$U_i := D_+(x_i) \cong \mathbb{A}^n_K,$$

$i \in I$ , where $D_+(x_i) := B\ell_D(\mathbb{A}^n_K) \setminus V(x_i)$ is the complement of $V(x_i)$. On $U_i$, the blowup $\pi$ is given by the morphism

$$\begin{array}{rcll} K[x_1, \ldots, x_n] & \longrightarrow & K[x'_1, \ldots, x'_n] =: K[\underline{x}'] & \\ x_j & \mapsto & x'_i\, x'_j, & \text{if } j \in I \setminus \{i\}, \\ x_j & \mapsto & x'_j, & \text{if } j = i \text{ or } j \notin I, \\ \lambda & \mapsto & \lambda, & \text{for } \lambda \in K. \end{array}$$

The image of $f \in K[\underline{x}]$ in $K[\underline{x}']$ is called the total transform of $f$ in $U_i$.

Note, that both definitions are well-defined.

In Chapter 7, we will use this characterization as a map on the exponents explicitly.

**Remark 2.2.6.** On the level of exponents, the morphism in Definition 2.2.5 corresponds to the map

$$\phi_{\pi,i}\colon \mathbb{Z}^n_{\geq 0} \longrightarrow \mathbb{Z}^n_{\geq 0},$$

where $A = (A_1, \ldots, A_n)$ is mapped to $A' = (A'_1, \ldots, A'_n)$, which is defined by $A'_i := \sum_{j \in I} A_j$ and $A'_j := A_j$ if $j \neq i$.

In every $U_i \cong \mathbb{A}^n_K$, we may choose a center $D_i$ of the same shape as $D$, and we can iterate blow-ups to obtain a sequence of local blow-ups. See Section 7 for more information.

Now we can connect blow-ups and Hironaka's polyhedral game. Player $\mathcal{P}_1$ chooses an index set $J \subset \{1, \ldots, n\}$ which encodes information on the center $V(x_j \mid j \in J)$ of the corresponding blow-up. Player $\mathcal{P}_2$ chooses the considered chart. Player $\mathcal{P}_1$ wins the game if after finitely many blow-ups or in the notion of the game after finitely many steps Player $\mathcal{P}_2$ could not choose a chart such that the singularity is resolved resp. the convex hull $N^i$ is an ordinate for some $i \in \mathbb{N}$. That is why resolution of singularities exists if there is a winning strategy for Player $\mathcal{P}_1$. For more information on the notion of resolution of singularities, we refer to Section 2.3.

We give now some properties of the blow-up.

**Lemma 2.2.7** (Basic Properties of the blow-up [49, Proposition II.7.13])**.** *Let $\pi\colon \tilde{W} \to W$ be the blow-up of $W$ with center $Y \subset W$. Then the exceptional divisor of the blow-up $\pi^{-1}(\mathcal{J}\mathcal{O}_{\tilde{W}})$ is an invertible sheaf on $\tilde{W}$. Furthermore,*

$$\pi\colon \pi^{-1}(W \setminus Y) \to W \setminus Y$$

*is an isomorphism.*

The first basic property gives us information about the exceptional divisor. The second property that we have an isomorphism on the complement of our center says that we only change our scheme in the center of our blow-up. Thus the basic idea behind the resolution of singularities is to choose a 'bad' locus and iteratively improve it and keep the information of the 'good' loci.

**Lemma 2.2.8** (Universal Property of the blow-up [49, Proposition II.7.14])**.** *If $f\colon Z \to W$ is any morphism such that $f^{-1}\mathcal{J}\mathcal{O}_Z$ is an invertible sheaf on $Z$, there exists a unique morphism $g\colon Z \to \tilde{W}$ factoring $f$.*

**Lemma 2.2.9** ([49, Proposition II.7.16.(b)]). *Let $X$ be a variety over a field $K$ and let $J \subseteq \mathcal{O}_X$ be a nonzero coherent sheaf of ideals on $X$, and let $\pi\colon \tilde{X} \to X$ be the blow-up with center $J$. Then $\pi$ is a birational, proper, surjective morphism.*

**Lemma 2.2.10** (Strict Transform [36, Lemma 50]). *Let $Z_1 \overset{i}{\hookrightarrow} W$ be a closed subscheme. Let $\pi_1\colon Z_1' \to Z_1$ be the blow-up of $Z_1$ along $i^{-1}\mathcal{J}\mathcal{O}_{Z_1}$. Then the following diagram commutes*

$$\begin{array}{ccc} Z_1' & \hookrightarrow & W' \\ \pi_1 \downarrow & & \downarrow \pi \\ Z_1 & \hookrightarrow & W. \end{array}$$

*$Z_1'$ is called the strict transform of $Z_1$ under the blow-up $\pi\colon W' \to W$.*

When calculating explicit examples, it is more convenient to pass to a covering of the scheme $W$ by affine charts. In each of these charts we are in the following situation.

Let $U \subset W$ be an affine open subset and denote $A := \Gamma(U, \mathcal{O}_W)$ and $J := \Gamma(U, \mathcal{J}) = \langle f_1, \ldots, f_m \rangle \subseteq A$.

Now the blow-up of $U$ in center $Y \cap U$ is

$$\pi^{-1}(U) = \operatorname{Proj}(\bigoplus_{d \geq 0} J^d).$$

For an explicit computation, we can consider the canonical graded $A$-algebra homomorphism

$$\Phi\colon A[y_1, \ldots, y_m] \to \bigoplus_{n \geq 0} J^n t^n \subseteq A[t],$$

which is defined by $\Phi(y_i) = t f_i$. Then

$$\bigoplus_{n \geq 0} J^n \cong A[y_1, \ldots, y_m] / \operatorname{Ker}(\Phi).$$

**Construction 2.2.11** ([36, Page 58]). Let $I \subset \mathcal{O}_W$ be an ideal and $X$ the corresponding subscheme of $W$. The exceptional divisor and the different transforms

of $X$ under the blow-up of $W$ at center $Y$ can be computed in the following way:

| | |
|---|---|
| exceptional divisor | $I(H) = J\mathcal{O}_{\tilde{W}}$ |
| total transform | $\pi^\star(I) = I\mathcal{O}_{\tilde{W}}$ |
| strict transform | $I_{\tilde{X}} = (I\mathcal{O}_{\tilde{W}} : J\mathcal{O}_{\tilde{W}}{}^{\infty})$ |
| weak transform | $(I\mathcal{O}_{\tilde{W}} : J\mathcal{O}_{\tilde{W}}{}^{k})$ |
| controlled transform | $(I\mathcal{O}_{\tilde{W}} : J\mathcal{O}_{\tilde{W}}{}^{c})$, |

where $(I_1 : I_2{}^{\infty}) := \bigcup_{i\geq 0}(I_1 : I_2{}^{i})$ is the saturation of $I_1$ with respect to $I_2$, where $k = \max\{\ell \in \mathbb{N} \mid (I\mathcal{O}_{\tilde{W}} : J\mathcal{O}_{\tilde{W}}{}^{\ell-1}) = (I\mathcal{O}_{\tilde{W}} : J\mathcal{O}_{\tilde{W}}{}^{\ell}) \cdot J\mathcal{O}_{\tilde{W}}\}$ and where $c \in \mathbb{N}$ is called control.

**Remark 2.2.12.** The exceptional divisor of the blow-up is the inverse image of the center $Y$. So it is the locus of $W'$ where $\pi$ fails to be an isomorphism.

## 2.3. Fundamental concepts and constructions for resolution of singularities

In the first subsection (Section 2.3.1), we discuss several resolution problems and the connection between the principalization of ideals and the resolution of singularities. When constructing a resolution of singularities, it is necessary to consider more data than the singular points. So we have to discuss basic objects (Section 2.3.2) and idealistic exponents (Section 2.3.3).

### 2.3.1. Resolution of singularities and Principalization of ideals

We can find different 'levels' of resolution strategies in the literature. We want to present them here and indicate the relationship between the resolution of singularities and the principalization of ideals. While the existence of resolution of singularities is one of the most important results in the area, principalization of ideals is related to the classical problem of elimination of base points of a linear system.

We use the approach of [63, Section 3] here.

**Definition 2.3.1** (Weakest Resolution Problem)**.** Let $X$ be a variety. The weakest resolution problem is given by the question:
Is there a projective variety $X'$ such that $X'$ is smooth and birational to $X$?

**Remark 2.3.2.** An example for a solving algorithm of the weakest resolution problem for curves is given by the Albanese method.
The main idea is to take a curve $C$ that spans a projective space $\mathbb{P}^m$ of sufficiently large dimension ($m > 2 \cdot \deg(C)$) and repeatedly projecting down from singular points to projective spaces of smaller dimension.

**Definition 2.3.3** (Weak Resolution Problem)**.** Let $X$ be a variety. The weak resolution problem is given by the question:
Is there a variety $X'$ and a birational proper morphism $\pi\colon X' \to X$ such that $X'$ is smooth ?

**Definition 2.3.4** (Strong Resolution Problem)**.** Let $X$ be a variety. The strong resolution Problem is given by the question:
Is there a variety $X'$ and a projective morphism $\pi\colon X' \to X$ such that the following holds:

1. $X'$ is smooth and $\pi$ is birational,
2. $\pi\colon \pi^{-1}(X \setminus \mathrm{Sing}(X)) \to X \setminus \mathrm{Sing}(X)$ is an isomorphism and
3. $\pi^{-1}(\mathrm{Sing}(X))$ is a divisor with simple normal crossings.

Note, that blow-ups in smooth centers are projective birational maps, so we can formulate these 'levels' in the notion of blow-ups.

**Definition 2.3.5** (Resolution by blowing up smooth centers Problem)**.** Let $X$ be a variety. The resolution by blowing up smooth centers problem is given by the question:
Is there a resolution $\pi_X\colon X' \to X$ such that $\pi_X$ is a composite of morphisms

$$\pi_X\colon X' = X_n \xrightarrow{\pi_{n-1}} X_{n-1} \xrightarrow{\pi_{n-2}} \cdots \xrightarrow{\pi_1} X_1 \xrightarrow{\pi_0} X_0 = X,$$

where each $\pi_i\colon X_{i+1} \to X_i$ is obtained by blowing up a smooth variety $Z_i \subset X_i$.

**Remark 2.3.6** (Strong Resolution by blowing up smooth centers Problem)**.** If we are in the situation of Definition 2.3.5 but we want $\pi_X$ to be a strong resolution, then we need additional requirements on $Z_i$. Often $Z_i \subset \mathrm{Sing}(X_i)$ is required, although we only need that $\pi_0 \cdots \pi_{i-1}(Z_{i-1}) \subset \mathrm{Sing}(X)$.

See Section 3 and 5 for examples.

Furthermore, every variant of resolution of singularities also has an embedded version. The most famous version is the following:

**Theorem 2.3.7** (Embedded Resolution of Singularities, Hironaka 1964 [54]). *Let $W$ be a smooth algebraic scheme over a field $K$ of characteristic zero and let $X$ be a subscheme with ideal sheaf $\mathcal{I}_X \subset \mathcal{O}_W$. There exists a sequence*

$$W = W_0 \xleftarrow{\pi_1} W_1 \xleftarrow{\pi_2} \cdots \xleftarrow{\pi_r} W_r$$

*of blow-ups $\pi_i \colon W_i \to W_{i-1}$ at smooth centers $C_{i-1} \subset W_{i-1}$ such that*

1. *The exceptional divisor of the induced morphism $W_i \to W$ has only normal crossings and $C_i$ has normal crossings with it.*
2. *Let $X_i \subset W_i$ be the strict transform of $X$. All centers $C_i$ are disjoint from $\mathrm{Reg}(X) \subset X_i$, the set of points where $X$ is regular or smooth (which are in characteristic zero equivalent).*
3. *$X_r$ is smooth and has normal crossings with the exceptional divisor of the morphism $W_r \to W$.*
4. *The morphism $(W_r, X_r) \to (W, X)$ is equivariant under group actions.*

Embeddedness requires two properties:

- The strict transform $X'$ has to be regular.
- The strict transform $X'$ should meet the exceptional locus $Y'$ in $W'$ transversally.

In general, regularity of the strict transform $X'$ will be reached earlier in the process of resolution. Then transversality requires some additional blow-ups. The following example should illustrate this.

**Example 2.3.8.** Consider the blow-up in the origin in $\mathbb{A}^2$. This desingularizes the curve $X = V(x^2 - y^3)$. We can see in Figure 2.4, Figure 2.5, and Figure 2.6 the process in the real picture of a single chart after several blow-ups. The figures can be read as follows:
In the first picture of Figure 2.4, we see the cusp in blue and the singular point at the origin in green. After blowing up the origin, we can see the exceptional divisor as the green y-axis, which is the preimage of the origin in this chart of the real picture, and the blue component is the strict transform of the cusp.
In Figure 2.5, we see that we have to blow-up a second time in the origin (red point) since the strict transform (blue component) is tangent to the exceptional divisor (green component). After this second blow-up, we get transversality but not normal

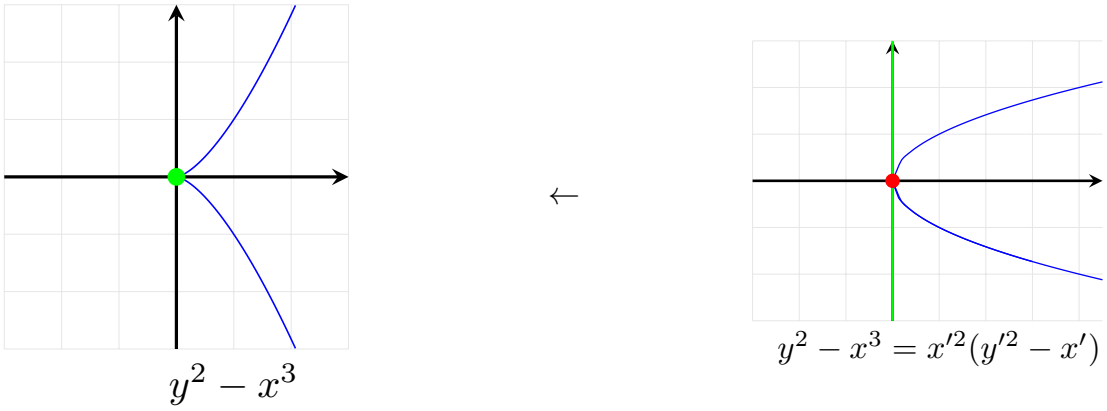



Figure 2.4.: First blow-up of the cusp

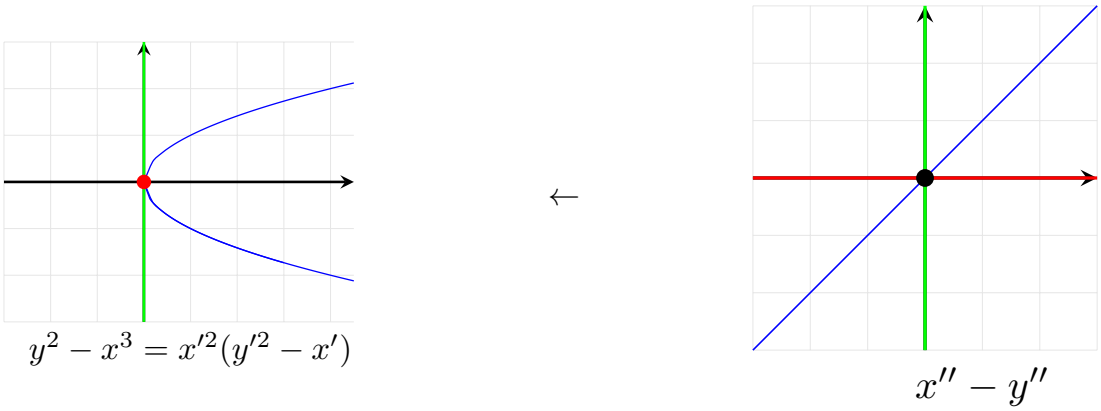



Figure 2.5.: Second blow-up of the cusp

crossings. After a third blow-up at the origin (one chart is to be seen in Figure 2.6), we achieved normal crossings.

Note, that embedded desingularization will also define a non-embedded desingularization of schemes that can be locally embedded in smooth centers. This is not a restriction at all if we consider Noetherian separated schemes of finite type over a field.

More details on how to construct this resolution variant can be found in Sections 3 and 5.2.

**Remark 2.3.9** (How good is a resolution algorithm in the computability sense)**.** In our setting, we are interested in computing an explicit resolution by hand or computer. One famous complexity result is the worst-case analysis of an effective Hironaka-style resolution by [5]. They showed that the complexity of resolution of an ideal on an $m$-dimensional variety is bounded by a function from class $\varepsilon^{m+3}$, where

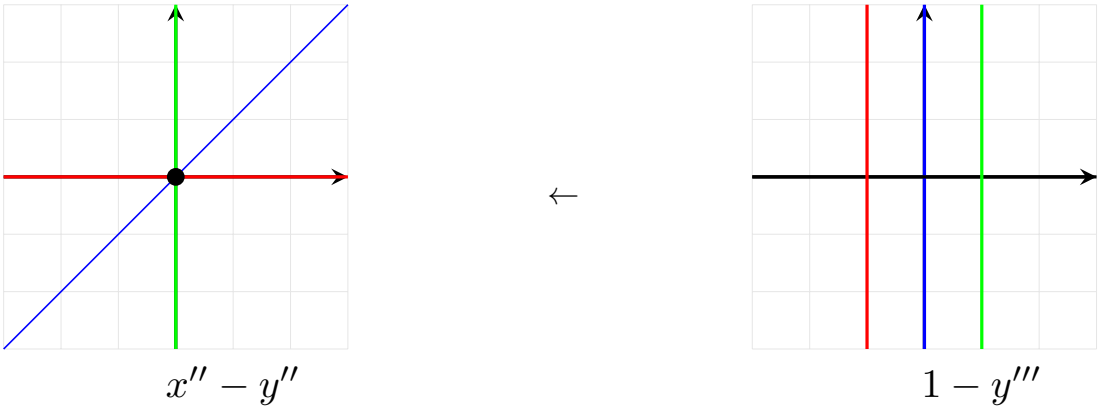



Figure 2.6.: Third blow-up of the cusp

$\varepsilon^\ell$ denotes the $\ell$-th class of the Grzegorczyk-Hierarchy, which consists of integer functions whose construction requires $\ell$ nested primitive recursions. See [47] and Section B.1 for more information about this hierarchy.
That could explain why the existing methods work badly, even on the simplest singularities. Some of these implementations are [13], and [37].
So for practical interests, one can define a good resolution algorithm to be one with a better worst-case or average-case complexity than the known ones.
Hence, some worst-case number of charts analysis of the determinantal algorithms are presented in this thesis.

Closely linked to the resolution of singularities is the principalization of ideals.

**Definition 2.3.10** ([16, Page 2])**.** Let $\mathcal{I}$ be a non zero sheaf of ideals in a non-singular variety $W$. A principalization of $\mathcal{I}$ is a proper birational morphism $\varphi\colon W' \to W$ such that $W'$ is non-singular and $\mathcal{I}\mathcal{O}_{W'}$ is locally principal.

The main goal of principalization of ideals is to get a locally principal ideal sheaf, while the resolution of singularities' goal is to get a non-singular ideal sheaf.

**Remark 2.3.11** (Resolution of singularities implies principalization)**.** Let $\pi\colon X_1 := B\ell_I(X) \to X$ be the blow-up in center $I$ of $X$. Then $\pi^{-1}(I)$ is locally principal (cf. [49, II.7.13]). Hence, if $h\colon X' \to X_1$ is any resolution, then $\pi \circ h\colon X' \to X$ is a principalization of $I$.

This remark allows us to use every resolution algorithm for calculating a principalization of ideals. In our main algorithm in Section 8, we will use the algorithm of Blanco and Encinas (Section 5.4) as such an algorithm for principalization of ideals. Then the entries of our matrix become locally monomial,

but the determinantal singularity is not resolved. So we apply the algorithm of Hu (Section 5.5) for establishing normal crossings. More methods are given in [10, Remark 2.22] or Remark 5.4.39. For more details, see Section 8.

## 2.3.2. Basic objects

The resolution of singularities and the principalization of ideals will be deduced from the resolution of more complicated objects. Such objects appear in various formulations in the literature. Hironaka introduces the notion of idealistic exponents (see Section 2.3.3), Abhyankar used the notion of trio, quartet, and quintet (see [2]), Bierstone and Milman introduced the infinitesimal presentation (see [7]) and in this subsection, we want to present the basic objects by Villamayor (see [77]).

From now on, we want to define these so-called basic objects. A basic object consists of the essential data of a sequence of blow-up in the Hironaka-style resolution. See Section 5. We follow the argumentation of Encinas, and Villamayor [34].

**Definition 2.3.12.** Let $(W, E = \{H_1, ..., H_r\})$, where $W$ is a regular variety (irreducible scheme separated of finite type and smooth over a field $K$), for $1 \leq i \leq r$ $H_i$ is a regular hypersurface in $W$ and $\bigcup_{i=1}^{r} H_i$ has normal crossings. A regular closed subscheme $Y \subset W$ has normal crossings with $E$ if at any $x \in Y$ there is a regular system of parameters $X_1, \ldots, X_n$ such that $I(Y)_x = \langle X_1, \ldots, X_s \rangle$ and for any $H_i$ containing $x$, $I(H_i) = \langle X_{i_j} \rangle$.

In this case, we say that $Y$ is permissible and we define a transformation of $(W, E)$ by blowing up $Y$ and setting $E_1 = \{H'_1, \ldots, H'_r, H_{r+1}\}$, where $H'_i$ is the strict transform of $H_i$, for all $1 \leq i \leq r$ and $H_{r+1} = \pi^{-1}(y)$ is the exceptional hypersurface in $W_1$.

**Remark 2.3.13.** Note, that the property permissible is defined more than once in this thesis, since every strategy of resolution construct a bit different centers, but the main idea is always the same.

**Definition 2.3.14.** A basic object is $(W_0, (J_0, b), E_0 = \{H_1, \ldots, H_r\})$, where $W_0$ is a regular variety, for $i = 1, \ldots, r$ $H_i$ is a regular hypersurface in $W$, $\bigcup_{i=1}^{r} H_i$ has normal crossings, $J \subseteq O_{W_0}$ is an ideal such that $(J_0)_x \neq 0$ for any $x \in W_0$ and $b$ is a positive integer.
We also call $(J_0, b)$ a pair.

**Definition 2.3.15.** The transform $(W_1, (\mathcal{I}_{X_1}, b), E_1)$ of a basic object $(W_0, (\mathcal{I}_{X_0}, b), E_0)$ under a blow-up $\pi\colon (W_1, (\mathcal{I}_{X_1}, b), E_1) \to (W_0, (\mathcal{I}_{X_0}, b), E_0)$ is defined as follows:

- $W_1 = \tilde{W}_0$,
- $\mathcal{I}_{X_1}$ is the ideal of the weak transform of $\mathcal{I}_{X_0}$,
- $b$ does not change,
- $E_1$ is the union of the strict transform of the exceptional divisors from $E_0$ and the new exceptional divisor.

**Definition 2.3.16** ([16, 12.2])**.** A basic object $(W_1, (\mathcal{I}_{X_1}, b), E_1)$ is contained in $(W_1, (\mathcal{I}_{X_2}, c), E_1)$ if $\mathrm{Sing}(\mathcal{I}_{X_1}, b) \subset \mathrm{Sing}(\mathcal{I}_{X_2}, c)$.

**Example 2.3.17** ([16, Example 12.3])**.** Let $\mathcal{I}_{X_1} \subset \mathcal{I}_{X_2}$ be two sheaves of ideals. Then

$$(W, (\mathcal{I}_{X_1}, b), E) \subset (W, (\mathcal{I}_{X_2}, b), E).$$

**Definition 2.3.18.** [16, Remark 12.4] Two basic objects $(W, (\mathcal{I}_{X_1}, b_1), E)$ and $(W, (\mathcal{I}_{X_2}, b_2), E)$ are called equivalent if $(W, (\mathcal{I}_{X_1}, b_1), E)$ is contained in $(W, (\mathcal{I}_{X_2}, b_2), E)$ and vice versa. We will use the notation

$$(W, (\mathcal{I}_{X_1}, b_1), E) \cong (W, (\mathcal{I}_{X_2}, b_2), E).$$

**Example 2.3.19** ([16, Example 12.6])**.** Let $(\mathcal{I}_{X_2}, b_2) := (\mathcal{I}_{X_1}^2, 2b)$ then

$$(W, (\mathcal{I}_{X_1}, b), E) \cong (W, (\mathcal{I}_{X_1}^2, 2b), E).$$

**Definition 2.3.20** ([16, 12.7])**.** Let $(W_1, (\mathcal{I}_{X_1}, b), E_1)$ and $(W_1, (\mathcal{I}_{X_2}, c), E_1)$ be two basic objects. The intersection $(W_1, (\mathcal{I}_{X_1}, b), E_1) \cap (W_1, (\mathcal{I}_{X_2}, c), E_1)$ is defined as the basic object $(W_1, (\mathcal{I}_{X_3}, d), E_1)$, where $\mathcal{I}_{X_3} = \mathcal{I}_{X_1}^c + \mathcal{I}_{X_2}^b$ and $d = bc$.

**Definition 2.3.21** ([26, Definition 6.25])**.** Let $(W_0, E_0)$ be like in Definition 2.3.12. We define a general basic object $(\mathcal{F}_0, W_0, E_0)$ on $(W_0, E_0)$ to be a collection of $(W_i, E_i)$ (like in Definition 2.3.12) with closed sets $F_i \subset W_i$ which have been constructed inductively to satisfy the following properties:

1. $F_0 \subset W_0$
2. Assume $F_i \subset W_i$ is defined for $(W_i, E_i)$. Let $\pi_{i+1}\colon (W_{i+1}, E_{i+1}) \to (W_i, E_i)$ be a restriction, then $F_{i+1} := \pi_{i+1}^{-1}(F_i)$.

3. Assume $F_i \subset W_i$ is defined for $(W_i, E_i)$. Let $\pi_{i+1}\colon (W_{i+1}, E_{i+1}) \to (W_i, E_i)$ be a transformation centered at $Y_i \subset F_i$, then $F_{i+1} \subset W_{i+1}$ is a closed set such that $F_i \backslash Y_i \cong F_{i+1} \backslash \pi_{i+1}(Y_i)$ by the map $\pi_{i+1}$.

These sets $F_i$ will be called the closed sets associated to $(\mathcal{F}_0, W_0, E_0)$.

## 2.3.3. Idealistic exponents

Another key idea similar to basic objects in Hironaka's original proof of resolution of singularities in charateristic zero is the idealistic exponent, which we want to discuss in the following.

The main idea is to encode the local resolution data of a scheme $X \subset Z$ at a point $x \in X$ in a pair $\mathbb{E} = (J, b)$ on the regular ring $R = \mathcal{O}_{Z,x}$, where $J \subset R$ is the ideal defining $X$ at $x$ locally and $b \in \mathbb{Q}_+$ is a positive rational number.

We use the notation of idealistic exponents introduced by Hironaka in [56],[58] on regular schemes of finite type over a perfect field. Later he extended this notion to excellent regular noetherian schemes, which are not necessarily of finite type, see [59]. See [71] for a more modern presentation.
Let $Z$ be a regular irreducible scheme of finite type over an arbitrary ring $K$.

**Definition 2.3.22** ([16, Definition 3.1])**.** A pair $\mathbb{E} = (J, b)$ on $Z$ is a pair consisting of an ideal sheaf $J \subset \mathcal{O}_Z$ and a positive integer $b \in \mathbb{Z}_+$.

The main idea is to encode the local resolution data of a scheme $X \subset Z$ at a point $x \in X$ in a pair $\mathbb{E} = (J, b)$ on the regular ring $R = \mathcal{O}_{Z,x}$, where $J \subset R$ is the ideal defining $X$ at $x$ locally and $b \in \mathbb{Q}_+$ is a positive rational number.

**Definition 2.3.23.** Let $\mathbb{E} = (J, b)$ be a pair on a regular local ring $R$ and let $(\underline{t}) = (t_1, \ldots, t_a)$ be an arbitrary finite system of independent indeterminates. Then the lift of $\mathbb{E}$ to $R[\underline{t}]$ is defined as the pair $\mathbb{E}[\underline{t}] = (J \cdot R[\underline{t}], b)$.

**Definition 2.3.24** ([71, Definition 1.2])**.** Let $X \subset Z$ and $J \subset R$ the stalk of $I(X)$ at $x$. Furthermore let $\mathfrak{m}_x$ be the maximal ideal in the local ring at $x$. Then

$$\operatorname{ord}_x(J) := \sup\{d \in \mathbb{Z}_0 \cup \{\infty\} \mid J_x \subseteq \mathfrak{m}_x^d\}$$

is called the order of $J$ at $x$.
The order of an idealistic exponent $\mathbb{E} = (J, b)$ at a not necessarily closed point $x \in Z$ is defined as

$$\operatorname{ord}_x(\mathbb{E}) = \begin{cases} \frac{\operatorname{ord}_x(J)}{b}, & \text{if } \operatorname{ord}_x(J) \geq b \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

**Definition 2.3.25** ([71, Definition 1.2])**.** Let $\mathbb{E} = (J, b)$ be an idealistic exponent. We define the singular locus of $\mathbb{E}$ as

$$\mathrm{Sing}(\mathbb{E}) = \{x \in Z \mid \mathrm{ord}_x(\mathbb{E}) \geq 1\}.$$

**Remark 2.3.26.** Note, that $\mathrm{Sing}(\mathbb{E})$ coincides with $\mathrm{Sing}(J, b)$ in the notation of, e.g., [16]. Since we define the order function on $\mathbb{E}$ with a division by $b$, we have in this case $\geq 1$ in our definition of $\mathrm{Sing}(\mathbb{E})$ and not like in the definition of [16] $\geq b$.

The following example should illustrate the role of the value of $b$ in the definition of the singular locus of $\mathbb{E}$:

**Example 2.3.27.**

- Let $\mathbb{E}_1 = (\langle y^2 + x^4 \rangle, 2)$ and let $\mathbb{E}_2 = (\langle y^2 + x^4 \rangle, 3)$. Then $\mathrm{Sing}(\mathbb{E}_1) = V(x, y)$ but $\mathrm{Sing}(\mathbb{E}_2) = \emptyset$.
- Let $\mathbb{E}_1 = (\langle x^3 - y^3 z^2 \rangle, 2)$ and let $\mathbb{E}_2 = (\langle x^3 - y^3 z^2 \rangle, 3)$. Then $\mathrm{Sing}(\mathbb{E}_1) = V(x, y) \cup V(x, z)$ and $\mathrm{Sing}(\mathbb{E}_2) = V(x, y)$.

**Definition 2.3.28** ([71, Definition 1.5.1])**.** Let $\mathbb{E} = (J, b)$ be a pair on $Z$. A blow-up $\pi \colon Z' \to Z$ with center $D$ is called permissible for $\mathbb{E}$, if $D \subset \mathrm{Sing}(\mathbb{E})$ and $D$ is regular, i.e, $D$ is a permissible center for $(E)$.

**Remark 2.3.29.** Let $\mathbb{E}_1$ and $\mathbb{E}_2$ be pairs on $R$. By definition

- $\mathrm{Sing}(\mathbb{E}_1 \cap \mathbb{E}_2) = \mathrm{Sing}(\mathbb{E}_1) \cap \mathrm{Sing}(\mathbb{E}_2)$ and
- $\mathrm{ord}_x(\mathbb{E}_1 \cap \mathbb{E}_2) = \min\{\mathrm{ord}_x(\mathbb{E}_1), \mathrm{ord}_x(\mathbb{E}_2)\}$, for $x \in \mathrm{Sing}(\mathbb{E}_1 \cap \mathbb{E}_2)$.

In older literature the order is also defined as $\widetilde{\mathrm{ord}}_x(\mathbb{E}) = \frac{\mathrm{ord}_x(J)}{b}$ (see, e.g., [56, Section 1, Remark 5]). The difference to our definition is that for some $x \notin \mathrm{Sing}(\mathbb{E})$ the equation $\mathrm{ord}_x(\mathbb{E}) = 0$ holds, but $0 \leq \widetilde{\mathrm{ord}}_x(\mathbb{E}) \leq 1$ is possible.

**Definition 2.3.30** ([71, Definition 1.7])**.** Two pairs $\mathbb{E}_1 = (J_1, b_1)$ and $\mathbb{E}_2 = (J_2, b_2)$ on $Z$ are equivalent, if the following holds:
Let $(\underline{t}) = (t_1, \ldots, t_a)$ be an arbitrary finite system of independent indeterminates. Then any sequence of local blow-ups over $R[\underline{t}]$ which is permissible for $\mathbb{E}_1[\underline{t}]$ is also permissible for $\mathbb{E}_2[\underline{t}]$ and vice versa.
We write $\mathbb{E}_1[\underline{t}] \sim \mathbb{E}_2[\underline{t}]$.
An idealistic exponent $\mathbb{E}_\sim$ is the equivalence class of a pair $\mathbb{E}$.

If it is clear from the context, we abuse notation and use $\mathbb{E}$ instead of $\mathbb{E}_\sim$.

**Example 2.3.31** ([70, Remark 1.1.7.]). Let $\mathbb{E} = (J, b)$ such that $\mathrm{Sing}(\mathbb{E}) = \emptyset$. Let $x$ be a fixed point on $Z$. Let $\mathbb{E}_1$ and $\mathbb{E}_2$ be in the same idealistic exponent as $\mathbb{E}$. Then $\widetilde{\mathrm{ord}_x(\mathbb{E}_i)} \in [0,1) \cap \mathbb{Q}$, for $i = 1, 2$ but the values do not have to coincide, where $\mathrm{ord}_x(\mathbb{E}_i) = 0$, for $i = 1, 2$.
Let $\mathbb{E}_1 = (\langle y + z \rangle, 2)$ and $\mathbb{E}_2 = (\langle y + z \rangle, 3)$ and $x = V(y, x)$, then

$$\widetilde{\mathrm{ord}_x(\mathbb{E}_1)} = \frac{1}{2} \neq \frac{1}{3} = \widetilde{\mathrm{ord}_x(\mathbb{E}_2)}$$

but

$$\mathrm{ord}_x(\mathbb{E}_1) = \mathrm{ord}_x(\mathbb{E}_2) = 0.$$

**Definition 2.3.32** ([71, Definition 1.5.2]). Let $\mathbb{E} = (J, b)$ be a pair on $R$, let $\pi$ be a permissible blow-up for $\mathbb{E}$ and $U' = \mathrm{Spec}(R') \subset Z'$ an affine chart. We define the controlled transform of $\mathbb{E}$ in $U'$ as $\mathbb{E}' = (J', b)$, where $J'$ is defined via the equation

$$J \cdot R' = J' H^b,$$

where $H$ denotes the exceptional divisor in $U'$ and $b$ is the control.

**Remark 2.3.33.** Let $d \in \mathbb{N}_0$ and let $R$ be a regular ring. Then $\mathrm{Diff}_{\mathbb{Z}}^{\leq m}(R)$ is the set of differential operators of order at most $m$ of $R$.

**Lemma 2.3.34** ([71, Facts 1.11]). *Let $R$ be a regular local ring, let $\mathbb{E}_1 = (J, b) = (J_1, b_1)$ and $\mathbb{E}_2 = (J_2, b_2)$ be pairs on $R$.*

1. *For every $a \in \mathbb{Z}_+$, $(J^a, ab) \sim (J, b)$ holds. Furthermore, if $b_2 = b_1 = b$ then $(J_1, b) \cap (J_2, b) = (J_1 + J_2, b)$ holds.*

2. *If $\mathrm{Sing}(J_1, b_1 + 1) = \mathrm{Sing}(J_2, b_2 + 1) = \emptyset$, then we have $(J_1, b_1) \cap (J_2, b_2) \sim (J_1 J_2, b_1 + b_2)$.*

3. *Let $\mathcal{D} \in \mathrm{Diff}_{\mathbb{R}}^{\leq m}(Z)$ be a differential operator of order $m \in \mathbb{Z}_{\geq 0}, m < b$ on $R$ such that $\mathcal{D}(R) \subset R$. Then $(J, b) \sim (\mathcal{D}J, b - m) \cap (J, b)$.*

*Proof.* The proof is straight forward and we are guided by the presentation of [16].

1. Let $\mathbb{E}_1 = (J, b), \mathbb{E}_2 = (J^a, ab)$ and $x \in Z$ arbitrary. Then

$$\frac{\mathrm{ord}_x(J^a)}{ab} = \frac{a \cdot \mathrm{ord}_x(J)}{ab} = \frac{\mathrm{ord}_x(J)}{b}$$

holds. This implies $\mathrm{ord}_x(\mathbb{E}_1) = \mathrm{ord}_x(\mathbb{E}_2)$. This argumentation is stable under the change from $(\mathbb{E}_i, Z)$ to $(\mathbb{E}_i[\underline{t}], Z[\underline{t}])$ for a finite system of indeterminates $(\underline{t})$.

Furthermore it is stable under suitable blow-ups $\pi\colon Z' \to Z$: The transform $\mathbb{E}_1' = (J', b')$ of $\mathbb{E}_1 = (J, b)$ under the blow-up $\pi$ is defined via $J\mathcal{O}_{Z'} = H^b J'$, where $H$ is the sheaf of the exceptional divisor under $\pi$.

$$J^a\mathcal{O}_{Z'} = (J\mathcal{O}_{Z'})^a = (H^b J')^a = H^{ab}J'^a$$

implies that the transform of $\mathbb{E}_2$ is $\mathbb{E}_2' = (J'^a, ab)$. So $\operatorname{ord}_{x'}(\mathbb{E}_1') = \operatorname{ord}_{x'}(\mathbb{E}_2')$ holds for every $x' \in Z'$. Hence, every local sequence of regular blow-ups which are suitable for $\mathbb{E}_1$ is also suitable for $\mathbb{E}_2$ and vice versa.

2. With an analog argumentation as in (1) and the use of

$$\operatorname{ord}_x(J_1J_2, b_1 + b_2) = \frac{b_1}{b_1 + b_2}\operatorname{ord}_x(J_1, b_1) + \frac{b_2}{b_1 + b_2}\operatorname{ord}_x(J_2, b_2)$$

we get $(J_1, b_1) \cap (J_2, b_2) \subset (J_1J_2, b_1 + b_2)$.
For the other direction, $\operatorname{Sing}(J_i, b_i + 1) = \emptyset$, for $i = 1, 2$ implies

$$\operatorname{ord}_x(J_i, b_i + 1) = \frac{\operatorname{ord}_x(J_i)}{b_i + 1} < 1,$$

for all $i = 1, 2$ and for all $x \in Z$. Suppose there is no local sequence of regular blow-ups which is regular for $(J_1J_2, b_1 + b_2)$ and not regular for $\mathbb{E}_1 \cap \mathbb{E}_2$. We may assume that this sequence is generated by a single blow-up $\pi\colon Z' \to Z$ with center $D$ and $D$ is not a suitable center for $\mathbb{E}_1 \cap \mathbb{E}_2$. Without loss of generality, we assume $D$ is not permissible for $\mathbb{E}_1$. This implies that there is a $y \in D$ such that

$$\operatorname{ord}_y(\mathbb{E}_1) = \frac{\operatorname{ord}_y(J_1)}{b_1} =: \frac{m_1}{b_1} < 1$$

but $\operatorname{ord}_y(J_1J_2, b_1 + b_2) \geq 1$. We write $\operatorname{ord}_y(\mathbb{E}_2) = \frac{m_2}{b_2}$. We get

$$\operatorname{ord}_y(J_1J_2, b_1 + b_2) = \frac{m_1}{b_1 + b_2} + \frac{m_2}{b_1 + b_2} < \frac{b_1}{b_1 + b_2} + \frac{m_2}{b_1 + b_2} = 1\frac{m_2 - b_1}{b_1 + b_2}.$$

Since $\operatorname{ord}_y(J_1J_2, b_1 + b_2) \geq 1$ we know $\frac{m_2 - b_1}{b_1 + b_2} \geq 0$. This yields

$$\operatorname{ord}_y(J_2, b_2 + 1) = \frac{m_2}{b_2 + 1} \geq 1$$

which contradicts $\operatorname{ord}_x(J_1J_2, b_1 + b_2) < 1$.

3. See [59, Theorem 3.4] or [70, Proposition 1.1.13] for a more modern presentation. □

## 2.4. Different measures of being singular

In different resolution strategies, it is often the idea to blow-up the locus of the most singular points. We first give an overview of the properties of such a measure, followed by some famous examples we need later in this thesis.

**Definition 2.4.1** ([9, Definition 2.1])**.** Let $g\colon X \to (A, \leq)$ be a function on a topological noetherian space $X$, where $A$ is a totally ordered set. We call $g$ upper semi-continuous, if

- $\mathrm{Im}(g) = \{a_1, \ldots, a_s\}$ is a finite subset of $A$ and
- the sets $\mathcal{F}_{a_i} = \{\xi \in X \mid g(\xi) \geq a_i\}$ are closed for all $i = 1, \ldots, s$.

**Lemma 2.4.2** ([20, Lemma 2.34(a)])**.** *A map $g\colon X \to (A, \leq)$ is upper semi-continuous if and only if*

1. *if $x, y \in X$ such that $x \in \overline{\{y\}}$, then $g(x) \leq g(y)$ and*
2. *for all $y \in X$ there is a dense open subset $U \subset \overline{\{y\}}$ such that $g(x) = g(y)$ for all $x \in U$.*

With this property and assuming that the invariant $g$ cannot increase under suitable blow-ups and that $g$ decreases only finitely many times until the strict transform of $X$ is regular, we can construct a finite sequence of blow-ups such that the minimal value achieved by the invariant of the resolution decreases strictly.

Then this implies an embedded resolution of singularities for $X \subset Z$. So the question is: Given a reduced scheme $X \subset Z$, can we find a finite sequence of blow-ups.

$$\begin{array}{ccccccc} Z =: Z_0 & \leftarrow B\ell_{D_0}(Z_0) =: & Z_1 & \leftarrow \cdots \leftarrow & B\ell_{D_{m_0-1}}(Z_{m_0-1}) =: & Z_{m_0} \\ \cup & & \cup & & & \cup \\ X = X_0 & \leftarrow & X_1 & \leftarrow \cdots \leftarrow & & X_{m_0} \end{array}$$

**Example 2.4.3.** Some examples of upper semi-continuous functions are

- The order function, see [54, Chapter III §3 Corollary 1 p.220], Definition 2.4.8 and Section 2.4.2,
- The Hilbert-Samuel function, see [20, Definition 2.28] or [7].
- The $E$-order function, see Definition 5.4.5 and Lemma 5.4.16.

- The multiplicity, see Section 2.4.1 and [26, Definition A.17].

Since we focused on the computation of a resolution of singularities, we do not discuss the Hilbert-Samuel function here. If we want to calculate the locus where the Hilbert-Samuel function is maximal, we would have to compute a stratification by the initial forms of the local standard bases (Definition 5.1.1) in all points seperately. This is not easy and efficient to calculate.

For resolution of singularities, it is standard to use upper semi-continuous functions as an invariant of the singularity. This stems from the fact that the level sets of an upper semi-continuous function are Zariski-closed, such that especially the singular loci are Zariski-closed.

We want to consider subsets of the singular locus in this thesis. Different approaches use different notions of this locus. We want to present them now.

**Remark 2.4.4.**

- In Section 5.4, we will consider the $E$-singular locus of a (binomial) ideal $J$ with respect to $c$ is (Definition 5.4.10)

$$E\text{-}\operatorname{Sing}(J,c) = \{\xi \in W \mid E\text{-}\operatorname{ord}_\xi(J) \geq c\}.$$

- Let $\mathbb{E} = (J, b)$ be an idealistic exponent. We define the singular locus of $\mathbb{E}$ as (Definition 2.3.25)

$$\operatorname{Sing}(\mathbb{E}) = \{x \in Z \mid \operatorname{ord}_x(\mathbb{E}) \geq 1\}.$$

  This is considered in Section 8.1.

- Let $t \in \mathbb{N}$. We define the $t$-singular locus of a surface $S \subset V$ (Definition 3.2.2) as the set

$$\operatorname{Sing}_t(S) := \{p \in V \mid \nu_p(S) \geq t\},$$

  where $\nu$ denotes the mulitplicity. We will use this locus in Chapter 3.

Note, that these constructions are quite analogous. Only the measure differs and the $\geq t$ resp. $\geq c$ term is a priori given by the order of an idealistic exponent.

## 2.4.1. Multiplicity

Since we need the definition of the multiplicity in the case of curve singularities, we define them first in this setting.

**Definition 2.4.5** ([49, 5.3])**.** Let K be an algebraically closed field of characteristic 0 and let $K[[x, y]]$ be its power series ring in two variables $x$ and $y$. Let

$$f = \sum_{i,j} a_{i,j} x^i y^j \in K[[x, y]]$$

be a non-unit such that $x$ does not divide $f$. We define the multiplicity of $f$ as

$$\text{mult}(f) := \min\{i + j \mid a_{i,j} \neq 0\}$$

and

$$\text{mult}(f(0, y)) := \min\{j \mid a_{0,j} \neq 0\}.$$

**Remark 2.4.6.** If $I = \langle f \rangle$ is a principal ideal, then the order of $I$ at a point $x$ (Definition 2.4.8) equals the multiplicity of the hypersurface $V(f)$ at $x$.

In the general setting the multiplicity has the following definition.

**Definition 2.4.7** ([26, Definition A.17])**.** Let $q$ be a point on a variety $W$ and let $J \subset \mathcal{O}_W$ be an ideal sheaf. The multiplicity of $J$ at $q$ is defined as

$$\nu_q(J) = \text{ord}_{\mathcal{O}_{W,q}}(J\mathcal{O}_{W,q}).$$

## 2.4.2. Locus of maximal order

In this subsection, we give a brief overview of the basic ideas of the maximal order and its locus which are influenced by [35].
Furthermore, we will use a refinement of the usual order in Section 5.6. The implementation of the calculation is described in Section A.1.

In this subsection, we assume that $X$ is a reduced excellent Noetherian scheme (of dimension two), embedded in some excellent regular scheme $Z$.

**Definition 2.4.8** ([35, Definition 2.2])**.** Let $X$ be a noetherian excellent scheme and let $x \in X$ be a point. Let $(R = \mathcal{O}_{Z,x}, \mathfrak{m}, k = R/\mathfrak{m})$ be the local ring of $Z$ at $x$ and let $I_X \subset R$ be the ideal which defines $X$ locally at $x$. We define

1. the order of $X$ in $x$ as the order of $I_X$ in $\mathfrak{m}$:

$$\text{ord}_x(X) := \text{ord}_{\mathfrak{m}}(I_X) := \sup\{t \in \mathbb{N} \mid I_X \subset \mathfrak{m}^t\},$$

2. the maximal order of $X$

$$\text{max-ord}(X) := \sup\{\text{ord}_x(X) \mid x \in X\},$$

3. the locus of maximal order of $X$

$$\text{Max-ord}(X) := \{x \in X \mid \text{ord}_x(X) = \text{max-ord}_x(X)\} \text{ and}$$

4. the initial form of $f$ with respect to $\mathfrak{m}$

$$\text{in}_{\mathfrak{m}}(f) := f \mod \mathfrak{m}^{d(f)+1} \in \text{gr}_{\mathfrak{m}}(R),$$

where $\text{gr}_{\mathfrak{m}}(R) = \bigoplus_{t\geq 0} \mathfrak{m}^t/\mathfrak{m}^{t+1}$ denotes the associated graded ring of $R$ at $\mathfrak{m}$ and $d(f) := \text{ord}_{\mathfrak{m}}(f)$,

5. the initial ideal of $I_X$ at $\mathfrak{m}$ as the ideal $\text{In}_{\mathfrak{m}}(I_X)$ in $\text{gr}_{\mathfrak{m}}(R)$ generated by the initial forms of the elements in $I_X$

$$\text{In}_{\mathfrak{m}}(I_X) := \langle \text{in}_{\mathfrak{m}}(f) \mid f \in I_X \rangle.$$

Note, that in contrast to Definition 2.3.24, we do not divide by a number $b$, here. So the definition of the order of an idealistic exponent and the definition of the order of a scheme differs in this detail.

**Definition 2.4.9** ([36, Definition 54])**.** In characteristic 0, we define $\Delta(\mathcal{I}_X) \subset \mathcal{O}_W$ for a basic object $(W, (\mathcal{I}_X, c), E)$ as the sheaf of ideals locally generated by

$$\Delta(\mathcal{I}_X) := \{g_i \mid 1 \leq i \leq s\} \cup \{\frac{\partial g_i}{\partial x_j} \mid 1 \leq i \leq s, 1 \leq j \leq d\},$$

where $x_1, \ldots, x_d$ is a regular system of parameters for $\mathcal{O}_{W,w}$ and $g_1, \ldots, g_s$ are a set of generators for $I_X$.
Moreover, we define $\Delta^i(\mathcal{I}_X)$ inductively as $\Delta(\Delta^{i-1}(\mathcal{I}_X))$.

**Remark 2.4.10** ([16, Page 33])**.** In characteristic 0, the order of $J$ is $c$ if and only if the order of $\Delta^{c-1}(J)$ is 1. An element of order 1 in $\Delta^{c-1}(J)$ defines a smooth hypersurface.
The locus of order at least $c$ of $J$ coincides with $V(\Delta^{c-1}(J))$.

**Remark 2.4.11** ([16, 13.7])**.** Assuming we are in characteristic 0. Let $J \subset \mathcal{O}_W$ be a non zero sheaf of ideals at any irreducible component of the smooth scheme $W$ then $\Delta(J)$ satisfies the following properties:

1. $J \subset \Delta(J) \subset \Delta^b(J) = \mathcal{O}_W$ for some $b \in \mathbb{N}$.

2. For any point $x \in W$ the equation $\text{ord}_{\mathcal{O}_{W,x}}(J_x) = b > 1$ holds if and only if $\text{ord}_{\mathcal{O}_{W,x}}(\Delta(J)_x) = b - 1$.

3. For any point $x \in W$ the in equality $\mathrm{ord}_{\mathcal{O}_{W,x}}(J) \geq b > 1$ holds if and only if $x \in V(\Delta^{b-1}(J))$.

Now we give an illustration of the idea of the refined order.

Since we will use this refined order only for the Cossart, Jannsen and Saito algorithm, we will assume that $X$ be a reduced excellent Noetherian scheme of dimension 2 embedded in some excellent regular scheme $Z$, to the end of the section.

**Remark 2.4.12** ([35, Observation 2.3])**.** We denote by $(R = \mathcal{O}_{Z,x}, \mathfrak{m}, k = R/\mathfrak{m})$ the local ring of $Z$ at $x$ which is excellent and regular. Let $\mathfrak{M} := \mathrm{In}_{\mathfrak{m}}(\mathfrak{m}) \in \mathrm{gr}_{\mathfrak{m}}(R)$. Set $I_1 = \mathrm{In}_{\mathfrak{m}}(I_X)$. If $\mathrm{ord}_{\mathfrak{M}}(I_1) = 1$, we can consider the image $\overline{I_1}$ in the degree 1 slice of $\mathrm{gr}_{\mathfrak{m}}(R)$. This yields a subspace of the finite dimensional $k$-vector space $\mathrm{gr}_{\mathfrak{m}}(R)_1 = \mathfrak{m}/\mathfrak{m}^2$.
We can find a basis $F_1, \ldots, F_a$ of $\overline{I_1}$ for some $a := a_x \in \mathbb{N}$. Either $I_1 = \langle F_1, \ldots, F_a \rangle_{\mathrm{gr}_{\mathfrak{m}}(R)}$ or $\mathrm{ord}_{\mathfrak{M}}(H) > 1$ for all elements $H$ in $I_{a+1}$ which denotes the ideal generated by the set $I_1/\langle F_1, \ldots, F_a \rangle_{\mathrm{gr}_{\mathfrak{m}}(R)}$.
We define

$$d_x := \begin{cases} 1, & \text{if } I_1 = \langle F_1, \ldots, F_a \rangle \\ \mathrm{ord}_{\mathfrak{M}}(I_{a+1}) > 1, & \text{otherwise.} \end{cases}$$

For each $F_i \in \mathrm{In}_{\mathfrak{m}}(I_X)$, we can choose a lift $f_i \in I_X$ such that $\mathrm{in}_{\mathfrak{m}}(f_i) = F_i$ for all $1 \leq i \leq a$. We have $\mathrm{ord}_{\mathfrak{m}}(f_i) = \mathrm{ord}_{\mathfrak{M}}(F_i) = 1$ and $(f_1, \ldots, f_a)$ forms a regular sequence and in particular $f_i \notin \langle f_1, \ldots, f_{i-1} \rangle$, for all $1 \leq i \leq a$.
$\mathcal{Y} := V(f_1, \ldots, f_a)$ defines a $(n-a)$-dimensional regular subscheme of $\mathcal{Z} = \mathrm{Spec}(R)$ which contains $\mathcal{X} = V(I_X)$. This implies

$$a_x \leq n_x - \dim \mathcal{X} \leq N = \dim Z.$$

The following are equivalent:

1. $a_x < n_x - \dim \mathcal{X}$
2. $d_x > 1$
3. $X$ is singular at $x$.

The value of $n_x - a_x$ coincides with the embedding dimension of $X$ at $x$ at closed point corresponding to a maximal ideal $\mathfrak{m}_x$ .

Remark 2.4.12 is precisely the beginning of a possible way to construct a standard basis for $I_X$ in the Hironaka sense and thus to determine the Hironaka invariant

$\nu^*(I_X, R)$. See Section 5.2 for more information about this invariant.
We set $I := \mathrm{In}_{\mathfrak{m}}(I_X)$ then

$$\nu^*(I_X, R) = (\nu^1(I), \nu^2(I), \ldots, \nu^a(I), \nu^{a+1}(I), \ldots) = (1, \ldots, 1, d_x, \nu^{a+2}(I), \ldots) \in (\mathbb{N} \cup \{\infty\})^{\mathbb{N}}.$$

The invariant in Section 5.6 max-$\nu$ at a closed point $x$ corresponding to a maximal ideal of $I_X$ is $(N - a_x, d_x)$. Hence, our invariant coincides with the truncation of the $\nu^*$-invariant after the first entry with value $> 1$.

**Definition 2.4.13** ([35, Definition 2.4])**.** Let $x, a_x$ and $N$ as in Remark 2.4.12.

1. $\nu_{\mathrm{ref}} := \nu_{\mathrm{ref\ X,Z}} : X \to (\mathbb{N}^2, \leq_{lex})$, $\nu_{\mathrm{ref}}(x) := (N - a_x, d_x)$ is called refined order of $X$.

2. Let $A$ in $\mathbb{N}^2$. We set $\mathcal{V}_{\geq A}(X) := \{x \in X \mid \nu_{\mathrm{ref}}(x) \geq A\}$. The maximal refined order is
$$\text{max-}\nu(X) := (\alpha, \delta) := \max\{\nu_{ref}(x) \mid x \in X\}.$$
The locus of maximal refined order is
$$\text{Max-}\nu(X) := \begin{cases} \mathcal{V}_{\geq \text{max-}\nu(X)}(X), & \text{if } \delta > 1, \\ X, & \text{if } \delta = 1. \end{cases}$$

**Remark 2.4.14.** If $X = V(f)$ is an affine hypersurface, then $\nu_{\mathrm{ref}}(X) = \mathrm{ord}(X)$.
If $\dim(X) = 2$ and $X$ is reduced, then Max-$\nu(X)$ has at most dimension one. In particular, Max-$\nu(X)$ itself has at most isolated singularities.

In [35, Proposition 2.8] shows that max-$\nu(\cdot)$ is upper semi-continous. Hence, in particular Max-$\nu(\cdot)$ is closed.

Moreover, [35, Proposition 2.8] shows that the refinement of the order function is more suitable than the usual order function, since it distinguishes regular from singular points, while the order function does not. The following example illustrates the problem.

**Example 2.4.15** ([35, Example 2.7])**.** Let $R = K[t, v, w, y, z]_{\mathfrak{m}}$, where $\mathfrak{m} = \langle t, v, w, y, z\rangle$. We consider the scheme $X := V(J) \subset \mathrm{Spec}(R)$, where $J := \langle t, v^2 - y^3, z^5 - y^2w^5\rangle$.
Then $\mathrm{ord}_{\mathfrak{m}}(J) = 1$, although the point in $X$ that corresponds to $\mathfrak{m}$ is singular.
The refined order is $\nu_{\mathrm{ref}} = (3, 2)$, since $\mathrm{ord}_{\mathfrak{m}}(v^2 - y^3) = 2$ is maximal.

More information about the explicit calculation is found in Section A.1.

# 3. Easiest cases of resolution of singularities

In this chapter we will present an introduction to the easiest cases in resolution of singularities. We give a brief overview about the most intuitive results in algorithmic resolution of singularities, namely the case of curve singularities and the case of surface singularities in characteristic zero. This chapter presents an intuition that is fundamental for the understanding of the remaining thesis.

The central point of resolution of singularities by a sequence of blow-ups is the appropriate choice of the centers. For this reason, desingularization algorithms are often stated as algorithms for the choice of centers. The choice is controlled by assigning an invariant which decreases (strictly) after blow-up in a suitable center.

We discuss the resolution of curve singularities for arbitrary fields and surface singularities in the characteristic zero setting (see Section 3.1 and Section 3.2). Chapter 5 discusses the background of the algorithmic approaches in more complex singularities.

## 3.1. Resolution of curve singularities in arbitrary characteristic

Resolution of curve singularities was investigated in the 19th century. The book of Cutkosky gives a good summary of the results. For that reason we follow the argumentation of his book [26, Chapter 3.1].

Let $K$ be an algebraic closed field of arbitrary characteristic. In this section we assume $C \subseteq \mathbb{A}^2_K$ to be a curve. On the algebraic side of view, $K[x,y]$ is a unique factorization domain. So we know that there exists some $f \in K[x,y]$ such that $V(f) = C$.

Let $q \in C$ be a closed point. In this chapter $\nu_q(C)$ denotes the multiplicity of $C$ at $q$. Note however that $\nu_q(C) = \mathrm{ord}_q(C)$, since $C$ is a hypersurface singularity and

$q$ is a closed point on $C$ (Remark 2.4.6). So one can interpret $\nu_q(C)$ as multiplicity or as order.

The following Lemma provides a criterion for regularity of points on plane curves.

**Lemma 3.1.1** ([26, Lemma 3.2])**.** *The point $q \in \mathbb{A}^2_K$ is a regular point of $C$ if and only if $\nu_q(C) = 1$.*

*Proof.* See [26, proof of Lemma 3.2]. □

**Remark 3.1.2.** Lemma 3.1.1 is also true if $q$ is a closed point on a regular surface $S$ over a field $K$ and $C$ is a curve contained in $S$. So we can apply it in Section 3.1.1, too.
The proof only has to be modified by replacing $R$ with the regular local ring $\mathcal{O}_{S,q}$ which has parameters $(\underline{x}, \underline{y})$ which are a $K(q)$ basis of $\mathfrak{m}_q/\mathfrak{m}_q^2$. Since $R$ is a unique factorization domain there exists some $f \in R$ such that $C = V(f)$ at $q$.
For more details see [26, Remark 3.3].

## 3.1.1. Blow-up points on a regular surface

In this subsection, we want to focus on blow-ups of points on regular surfaces. Later we discuss the resolution of curve singularities in the embedded case. We will see that our curve $C$ can be embedded into a regular surface and that we only need to blow-up singular points for the resolution of $C$.

**Lemma 3.1.3** ([26, Lemma 3.10])**.** *Let $X$ be a regular surface over an algebraically closed field $K$, and let $C$ be a curve on $X$. Furthermore, let $p \in X$ such that $\nu_p(C) = r$ and let $\pi\colon B\ell_p(X) \to X$ the blow-up in center $p$. We denote by $C'$ the strict transform of $C$ under $\pi$ and we assume that $q \in \pi^{-1}(p)$.*
*Then $\nu_q(C') \leq r$ and if $r \in \mathbb{Z}_{>0}$ there is at most one point $q \in \pi^{-1}(p)$ such that $\nu_q(C') = r$.*

*Proof.* see [26, proof of Lemma 3.10]. □

**Theorem 3.1.4** (Jacobian Criterion, [14, Theorem 2.1])**.** *Let $K$ be an algebraically closed field, let $Z = V(f_1, \ldots, f_s) \subset \mathbb{A}^n_K$ be an affine variety which is equidimensional of dimension $d$. Let*

$$I_Z = \{f \in K[\underline{x}] \mid f(p) = 0 \text{ for all } p \in Z\}$$

*be the vanishing ideal of $Z$. Write $I_{n-d}(\mathcal{J})$ for the ideal generated by the $(n-d)$-minors of the Jacobian matrix $\mathcal{J} = (\frac{\partial f_i}{\partial x_j})$.*

*If $I_{n-d}(\mathcal{J}) + I_Z = \langle 1 \rangle$, then $Z$ is smooth and the ideal $\langle f_1, \ldots, f_s \rangle \subset K[\underline{x}]$ is equal to the vanishing ideal $I_Z$ of $Z$.*

Note that the Jacobian criterion is only appliable for regularity over perfect fields, since over non-perfect fields regularity and smoothness are not equivalent.

**Example 3.1.5** ([26, Example pages 25-26])**.** Let $C = V(y^2 - x^3)$ in $\mathbb{A}^2_K$. With the Jacobian criterion (Theorem 3.1.4), we see that the only singular point is the origin, so $p = V(x, y)$. Let $\pi \colon B\ell_p(C) \to C$ the blow-up of $C$ with center $p$. We denote by $C'$ the strict transform of $C$ and with $E = \pi^{-1}(p)$ the exceptional divisor of $\pi$.
We now have a look at the two charts $U_1 = \operatorname{Spec}(K[\frac{x}{y}, y]) \subset B\ell_p(C)$ and $U_2 = \operatorname{Spec}(K[x, \frac{y}{x}]) \subset B\ell_p(C)$.
In $U_1$ we have coordinates $x_1, y_1$ with $x = x_1 y_1$ and $y = y_1$. So we have $y^2 - x^3 = y_1^2(1 - x_1^3 y_1)$. The exceptional divisor $E$ on $U_1$ is $V(y_1)$ and the strict transform is $CV(1 - x_1^3 y_1)$ on $U_1$. $(1 - x_1^3 y_1)$ is a unit on $E \cap U_1$ so $C' \cap E \cap U_1 = \emptyset$.

On $U_2$ we get, by abuse of notation, the exceptional divisor is $V(x_1)$ and the strict transform of $C$ is $V(y_1^2 - x_1)$, which has order $\leq 1$ everywhere, so $C'$ is regular.

This example provides us with an intuition to an algorithm to resolve curve singularities:

**Construction 3.1.6.** 1. Blow-up the most singular points of the curve.

2. a) If the curve is resolved, we are finished.

   b) If the curve is not resolved, repeat step 1. with the new most singular points as long as the curve is resolved.

**Algorithm 1** Algorithm for resolution of curve singularities

**INPUT:** $C = V(f)$

1: **while** $\exists$ singular points on $C$ **do**
2: Choose a most singular point $x \in C$
3: blow-up with center $x$
4: stop

Figure 3.1 presents a flow chart of the Algorithm 1. Note that we abuse notation in this flow chart. There, $C$ denotes the curve and the strict transforms of $C$.
The algorithm in Construction 3.1.6 only makes sense, if we can guarantee that we only repeat 1. finitely many times, which we will be proven in the following subsection.

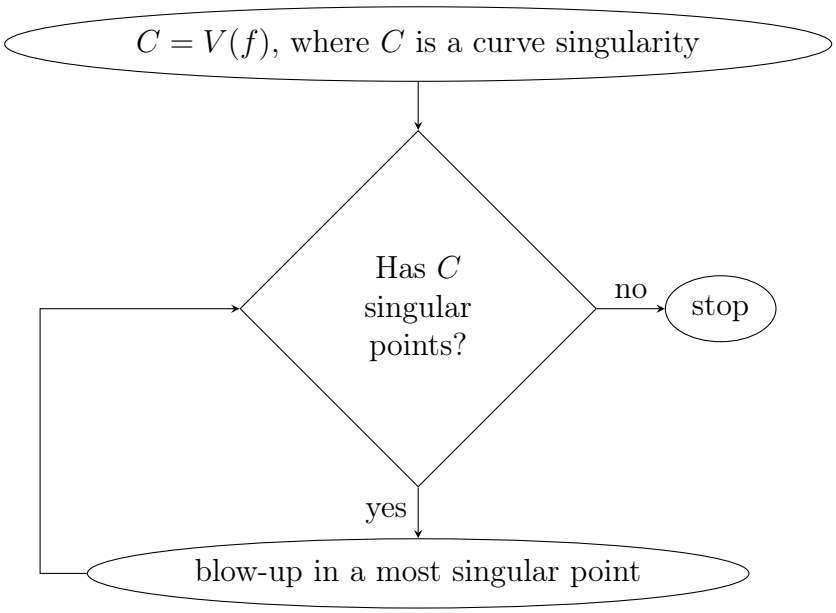



Figure 3.1.: Flow chart of resolution of curve singularities

## 3.1.2. Resolution of curve singularities

In this section, we consider singular curves embedded in regular surface over $K$. Our goal in this section is to prove that Algorithm 1 terminates after finitely many steps.

Before we can state the main theorem of this section, we want to state a Lemma to simplify the proof of this goal.

**Lemma 3.1.7** ([26, Lemma 3.14])**.** *Let $S$ be a regular surface over $K$, let $C$ be a curve on $S$ and let $p$ be a closed point on $C$. Furthermore, let $\pi\colon B\ell_p(S) \to S$ be the blow-up of $S$ with center $p$. We denote by $C'$ the strict transform of $C$ on $B\ell_p(S)$ and assume that $q \in \pi^{-1}(p) \cap C'$.*
*Then $\nu_q(C') \leq \nu_p(C)$, and if we have equality than $K(q) = K(p)$.*

*Proof.* See [26, Lemma 3.14]. □

The following theorem shows that Algorithm 1 terminates after finitely many steps.

**Theorem 3.1.8** ([26, Theorem 3.15])**.** *Let $C$ be a curve which is a subvariety of a regular surface $X$ over a field $K$. Then there exists a sequence of blow-ups of points $\lambda\colon Y \to X$ such that the strict transform $C'$ of $C$ in $Y$ is regular.*

*Proof.* See [26, Theorem 3.15]. □

Note, that the presented resolution of singularities is not an embedded resolution since the exceptional divisors does not need to have normal crossings. Example 2.3.8 illustrates the differences and shows that we have to do some more blow-ups for an embedded resolution of curve singularities.

## 3.2. Resolution of surface singularities in characteristic zero

The first complete proof of the existence of a resolution of surface singularities over $\mathbb{C}$ was given by R. J. Walker [78] in 1935. In 1939, Zariski [80] proved the existence of a resolution of singularities given by irreducible surfaces over algebraic closed fields of characteristic zero. In 1944, he [81] proved the existence of resolution of surface singularities embedded in a regular threefold, over characteristic zero.

We have seen in the section above that it is a good idea to blow-up the whole singular locus in the curve case. Now we want to consider surfaces for which the situation is more complicated but the idea still works but some cautions have to be taken. We will see that the singular locus of a surface consists of a finite number of isolated points and irreducible curves which may even be singular. Curves are not allowed to be centers of blow-ups in a strong resolution if they are not regular or if they intersect. So the idea is to make them regular (we have seen in the section before that this is possible) by some blow-ups then separate them from each other by blow-ups and afterwards take their union as center.

We follow the argumentation of Cutkosky [26] and show how to desingularize surface singularities by the good point algorithm of Abhyankar [2] in characteristic zero.

In this section, we assume $K$ to be an algebraic closed field of characteristic zero. Furthermore we assume $V$ to be a regular three dimensional variety over $K$ and $S \subset V$ is a surface.

**Theorem 3.2.1** ([26, Theorem 5.2])**.** *Let $S$ be a hypersurface of dimension 2 embedded in a regular variety $V$ of dimension 3 over $K$. Then there exists a finite sequence of blow-ups of points and regular curves contained in the strict transform $S_i$ of $S$*

$$V_n \to V_{n-1} \to \cdots \to V_1 \to V,$$

*such that $S_n$ is regular on $V_n$.*

**Definition 3.2.2** ([26, Page 45])**.** Let $t \in \mathbb{N}$. We define the $t$-singular locus of a surface $S \subset V$ as the set

$$\mathrm{Sing}_t(S) := \{p \in V \mid \nu_p(S) \geq t\}.$$

In particular, we use the multiplicity here to construct the $t$-singular locus as the locus where the multiplicity is greater than or equal to $t$.

**Remark 3.2.3.** Recall that the locus $\mathrm{Sing}_t(S)$ is Zariski closed in $V$, since $\nu(\cdot)$ is an upper semi-continuous function.

In the topic of resolution of singularities it is standard to consider only the most singular points of a variety. We have already seen this in Section 2.4.2, where we have defined the maximal order.

From now on, $r := \max\{t \mid \mathrm{Sing}_t(S) \neq 0\}$ denotes the maximal multiplicity of points of $S$.

**Remark 3.2.4.** The irreducible components of $\mathrm{Sing}_r(S)$ are points and curves.

In contrast to the curve singularities, where we only have to blow-up of points, we have to consider in the surface case two types of blow-ups of regular subvarieties on a regular three-dimensional variety. These two types are

1. a blow-up of a point,

2. a blow-up of a regular curve.

So we see that the actual situation is a bit more complicated then the case of a surface singularity. The additional difficulty is that we also have to look at the blow-up in a regular curve.
Lemma 3.2.8 provides us with more information on the singular locus of the strict transforms after blow-up in a regular curve contained in the singular locus. For the proof, we need the Weierstrass preperation theorem (Theorem 3.2.5) and the Tschirnhausen transformation (Definition 3.2.6).

**Lemma 3.2.5** (Weierstrass preperation theorem [26, Lemma 3.7])**.** *Let $K$ be a field and suppose that $f \in K[[x_1, \ldots, x_n, y]]$ fulfills*

$$0 < r = \nu(f(0, \ldots, 0, y)) = \max\{n \mid y^n \text{ divides } f(0, \ldots, 0, y)\} < \infty.$$

*Then there is a unit series $u$ in $K[[x_1, \ldots, x_n, y]]$ and non-unit series $a_i \in K[[x_1, \ldots, x_n]]$ such that*

$$f = u(y^r + a_1 y^{r-1} + \ldots + a_r).$$

*Proof.* See [83, Chapter VII, Section 1, Theorem 5]. □

**Definition 3.2.6** (Tschirnhausen transformation [26, Definition 3.8])**.** Let $K$ be a field of characteristic $p \geq 0$ and let $f \in K[[x_1, \ldots, x_n, y]]$ have an expression

$$f = y^r + a_1 y^{r-1} + \ldots + a_r$$

with $a_i \in K[[x_1, \ldots x_n]]$ and $p = 0$ or $p$ does not divide $r$. Then the Tschirnhausen transformation of $f$ is the change of variables replacing $y$ with $y' = y + \frac{a_1}{r}$.

**Remark 3.2.7.** Let $f$ be like in Definition 3.2.6. By applying the Tschirnhausen transformation, $f$ has an expression

$$f = (y')^r + b_2 (y')^{r-2} + \ldots + b_r,$$

for some $b_i \in K[[x_1, \ldots, x_n]]$, for $i = 2, \ldots, r$.

**Lemma 3.2.8** ([26, Lemma 5.4])**.** *Let $V$ be a regular three-dimensional variety, $S \subset V$ be a surface, $C \subset \mathrm{Sing}_r(S)$ be a regular curve, $\pi \colon B\ell_C(V) \to V$ is the blow-up of $C$ and $S'$ is the strict transform of $S$ in $B\ell_C(V)$. Let $p \in C$ be a closed point.*
*Then $\nu_q(S') \leq r$ for all $q \in \pi^{-1}(p)$ and there is at most one point $q \in \pi^{-1}(p)$ with $\nu_q(S') = r$.*
*In particular, if $E = \pi^{-1}(C)$, then either $\mathrm{Sing}_r(S') \cap E$ is a regular curve which maps isomorphically onto $C$ or $\mathrm{Sing}_r(S') \cap E$ is a finite union of points.*

*Proof.* By applying Weierstrass preparation theorem and a Tschirnhausen transformation. See [26, Lemma 5.4] for more details. □

Now we state the analogous result for a blow-up in a point which follows with the same argument as above.

**Lemma 3.2.9** ([26, Lemma 5.5])**.** *Assume $p \in \mathrm{Sing}_r(S)$ is a point, $\pi \colon B\ell_p(V) \to V$ is the blow-up of $p$, $S'$ is the strict transform of $S$ in $B\ell_p(V)$ and $E = \pi^{-1}(p)$.*
*Then $\nu_q(S') \leq r$ for all $q \in \pi^{-1}(p)$ and either $\mathrm{Sing}_r(S') \cap E$ is a regular curve or $\mathrm{Sing}_r(S') \cap E$ is a finite union of points.*

**Definition 3.2.10.** $\mathrm{Sing}_r(S)$ has simple normal crossings (SNCs) if

1. all reducible components of $\mathrm{Sing}_r(S)$ are regular and
2. $p$ is a singular point of $\mathrm{Sing}_r(S)$, then there exists a system of regular parameters $(x, y, z)$ in $\mathcal{O}_{V,p}$ such that $\mathcal{I}_{\mathrm{Sing}_r(S),p} = \langle xy, z\rangle$.

**Corollary 3.2.11** ([26, Lemma 5.7])**.** *Assume that* $\mathrm{Sing}_r(S)$ *has simple normal crossings,* $W$ *is a point or an irreducible curve contained in* $\mathrm{Sing}_r(S)$*,* $\pi\colon V' = B\ell_W(V) \to V$ *and* $S'$ *is the strict transform of* $S$ *under* $\pi$*. Then* $\mathrm{Sing}_r(S')$ *has simple normal crossings.*

**Definition 3.2.12** ([26, Definition 5.8])**.** We say a closed point $p \in S$ is a pregood point if $\mathrm{Sing}_r(S)$ is either

- empty,
- a regular curve through $p$ in a neighborhood of $p$ or
- a union of two regular curves intersecting transversally at $p$.

**Definition 3.2.13** ([26, Definition 5.9])**.** A closed point $p \in S$ is a good point, if $p$ is a pregood point and for any sequence of blow-ups of regular curves in $\mathrm{Sing}_r(S_i)$

$$X_n \to X_{n-1} \to \ldots \to X_1 \to \operatorname{Spec} \mathcal{O}_{V,p},$$

where $S_i$ is the strict transform of $S \cap \operatorname{Spec} \mathcal{O}_{V,p}$ on $X_i$ implies that $q$ is a pregood point for all closed points $q \in \mathrm{Sing}_r(S_n)$. In particular, $\mathrm{Sing}_r(S_n)$ does not contain isolated points.
A point which is not good is called a bad point.

The following Lemma shows that there is a resolution procedure in which we have to blow-up only the regular curves in $\mathrm{Sing}_r(S_i)$, if all points of $\mathrm{Sing}_r(S)$ are good points.

**Lemma 3.2.14** ([26, Lemma 5.10])**.** *Assume that all points of* $\mathrm{Sing}_r(S)$ *are good. Then there exists a sequence of blow-ups of regular curves containing* $\mathrm{Sing}_r(S_i)$

$$V' = V_n \to \ldots \to V_1 = V,$$

*where* $S_i$ *is the strict transform of* $S$ *in* $V_i$*, such that* $\mathrm{Sing}_r(S') = \emptyset$*, where* $S'$ *is the strict transform of* $S$ *in* $V'$*.*

*Proof.* See [26, Lemma 5.10]. □

With the previous Lemma we know how to finish the resolution if singularities, if we only have good points in $S$. Because the idea is to blow-up in the bad points and than apply Lemma 3.2.14. This is possible in finite many steps if there are only finitely many bad points. This is guaranteed by the following lemma.

**Lemma 3.2.15** ([26, Lemma 5.11])**.** *There are only finitely many bad points on $S$.*

*Proof.* See [26, Lemma 5.11]. □

**Theorem 3.2.16** ([26, Theorem 5.12])**.** *Let*

$$\ldots \to V_n \to V_{n-1} \to \ldots \to V_1 \to V$$

*be the sequence where $\pi_n \colon V_n \to V_{n-1}$ is the blow-up of all bad points in the strict transform $S_{n-1}$ of $S$. This sequence terminates after finitely many steps in a variety $V_m$ such that all points of $\mathrm{Sing}_r(S_m)$ are good.*

*Proof.* See [26, Theorem 5.12]. □

**Construction 3.2.17.** In summary, we can resolve $\mathrm{Sing}_r(S)$ by blowing up all bad points of $S$ and all bad points of the strict transform $S'$ of $S$. After finitely many steps, $S'$ has only good points. Then we apply Lemma 3.2.14 and blow-up in regular curves in $\mathrm{Sing}_r(S')$ in order to resolve $\mathrm{Sing}_r(S')$. After doing so, we can apply induction on $r$ until we reach $\mathrm{Sing}_2(S') = \emptyset$, where $S'$ is the strict transform of $S$ under all of these blowing ups.

This leads us to the Algorithm 2.

**Algorithm 2** Algorithm for resolution of surface singularities

**INPUT:** $S$ a singular surface
1: $r := \max\{t \mid \mathrm{Sing}_t(S) \neq \emptyset\}$
2: **while** $r > 1$ **do**
3: **while** $\exists$ bad points in $S$ and $\mathrm{Sing}_r(S) \neq \emptyset$ **do**
4: Choose a bad point point $x \in S$
5: Do the blow-up with center $x$
6: **while** $\exists$ regular curves on $\mathrm{Sing}_r(S)$ and $\mathrm{Sing}_r(S) \neq \emptyset$ **do**
7: Choose a regular curve $C \subset \mathrm{Sing}_r(S)$
8: Do the blow-up with center $C$
9: $r := \max\{t \mid \mathrm{Sing}_t(S) \neq \emptyset\}$
10: stop

Figure 3.2 presents a flow chart of the Algorithm 2.
By abuse of notation $S$ denotes the strict transform of $S$, too.

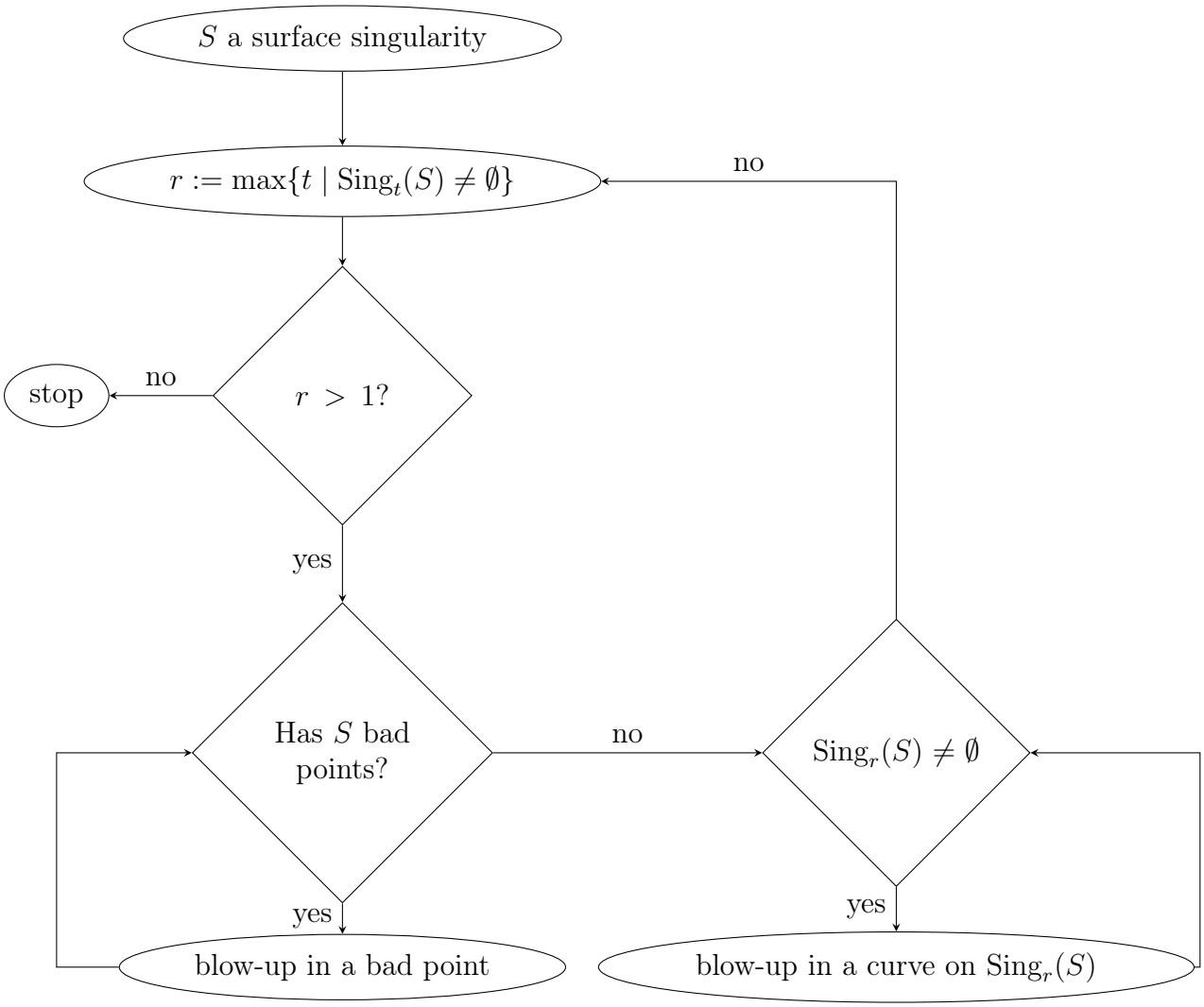



Figure 3.2.: Flow chart of the good point algorithm for resolution of surface singularities

# 4. Simpified main ideas of resolution of determinantal singularities

In this chapter, we want to anticipate the intuitive idea of the main strategies of this thesis.
We have seen in Chapter 2 and Chapter 3 what the resolution of singularities is. We have also noticed that the so-called blow-up (Definition 2.2.1) has proven to be a good tool in resolution of singularities. The presented algorithms for desingularization of curve and surface singularities blow-up in the whole singular locus. Maybe we had to make the singular components regular first. However, this construction fails for three-folds and higher dimensional schemes, mainly since the passage to the singular locus does not commute with blow-ups. The singular locus of $X'$ may have singular components in the exceptional locus, which do not correlate with the singular locus of $X$ (see [50]).
That is why it is more challenging to desingularize determinantal singularities or singularities in general. We discuss the general approach in Section 5.2 and Section 5.3.

The starting point of resolution of determinantal singularities in this thesis is the solved problem (by [75] and [71]) of the resolution of singularities of determinantal singularities generated by generic matrices.

The idea behind the solution to the problem is illustrated in the following remark.

**Remark 4.0.1.** Let $R_0$ be a regular ring, $n \in \mathbb{Z}_+$ and let $R = R_0[x_{i,j} \mid 1 \leq i, j \leq n]$ be the polynomial ring in $n^2$ independent variables.
Let $M = (x_{i,j})_{1 \leq i,j \leq n}$. We want to consider the determinantal singularity $V(f)$ generated by a single equation $f := \det(M)$.
We blow-up in the origin $V(x_{i,j} \mid 1 \leq i, j \leq n)$. Considering only the $X_{1,1}$-chart suffices for an intuitive idea.

The strict transform $f'$ of $f$ is given by

$$f' = \det \begin{pmatrix} 1 & x'_{1,2} & \dots & x'_{1,n} \\ x'_{2,1} & x'_{2,2} & \dots & x'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{n,1} & x'_{n,2} & \dots & x'_{n,n} \end{pmatrix}.$$

Since the addition of rows does not change the determinant, we can add $-x_{1,j}$-times row 1 to row $j$, for $2 \leq j \leq n$. We obtain

$$f' = \det \begin{pmatrix} 1 & x'_{1,2} & \dots & x'_{1,n} \\ 0 & x'_{2,2} - x'_{2,1}x'_{1,2} & \dots & x'_{2,n} - x'_{2,1}x'_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x'_{n,2} - x'_{n,1}x'_{1,n} & \dots & x'_{n,n} - x'_{n,1}x'_{1,n} \end{pmatrix}.$$

By applying $\mathrm{col}_\ell \mapsto \mathrm{col}_\ell - x'_{1,\ell} \cdot \mathrm{col}_1$, for $2 \leq \ell \leq k$, we obtain

$$f' := \det \left( \begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & x'_{2,2} - x'_{2,1}x'_{1,2} & \dots & x'_{2,k} - x'_{2,1}x'_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x'_{r,2} - x'_{r,1}x'_{1,2} & \dots & x'_{r,k} - x'_{r,1}x'_{1,k} \end{array} \right).$$

By the Laplace Expansion of the determinant we get

$$f' = 1 \cdot \det \begin{pmatrix} x'_{2,2} - x'_{2,1}x'_{1,2} & \dots & x'_{2,n} - x'_{2,1}x'_{1,n} \\ \vdots & \ddots & \vdots \\ x'_{n,2} - x'_{n,1}x'_{1,n} & \dots & x'_{n,n} - x'_{n,1}x'_{1,n} \end{pmatrix}.$$

Since all variables are independent, all entries of the former matrix are principalized and have normal crossings. We can apply a change of variables such that variables $y_{i,j} := x_{i,j} - x_{i,1}x_{1,j}$, for $1 \leq i, j \leq n$.

$$f' = 1 \cdot \det \begin{pmatrix} y_{2,2} & \dots & y_{2,n} \\ \vdots & \ddots & \vdots \\ y_{2,n} & \dots & y_{n,n} \end{pmatrix}.$$

So we can trace the problem of desingularizing the determinant of a generic $(n \times n)$-matrix to the problem of desingularizing the determinant of a generic $((n-1) \times (n-1))$-matrix. By induction, this procedure ends after finitely many steps.

The next step is to solve the problem of resolving determinantal singularities generated by a monomial matrix.
The goal is to principalize the monomial ideal $\langle \underline{x}^{A_{i,j}} \mid 1 \leq i,j \leq m \rangle$ generated by the entries of a matrix $M = (\underline{x}^{A_{i,j}})_{1 \leq i,j \leq m} \in K[\underline{x}]^{n \times n}$, where $K$ is an algebraically closed field.
The subgoal is to blow-up in the ideal of the 1-minors of $M$ as before in the generic setting, but this ideal could be an arbitrary monomial ideal which is potentially very singular. We will reduce the problem to the problem of desingularizing a binomial ideal. The algorithm of Blanco and Encinas (see Section 5.4) solves this problem.

After applying the algorithm of Blanco and Encinas, the ideal of entries of the matrix equals $\langle 1 \rangle$. For simplicity, we assume in this chapter that there is a 1 entry in the matrix. In general, this has not to be automatically the case, but we refer here to Chapter 8 for the technical details. After applying the step of the Gaussian algorithm like in the generic case, we have in each chart the following situation: We have a matrix

$$M' = \begin{pmatrix} m_{2,2} - m_{2,1}m_{1,2} & \dots & m_{2,r} - m_{2,1}m_{1,r} \\ \vdots & \ddots & \vdots \\ m_{k,2} - m_{k,1}m_{1,2} & \dots & m_{k,r} - m_{k,1}m_{1,r} \end{pmatrix},$$

where $m_{i,j} \in K[\underline{x}]$ are arbitrary monomials. Since we do not have several variables $x_{i,j}$ but some arbitrary monomials here, we are not guaranteed to find a linear transformation such that every entry of our matrix is a monomial.

Altogether, in local charts, we are in a situation which is simplified described by Figure 4.1.

That explains why our goal is to desingularize determinantal singularities of at most binomial type as the next more difficult problem.

We will later see that our algorithm is not restricted to binomial entries but entries for which we can find a principalization and establish normal crossings. So in characteristic zero, we can use the resolution of Hironaka (see Section 5.2) instead of the algorithm of Blanco and Encinas.

Furthermore, there are topics we need to discuss before presenting the main algorithm of this thesis.

So we have to discuss the algorithm of Villamayor (Section 5.3) for characteristic zero as an example for a Hironaka-style resolution (Section 5.2), the algorithm of Blanco and Encinas (Section 5.4) for the characteristic $p > 0$ case and determinantal

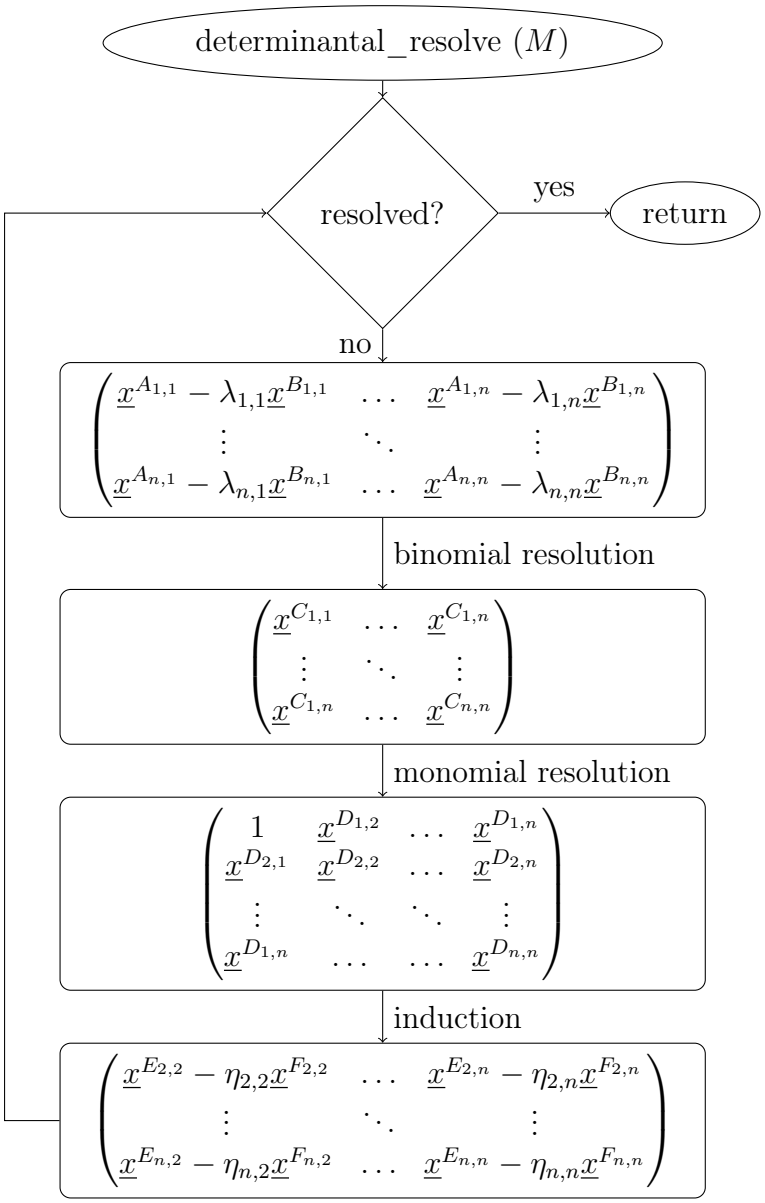



Figure 4.1.: Simplified local illustration of the flowchart of the main algorithm.

singularities of at most binomial type and the algorithm of Hu (Section 5.5) for establishing normal crossings for our entries in Chapter 5.

# 5. Algorithmic background of resolution of singularities

In this chapter, we discuss some algorithmic strategies for resolution of singularities, which influences the main strategy of this thesis. First, we discuss different notions of standard bases (Section 5.1). Then we discuss the main idea of Hironaka's famous proof (Section 5.2). This was the first strategy for resolution of Excellent noetherian schemes in characteristic zero. Since the original proof consists of more than 200 pages, we restrict ourselves to the main ideas and constructions.
In Section 5.3, we give a more constructive view on a Hironaka-style resolution with a briefly overview of the algorithm of Bravo, Encinas and Villamayor.

The Hironaka-style resolution resolves singularities in characteristic 0 but the complexity of a constructive Hironaka-style resolution of an excellent noetherian scheme of dimension $\ell$ is classified in the $(\ell + 3)$-th level of the Grzegorczyk hierarchy, see [5] or Section B.1 for more details about the hierarchy. For practical reasons, it might be interesting to restrict the complexity by dealing with a smaller input class.
After this section, we reduce the complexity of the problem by assuming some special structure of the generators of the singularity. The first such structural requirement is that we only deal with binomial ideals, i.e., ideals with a generating set containing at most binomials. For this, we discuss the algorithm of Blanco and Encinas (Section 5.4), which resolves singularities that can be generated by binomial ideals in arbitrary characteristic. Since the algorithm of Villamayor and the one of Blanco are similar we discuss the main ideas in the more general case in Section 5.3 and the ideas and constructions which are specific for resolution of binomial singularities in Section 5.4. Later in this thesis we need this constructions for the reimplementation of [12]. This is necessary since we need a more modular structure that allows us to reuse the algorithm resp. the choice of center as a black box in our main algorithm in Chapter 8.

We will see that our use of the algorithm of Blanco and Encinas gives us principalized entries which do not need to have normal crossings. To resolve this problem, we use the algorithm of Hu [60] for the resolution of simple arrangements. This algorithm is described in Section 5.5.

For surfaces, there exists a different approach we want to discuss here. This is the algorithm of Cossart, Jannsen, and Saito (CJS algorithm). We will discuss it in Section 5.6. It is formulated for arbitrary dimensional excellent schemes, but it is only known that it resolves surface singularities in characteristic 0, positive characteristic, and even mixed-characteristic (see [20]). Algorithmic aspects can be found in [35]. It is unknown if the CJS algorithm also resolves singularities in larger dimensions. Therefore we have implemented a dimension-free variant in the computer algebra system Singular for studying this algorithm to find some interesting examples. It is to be found in the appendix in Section A.2.

## 5.1. Different notions of standard bases

In this section, we will discuss the different notions of standard bases in terms of computer algebra and resolution of singularities. In the classical setting of computer algebra we refer to Definition 5.1.4.
In the setting of resolution of singularities, a standard basis in the sense of Hironaka (see Definition 5.1.1) is often meant.
Both constructions are in the same flavor differing in minor aspects. While standard bases in computer algebra belong to an arbitrary monomial ordering, Hironaka standard bases are a local construction.

After defining both cases, we want to discuss the idea behind them and their commonalities.

In order to compute and control the invariants associated with the ideal which have to be resolved, Hironaka introduced standard bases.

**Definition 5.1.1.** Let $R$ be a local ring, $\mathfrak{m}$ its maximal ideal, and $0 \neq I \subset R$ be an ideal in $R$. Let $\{\underline{f}\} = \{f_1, \ldots, f_r\}$ be a set of elements in $I$. Set $b_i := \mathrm{ord}_{\mathfrak{m}}(f_i)$ and $F_i := \mathrm{in}_{\mathfrak{m}}(f_i)$, for $1 \leq i \leq r$.
Then $\{\underline{f}\}$ is called a standard basis for $I$ (in the Hironaka-sense) if the following properties hold:

1. $\langle F_1, \ldots, F_r \rangle = \mathrm{In}_{\mathfrak{m}}(I) \subset \mathrm{gr}_{\mathfrak{m}}(R)$,

2. $b_1 \leq b_2 \leq \ldots \leq b_r$ and

3. $F_i \notin \langle F_1, \ldots F_{i-1} \rangle$, for all $2 \leq i \leq r$.

For detecting singularities and controlling the resolution process, the theory of standard bases for local rings and their completions in the Hironaka-sense were developed.

**Remark 5.1.2.**

- A standard basis for $I$ generates $I$.
- Let $\{\underline{f}\}$ be a standard basis for $I$. The strict transform of $\{\underline{f}\}$ under reasonable blow-up generates the strict transform of $I$ under the same blow-up. This is not true for any set of generators, in general. Let $I := \langle x^2 - y^3, x^2 - z^5 \rangle \subset K[x,y,z]_{\langle x,y,z \rangle}$, where $K$ is any field. The blow-up of the maximal ideal is a counterexample:
  If we consider only the $X$-chart, the strict transform of the generators are $1 - xy^3$ and $y^3 - z^5x^2$ and $J := \langle 1 - xy^3, y^3 - z^5x^2 \rangle \neq I' = \langle 1 - xy^3, 1 - x^3z^5 \rangle$, where $I'$ is the strict transform of $I$.

This remark illustrates the reason why Hironaka standard bases play a big role in resolution of singularities.

Now we take a look on the standard bases in the computer algebra setting. First, we consider the global construction of Gröbner bases.

A Gröbner basis is defined with respect to a global monomial ordering. For a fixed monomial ordering $\leq$ we denote by $K[\underline{x}]_{\leq}$ the polynomial ring $K[\underline{x}]$ equipped with the monomial ordering $\leq$.

**Example 5.1.3** ([46, Pages 13 and 14])**.** Let $n$ be the number of variables in $K[\underline{x}]$. The following are examples of monomial orderings:

1. The lexicographical ordering $\leq_{\ell p}$ is defined as

$$\underline{x}^{\alpha} <_{\ell p} \underline{x}^{\beta} \quad \Longleftrightarrow \exists 1 \leq i \leq n \colon \alpha_1 = \beta_1, \ldots, \alpha_{i-1} = \beta_{i-1}, \alpha_i < \beta_i,$$

   i.e., $\underline{x}^{\alpha} <_{\ell p} \underline{x}^{\beta}$ if the number of occurrences of $x_1$ in $\underline{x}^{\alpha}$ is less than in $\underline{x}^{\beta}$, or they are equal and the number of occurrences of $x_2$ in $\underline{x}^{\alpha}$ is less than in $\underline{x}^{\beta}$, or they are equal too and so on.

2. The degree lexicographical ordering $\leq_{Dp}$ is defined as

$$\underline{x}^{\alpha} <_{Dp} \underline{x}^{\beta} \quad \Longleftrightarrow \quad \sum_i \alpha_i < \sum_i \beta_i \text{ or } \left( \sum_i \alpha_i = \sum_i \beta_i \text{ and } \underline{x}^{\alpha} <_{\ell p} \underline{x}^{\beta} \right).$$

   In other words, the degree lexicographical ordering first sorts by total degree, and then uses lexicographical ordering as a tie-breaker.

3. The degree reverse lexicographical ordering $\leq_{dp}$ is defined as

$$\underline{x}^{\alpha} <_{dp} \underline{x}^{\beta} \quad \Longleftrightarrow \quad \sum_i \alpha_i < \sum_i \beta_i \text{ or } \left( \sum_i \alpha_i = \sum_i \beta_i \text{ and } \exists 1 \leq i \leq n \colon \alpha_n = \beta_n, \ldots, \alpha_{i+1} = \beta_{i+1}, \alpha_i < \beta_i \right).$$

   In other words, the degree reverse lexicographical ordering first sorts by total degree, and then uses the reverse lexicographical ordering, i.e., the lexicographical ordering starting at the end and comparing in the reverse order than the lexicographical ordering, as a tie-breaker.

**Definition 5.1.4** ([46, Definition 1.6.1])**.** Let $R$ be the localization of $K[\underline{x}]$ with respect to a fixed monomial ordering $<$ and let $I \subset R$ be an ideal.

1. A finite set $G \subset R$ is called a standard basis (in the computer algebra sense) of $I$ if

$$G \subset I \text{ and } L(I) = L(G),$$

   i.e., $G$ is a standard basis, if the leading monomials of the elements of $G$ generate the leading ideal of $I$. Note, that $L(G)$ resp. $L(I)$ denotes the leading ideal of $G$ resp. $I$.

2. If $<$ is a global ordering, a standard basis is called Gröbner basis.

**Remark 5.1.5.** Note, that a standard basis referring to an arbitrary monomial ordering while Gröbner bases need a global monomial ordering.

Definition 5.1.4.1. says that for any $f \backslash \{0\}$ there exists a $g \in G$ with $\mathrm{LM}(g) \mid \mathrm{LM}(f)$, where LM denotes the leading monomial.

**Definition 5.1.6** ([46, Definition 1.6.2])**.** Let $R$ be the localization of $K[\underline{x}]$ with respect to a fixed monomial ordering $<$ and let $G \subset R$ be any subset.

1. $G$ is called interreduced if $0 \notin G$ and if $\mathrm{LM}(g) \nmid \mathrm{LM}(f)$ for any two elements $f \neq g \in G$.

2. $f \in R$ is called reduced with respect to $G$ if no monomial of the power series expansion of $f$ is contained in $\mathrm{L}(G)$.

3. $G$ is called reduced if $G$ is interreduced and if for any $g \in G$ the leading coefficient of $g$ equals 1 and the tail of $g$ is completely reduced with respect to $G$.

**Remark 5.1.7.** Let $R$ be the localization of $K[\underline{x}]$ with respect to a fixed global monomial ordering $<$. For every ideal $I \subset R$ a unique reduced Gröbner base can be computed.

**Definition 5.1.8** ([46, Definition 1.6.4])**.** Let $R$ be the localization of $K[\underline{x}]$ with respect to a fixed monomial ordering $<$. Let $\mathcal{G}$ be the set of all finite lists $G \subset R$.

$$\mathrm{NF}\colon R \times \mathcal{G} \to R, (f, G) \mapsto \mathrm{NF}(f \mid G),$$

is called a normal form on $R$, if for all $G \in \mathcal{G}$ and for all $f \in R$,

1. $\mathrm{NF}(0 \mid G) = 0$

2. $\mathrm{NF}(f \mid G) \neq 0 \implies \mathrm{LM}(\mathrm{NF}(f \mid G)) \notin L(G)$.

3. If $G = \{g_1, \ldots, g_s\}$ then $f - \mathrm{NF}(f \mid G)$ has a standard representation with respect to $\mathrm{NF}(- \mid G)$,

$$f - \mathrm{NF}(f \mid G) = \sum_{i=1}^{s} a_i g_i, a_i \in R, s \geq, 0$$

such that $\mathrm{LM}(\sum_{i=1}^{s} a_i g_i) \geq \mathrm{LM}(a_i g_i)$ for all $i$.

NF is called a reduced normal form if $\mathrm{NF}(f \mid G)$ is also reduced with respect to $G$.

The following remark states equivalent characterizations of Gröbner bases.

**Remark 5.1.9.** Let $I$ be an ideal in $R = K[\underline{x}]_{\leq}$ and let $G = \{g_1, \ldots, g_s\} \subseteq I$ be a finite set. The following statements are equivalent.

- $\mathrm{In}(I) = \langle \mathrm{LT}(G) \rangle$, where LT denotes the sets of all leading terms of polynomials in $G$.

- For all $f \in R$ there is a unique $r$ such that $f - r \in I$ and no $\mathrm{LT}(g_i)$ divides any $m \in \mathrm{supp}(r)$.

- For all $f \in R$, $f \in I$ if and only if $\mathrm{NF}(f \mid G) = 0$.

**Remark 5.1.10** (Computational aspects of Gröbner bases).

- The computation of Gröber bases is implemented in several computer algebra systems, e.g., Singular, Oscar, Magma, Maple,...
- The computational complexity of a Gröbner basis computation is EXPSPACE-complete. For more information on the complexity reductions, we refer to [61]. For more information about the notion of EXPSPACE-completeness, see Appendix B.1.

Gröbner bases and standard bases in the sense of Hironaka define the same construction in a slightly different manner. While Gröbner bases work with a fixed global ordering, standard bases in the computer algebra sense work in arbitrary orderings and Hironaka standard bases in local orderings. That is why they need a modified normalform algorithm in the sense of Mora.
Furthermore, Gröbner bases and standard bases in the sense of computer algebra consider the initial terms of the leading terms or initial monomials and Hironaka standard bases consider the initial form.

**Example 5.1.11.** Let $f = x^2 + y^2 + z^3 \in K[x, y, z]$. The leading term of $f$ with respect to $\leq_{dp}$ is $z^3$.

**Remark 5.1.12.** A Gröbner basis with respect to a negative weighted degree ordering is also a standard basis in the sense of Hironaka, since the ordering refines Hironaka's ordering on the initial parts.
In the homogeonous setting, the lexicographical order coincide with the graded lexicographical order, so Gröbner bases and standard bases coincide, too.
We will use this special case in Section 6.2.

In the theoretical part of this thesis, we are dealing with standard bases in the case of Hironaka. Later, when constructing the algorithms, we calculate Gröbner bases, too, when we can guarantee that they coincide with the standard bases in the Hironaka sense.

## 5.2. Ideas of a Hironaka-style Resolution

We have seen in Section 3 that it is a good idea to consider the whole singular locus as the center of blow-ups for curves and surfaces. Maybe we had to make the singular components regular first. However, this center construction fails for three-folds and higher dimensional schemes. This mostly relies to the fact that the passage to the

singular locus does not commute with blow-ups. The singular locus of $X'$ may have singular components in the exceptional locus, which have nothing to do with the singular locus of $X$ (see [50]). That is the reason why it seems impossible to make the singular locus regular as it is done in the surface case. This section should give a simplified intuition to the main ideas of a Hironaka-style resolution. More general (and algorithmic) formulations of such a resolution are given in the next two sections.

The strategy of a Hironaka-style resolution is the following: We have to develop certain invariants which measure how 'good' or 'bad' a single point of a singularity is. These invariants should construct a sequence of blow-ups for which the invariants are non-increasing and can decrease only finitely many times. When they decrease to the minimum, we should be able to conclude that we have reached a regular scheme.
So the choices for the centers of the blow-ups are made canonical by considering the loci where the invariants are maximal, i.e., we blow-up the 'worst' locus. This is illustrated by Algorithm 3 and Figure 5.1. The main point of the proof is to show that the invariants finally decrease.

Considering Theorem 2.3.7 (embedded resolution of singularities) from a practical point of view, one have to discuss two central tasks.

1. The calculation of the blow-up (see Section 2.2)
2. The choice of the center
   a) strategy
   b) canonicity

There is a flexibility in the choice of the upper semi-continuous function (see Section 2.4) controlling the process.
We take a special invariant and calculate the locus of the maximal invariant. This locus is the center of the next blow-up. This invariant has to fulfill the following properties:

1. Zariski- upper semi-continuous
2. infinitesimally upper semi-continuous
3. show when resolution of singularities is finished
4. being constant on the complement of the singular locus.

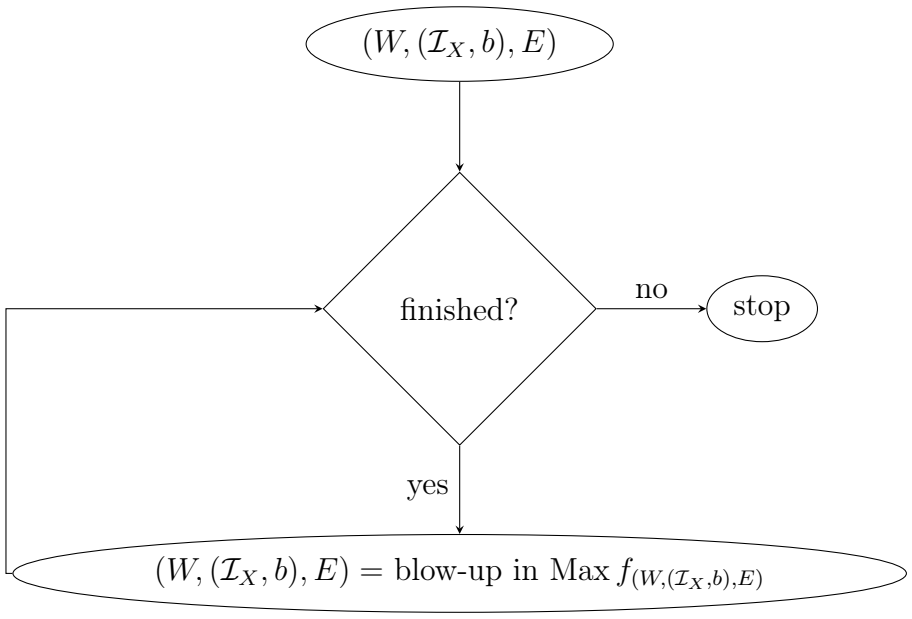



Figure 5.1.: Flow chart of a Hironaka-style resolution

The Zariski-upper semi-continuity implies that the chosen center is Zariski closed. The infinitesimally upper semi-continuity implies that the invariant will not grow during the process.

Algorithm 3 and Figure 5.1 illustrate this procedure. Again, we abuse notation and every strict transform of the basic object is also denoted as $(W, (\mathcal{I}_X, b), E)$.

The heart of the resolution process is to find an invariant $f_{(W,(\mathcal{I}_X,b),E)}$ with respect to a basic object $(W, (\mathcal{I}_X, b), E)$ and with some special properties, discussed in this Section. We follow the argumentation of [51], [52] and [36].

**Algorithm 3** Hironaka-style resolution

**INPUT:** $(W, (\mathcal{I}_X, b), E)$

1: **while** $f_{(W,(\mathcal{I}_X,b),E)}$ is not minimal **do**
2: Calculate the center $D = \text{Max}\, f_{(W,(\mathcal{I}_X,b),E)}$ ▷ depending on the strategy
3: $(W, (\mathcal{I}_X, b), E)$ = blow-up with center $D$
4: stop

In contrast to the previous sections, the general situation is more complicated than the curve or the surface case. We had discussed in Section 2.3.1 how to take the exceptional divisors into account, and we will see in Example 5.6.3 that it is necessary to distinguish the exceptional components.

Since the original proof of Hironaka is about 200 pages and is very technical, we only give an intuition of the basic ideas of the proof here. For simplicity, we will

sometimes change the point of view to a covering of affine charts. There we will abuse notation and we will look at a chart and the corresponding ideal $J$ instead of the global ideal sheaf $\mathcal{I}_X$.
More technical details can be found in Section 5.3 where the notion of basic objects come into play. This present section only gives a simplified intuition.

Like in Theorem 2.3.7, we assume $X$ to be a scheme in a regular ambient space $W$. The scheme $X$ is defined by an ideal sheaf $\mathcal{I}_X \subset \mathcal{O}_W$. Furthermore, we assume that the embedding of $X$ in $W$ is minimal.
The basic information on how singular $X$ is at a point $a$ can be found in the number $\mathrm{ord}_a(\mathcal{I}_X)$. This is the same number we obtain when working in the completed ring $\hat{\mathcal{O}}_{W,a}$ instead of $\mathcal{O}_{W,a}$.
We know by [54, Chapter III §3 Corollary 1 p.220] that ord is an upper semi-continuous function, so for all $c \in \mathbb{N}$ the locus of order at least $c$, $\mathrm{Sing}(\mathcal{I}_X, c)$ is closed in $X$.
Hence, we have a stratification of $X$ by locally closed subschemes along which the order of $X$ is constant. Max-ord$(X)$ is the smallest subscheme in this stratification. It is a closed reduced subscheme of $X$, which might be singular.

**Remark 5.2.1.** Let $a$ be a point in $X \subset W$. Let $\pi\colon W' \to W$ be the blow-up of $W$ in a regular center $Z$ contained in a stratum along which the order of $\mathcal{I}_X$ is constant. We denote with $X^{st}$ the strict transform of $X$. Then

$$\mathrm{ord}_{a'}(X^{st}) \leq \mathrm{ord}_a(X).$$

So the order of the strict transform $X^{st}$ at the preimage $a'$ of $a$ does not increase. In particular, this holds for $Z$ being contained in Max-ord$(X)$. We call such a $Z$ *permissible center*.

**Remark 5.2.2.** Note, that the order of the total transform of $X$ usually increases in the situation of Remark 5.2.1.

After blow-up in a center contained in the locus of maximal order of $Z$ the order of $X$ decreases at most points of the exceptional divisor. At these points, the situation has improved and we can apply an induction on the order. We call this *vertical induction*.

In some special point, which we call equiconstant or very infinitely near points of $\mathcal{I}_X$ or $X$ in $W'$, the order remains constant.
The following example shows an illustration of this fact in an affine chart of $X$.

**Example 5.2.3** ([51, page 339])**.** Let $g = x^2 - z^3(z - y^2) \in K[x, y, z]$. Then $\text{ord}_{\langle x,y,z\rangle}(g) = 2$. After blow-up in the origin, we consider the strict transform $g' = x^2 - z(z - y^2)$ of $g$ in the $Z$-chart. Again, $\text{ord}_{\langle x,y,z\rangle}(g') = 2$, so the origin is an equiconstant point.

At these points, we need extra information on the singularity of $X'$ in order to have a measure that the situation there also improved. The idea is to add a second local invariant as second element of a pair. We want to measure the improvement by a comparison with respect to the lexicographical order.

The equiconstant points of $\mathcal{I}_X$ in the exceptional divisor $Y'$ can be determined from the tangent cone of $\mathcal{I}_X$ at $a$.

**Remark 5.2.4** ([51, page 347])**.** Let $x_1, \ldots, x_n$ be coordinates of $W$ at $a$ such that the coordinate $x_i$ appears in the tangent cone of $\mathcal{I}_X$ for some $i \leq k$ then the equiconstant points $a'$ above $a$ lie in the hypersurface $V(x_i)$ of $W'$.
In other words, let $V \subset W$ be the regular hypersurface $V(x_i)$ and let $V' \subset W'$ be its strict transform. Then all equiconstant points of $\mathcal{I}_X$ lie in $V'$.
We call such hypersurfaces adjacent to $\mathcal{I}_X$.

We use in the remark that the coordinates are chosen such that the number of coordinates appearing in the tangent cone is minimal.

Assume, this hypersurface $V'$ has a regular image $V$ in $W$ and that $V' = V^{st}$ is the strict transform of $V$. Then the following diagram commutates:

$$\begin{array}{rcl} V' & \subset & W' \\ \pi|_{V'} \downarrow & & \downarrow \pi \\ Z \subset V & \subset & W \end{array}$$

Let $a \in Z \subset V$ be a point. Let $a' \in Y' \cap V'$ be a point above $a$. The task still is to find the second component of the induction invariant of $J$ at $a$ and its strict transforms $J' = J^{st}$. The natural idea is to use $V$ and $V'$ for its definition.

The main idea is to associate to the ideals $J$ in $W$ a $J'$ in $W'$ and some ideals $J\_$ in $V$ and $(J')\_$ in $V'$ which measure the improvement. The minus sign in the index refers to the decrease in the embedding dimension. Additionally, we want to

construct these ideal such that $(J')\_$ is the transform of $J\_$ in $V'$, i.e., the diagram

$$\begin{array}{ccc} J' & \to & (J')\_ = (J\_)' \\ \pi|_{J'} \downarrow & & \downarrow \pi|_{(J')\_} \\ J & \to & J\_ \end{array}$$

commutates. This means that the descent in dimension has to commutate with blow-ups.

This additional requirement allows us to control the change between $J\_$ and $(J')\_$.
Again, if the center $Z$ is contained in Max-ord$(J_)$ with Remark 5.2.1 the order would not increase after blow-up. So we can apply induction here. Furthermore, the center $Z$ has to be contained in Max-ord$(J)$ and Max-ord$(J_)$. It remains to be shown that the order does not depend on the local choice of $V$.

$J\_$ is in $V$ defined in a lower dimensional ambient space. By induction, we may assume that we know how to associate a local invariant with $J\_$. This invariant has to be a vector of numbers given as the order of a string of ideals and we have to measure the improvement by the lexicographical order. This is the Hironaka invariant $\nu^*$. We want to call this induction *horizontal induction.*

Using this horizontal induction we either arrive in some dimension at an ideal of order 0 or at dimension 1.

1. The first case is a special case what Hironaka called monomial case. We will discuss this case at the end of Section 5.3.

2. The second case uses the fact that the order of an ideal in a one-dimensional regular scheme always drops under blow-up to 0, when passing through its strict transform.

So we have to answer the question of how to construct the ideal $J\_$. There are at least two possible ways. The first is to start from the ambient space $W$, and the second is to start from a hypersurface $V$.

- We can start from $X = V(J)$ in $W$. This could only work locally on $W$, depending on the choice of the local hypersurface $V$. This choice is neither unique, nor patches on overlaps give a global hypersurface.
  So we consider the other direction.

- We start from $V$. It is only required that at each point $a$ in $W$, the hypersurface $V$ accompanies the resolution process of $J$ as long as the order of $J$ remains constant. Such a hypersurface is said to be in permanent contact with $J$. In characteristic 0, such a hypersurface exists. We can even choose such a hypersurface so it contains the locus of maximal order of $J$ locally. Such a hypersurface is called a hypersurface of maximal contact.

So we use the second way.

Let us have a look at examples of how such hypersurfaces $V$ may be computed in affine charts.

**Example 5.2.5** ([51, Example 4])**.** Let $X = V(x^a+y^b)$ in $\mathbb{A}^2$ with $b \geq a$ and let $J$ be the ideal describing $\mathcal{I}_X$ in this chart. So max-ord$(J) = a$ and Max-ord$(\mathcal{J}) = V(x, y)$. The blow-up $W'$ of $W = \mathbb{A}^2$ in the origin is covered by two affine charts. The total transforms are $x^a+x^by^b = x^a(1+x^{b-a}y^b)$ in the $x$-chart and $x^ay^a+y^b = y^a(x^a+y^{b-a})$ in the $Y$-chart. We are only interested in points $a' \in Y' \cong \mathbb{P}^1$. It is useful to partition $Y'$ into two sets, namely the entire $X$-chart and the orgin of the $Y$-chart. In the first set the order of $J'$ is everywhere 0. So we do not need to consider these points. We are left with the origin of the $Y$-chart with strict transform defined by $x^a + y^{b-a}$. Obviously the origin lies in the hypersurface $V(x)$ in the $Y$-chart of $W'$. Whether the order has dropped or not depends on the value of $b - a$. If $b \geq 2a$ the order has not dropped.
For later reference we note that the order of restriction of $J_{|_{V(x)}}$ comes into play. The strict transform $V'$ of $V(x)$ contains the only possible equiconstant point. The improvement of $J'$ seems to be captured by $J'_{|_{V'}} = \langle y^{b-a} \rangle$ whose order is strictly smaller than $J_{|_{V(x)}} = \langle y^b \rangle$.

Now we consider an example for a generalization of the whitney umbrella ($a = b = 2, c = 1$):

**Example 5.2.6** ([51, Example 5])**.** Now we consider the surface $X = V(x^a + y^bz^c)$ with $b + c \geq a$. Then $V(x, y, z)$ is contained in the locus of maximal order of $X$.

- If $b, c < a$, then Max-ord$(X) = V(x, y, z)$.
- If $b < a \leq c$, then Max-ord$(X) = V(x, z)$.
- If $c < a \leq b$, then Max-ord$(X) = V(x, y)$ .
- If $b, c \geq a$, then Max-ord$(X) = V(x, y) \cup V(x, z)$. In this case there are three possible choices for $Z$:

- $V(x, y, z)$,
- $V(x, y)$ or
- $V(x, z)$.

In all cases the tangent cone of $J$ consists of the monomial $x^a$ except if $b + c = a$, in which case it is $x^a + y^b z^c$.
The hypersurface $V = V(x)$ is always a good candidate for finding equiconstant points since $x$ appears in the tangent cone.

**Remark 5.2.7** ([51, page 376])**.** Note, that in the literature it is often a good idea to select a center among the possible candidates by taking the intersection of components whose total age (sum of individual ages (number of blow-ups after which the components is born) of the components) is the largest or the smallest.

In characteristic $p > 0$ the locus of maximal order is not contained in any regular hypersurface, in general, since hypersurfaces of maximal contact do not need to exist. That is one of the main points why Hironaka's proof fails in positive characteristic. The following example illustrates the problem for characteristic 2.

**Example 5.2.8** ([67])**.** Let $f = x^2 + yz^3 + zw^3 + y^7 w \in \mathbb{F}_2[x, y, w, z]$ and $X = V(f) \subset \mathbb{A}^4_{\mathbb{F}_2}$. The maximal order is 2 and the singular locus is given by

$$\mathrm{Sing}(X) = V(x^2 + yz^3 + zw^3 + y^7 w, z^3 + y^6 w, yz^2 + w^3, zw^2 + y^7).$$

The singular locus contains the curve $C := \mathrm{Im}(t \mapsto (t^{32}, t^7, t^{19}, t^{15}))$. $C$ is not contained in any smooth hypersurface:
Assume that $V(g)$ is a hypersurface containing $C$ and assume that $V(g)$ is smooth at the origin. Then one of $x, y, w, z$ appears linearly in $g$ and $g(t^{32}, t^7, t^{19}, t^{15}) \equiv 0$. This linear term yields a nonzero $t^m$ for some $m$ in $\{32, 7, 19, 15\}$, so it must be canceled out by another term $t^n$.
This implies that we can write $m = 32a + 7b + 19c + 15d$ and $a + b + c + d \geq 2$ for $a, b, c, d \geq 0$. This is a contradiction since none of the numbers $32, 7, 19, 15$ is a positive linear combination of the others.

Back in characteristic 0, assuming we have chosen a hypersurface of maximal contact $V$. It is essential here that $V$ in $W$ has the transform $V'$ in $W'$, which contains all equiconstant points of $J$ in $W'$. We can expand each element of $J$ as a power series with respect to a local coordinate defining $V$ in $W$. The resulting coefficients can be equilibrated by raising them to a suitable power and generating an ideal in $V$. This ideal is called the coefficient ideal $\mathrm{Coeff}_V(J)$ of $J$ in $V$ (see

Definition 5.3.4 for more details). Analogous ideals can be constructed for $J'$ and $V'$. We are not ready yet, since the coefficient ideal of the strict transform $J'$ of $J$ at an equiconstant point $a'$ is not the strict transform of the coefficient ideal of $J$ at $a$. So the diagram does not commutate.

This can be remedied by suitable factorization of the coefficient ideal $\mathrm{Coeff}_V(J)$ into a product of a principle monomial ideal supported by the exceptional divisor and another ideal. This other ideal is the relevant part. With this second part, the commutativity with blow-ups can be established. This second ideal is the searched $J\_$.

Altogether, we have seen that the basis of the proof is a cartesian induction, i.e., the horizontal induction on the local embedding dimension is merged with the vertical induction on the resolution invariant.

**Remark 5.2.9** ([51, page 381])**.** The properties commutativity and decrease are proven by descending horizontal induction on the embedding dimension and refer to the vertical map given by the blow-up. The property transversality is proven by the vertical induction on the sequence of blow-ups and refers to the horizontal structure in $W$.

Hironaka's original proof is not constructive. E.g., the hypersurface of maximal contact was not constructed.
A constructive variant of Bravo, Encinas and Villamayor is briefly discussed in the following section.

## 5.3. Resolution of singularities in characteristic zero: Algorithm of Bravo, Encinas and Villamayor

In this section we want to convey an intuition of the main ideas when constructing a Hironaka-style resolution. Whereas the Section 5.2 gives the ideas and intuitions this section deals with some more details about the definitions and constructions. More specific constructions are given in the much easier binomial case in Section 5.4. The first constructive resolution of singularities was given by Bravo, Encinas and Villamayor [16] and is implemented in the computer algebra system Singular by Frühbis-Krüger and Pfister [37] and in maple by Bodnár and Schicho [13].

We use parts of this theoretical background for the reimplementation of Blanco's algorithm in Section A.3.

Hironaka gave an existential proof of resolution of singularities and also principalization of ideals in characteristic zero. Originally, he considers a stratification of the considered variety by means of the Hilbert-Samuel function and provides an existential argument to show that the maximum of the Hilbert-Samuel function drops.

Bravo, Encinas and Villamayor show that embedded desingularization can be archieved avoiding the Hilbert-Samuel function and fills the non-constructive gaps in the proof. Their strategy is based on the reduction of a simpler result, namely the algorithmic resolution of basic objects. Embedded desingularization is a direct consequence of the existence of an algorithmic resolution of basic objects.
As before in Section 5.2, we assume the base field $K$ to be of characteristic 0 in this section.

**Definition 5.3.1** ([16, Definition 5.5])**.** Let $(W, (\mathcal{I}_X, b), E)$ be a basic object. Let $(\mathfrak{J}, \leq)$ be a totally ordered set. A family of functions

$$f_{(W,(J,b),E)} \colon X \to \mathfrak{J},$$

which is equivariant under isomorphisms of basic objects, is said to be governing a blow-up

$$\pi \colon (W_1, (\mathcal{I}_{X_1}, b), E_1) \to (W_0, (\mathcal{I}_{X_0}, b), E_0)$$

if the following conditions hold:

1. The set of points $\underline{\mathrm{Max}} f_{(W_0,(\mathcal{I}_{X_0},b),E_0)} \subset X_0$, where $f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}$ takes its maximal value $\max f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}$, is a closed subset of $W_0$.

2. $\underline{\mathrm{Max}} f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}$ is a permissible center, i.e., it is regular, has normal crossings with $E_0$ and is disjoint from

$$\{x \in X_0 \mid x \notin \mathrm{Sing}(X_0, b), x \notin E_0, \text{ for all } 1 \leq i \leq \#E_0\}.$$

3. $\underline{\mathrm{Max}} f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}$ is the center of the blow-up $\pi$.

4. $\max f_{(W_1,(\mathcal{I}_{X_1},b),E_1)} < \max f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}$.

5. $f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}(x) = f_{(W_1,(\mathcal{I}_{X_1},b),E_1)}(x)$ for all points $x \in X_0 \setminus \mathrm{Max}\, f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}$.

An algorithm for the resolution of basic objects consists of such a family of functions dictating the subsequent blow-ups for any given basic object $(W, (\mathcal{I}_X, b), E)$ subject to the additional conditions that

1. There is a finite index $N$ depending on the basic object such that the object is resolved after $N$ steps.

2. If $X_0$ is a regular pure-dimensional subscheme of dimension $r$, $b = 1$ and $E_0 = \emptyset$, then there is a value $\mathfrak{s}(r) \in \mathfrak{J}$ such that $f_{(W_0,(\mathcal{I}_{X_0},b),E_0)}(x) = \mathfrak{s}(r)$ for all $x \in X_0$.

Given a basic object $(W, (\mathcal{I}_X, b), E)$ the first two entries of the governing function are

$$f_{(W,(\mathcal{I}_X,b),E),\text{trunc}} = (\text{ord}_x(\mathcal{I}), N_E(x)),$$

where $N_E(x)$ is an integer which counts the exceptional divisors containing $x$ and which are been born before the order at $x$ attained its current value.

**Definition 5.3.2** ([36, Definition 45])**.** Let $(W, \mathcal{I}_X, b, E)$ be a basic object, where $b$ is the maximal order of $\mathcal{I}_X$. A smooth hypersurface $Z \subset W$ is called a hypersurface of maximal contact, if it fulfills all of the following requirements:

1. For every open set $U \subset W$, the locus of maximal value of the truncated governing function $f_{(W,(\mathcal{I}_X,b),E),\text{trunc}}$ is contained in $Z|_U$.

2. For every open set $U \subset W$ and every sequence of blow-up at centers of maximal value of the invariant function starting at the basic object $(U, \mathcal{I}_X|_U, b, E|_U)$ the center of every blow-up is again contained in the respective strict transform $Z|_U$.

3. $Z$ has transversal intersections with each exceptional divisor which arose after the maximal order dropped to the current value.

4. The set $\{E_i \cap Z \mid E_i \in E \text{ born after maximal order dropped to } b\}$ is normal crossing.

The conditions 1. and 2. ensure that we do not loose any points of maximal value of the truncated governing function when passing to the hypersurface of maximal contact and that we do not need to choose a new hypersurface as long as our maximal value does not drop.
Condition 3. ensures that centers determined by means of passing to the hypersurface

$Z$ are also permissible as centers for the given basic objects. Condition 4. ensures normal crossings after passing to $Z$.

We have seen in Section 5.2, that we want to descent in dimension. Therefore we have to construct a new basic object with help of the coefficient ideal. We mark the descent in dimension by ; and we get the whole invariant

$$f_{\mathcal{B}}(x) = (f_{\mathcal{B},\text{trunc}}(x); f_{\mathcal{B}_{\dim(W)-1},\text{trunc}}(x); \ldots; f_{\mathcal{B}_2,\text{trunc}}(x)),$$

where $\mathcal{B} = (W, (\mathcal{I}_X, c), E)$ and $\mathcal{B}_{\dim(W)-1}, \ldots, \mathcal{B}_2$ are the auxiliary basic objects in the descent in dimension.

From the constructive point of view, the only thing to do is to calculate the choice of center, since this leads to a resolution of singularities. We split this up into several subtasks:

1. Computation of the locus of maximal order of a given basic object (Section A.1).
2. Computation of the locus of maximal $N_E$ inside the locus of maximal order.
3. Descent in dimension and construction of the auxiliary basic object.

Since we discuss the first part in the Appendix in Section A.1 and the second part is straight forward, we only have to discuss the third subtask, here.

Our task is to find a hypersurface of maximal contact and determine an auxiliary basic object permitting the induction step of the resolution process depending on the basic object $(W, (\mathcal{I}_X, c), E)$.
We can choose a hypersurface of maximal contact as follows:

**Lemma 5.3.3** ([36, Lemma 56])**.** *Let $(W, (\mathcal{I}_X, c), E)$ be a basic object, where $b$ is the maximal order of $\mathcal{I}_X$. Let $w \in X \subset W$ be a point. Any element with order 1 of*

$$\Delta^{b-1}(\mathcal{I}_X)_w \subset \mathcal{O}_{W,w},$$

*which satisfies conditions 3 and 4 of Definition 5.3.2, can be chosen as a hypersurface of maximal contact in a sufficiently small neighborhood of the point $w$.*

Afterwards, the auxiliary basic object can defined as follows:

**Definition 5.3.4.** Let $(W, (\mathcal{I}_X, c), E)$ be a basic object, where $b$ is the maximal order of $\mathcal{I}_X$. Let $U \subset W$ be an open set where a hypersurface $Z$ can be chosen

as hypersurface of maximal contact for $(W, (\mathcal{I}_X, c), E)$ like in Lemma 5.3.3. The auxiliary basic object $(Z, (\mathcal{I}_{\text{new}}, c), E_{\text{new}})$ on $U$ is defined as

$$\begin{aligned} \mathcal{I}_{\text{new}} &:= \text{Coeff}_Z(\mathcal{I}_X) = \sum_{i=0}^{b} (\Delta^i(\mathcal{I}_X)\mathcal{O}_Z)^{\frac{b!}{b-i}} \\ c &:= b! \\ E_{\text{new}} &:= \{E_i \cap Z \mid E_i \in E \text{ born after maximal order dropped to } b\}. \end{aligned}$$

**Remark 5.3.5.** In the literature there are different definitions of the coefficient ideal. The differences are the exponents. Here we use $\frac{b!}{b-i}$ to avoid rational constants. Sometimes there is also the exponent $\frac{b}{b-i}$ given, because it is related to projections of the newton polygon of the ideal.

**Example 5.3.6** ([16, Example 13.12])**.** Let $W = \mathbb{A}^3_{\mathbb{C}}$ and consider the ideal $J = \langle z^2 + x^3y^3 \rangle$. Then

$$\Delta(J) = \langle z^2 + x^3y^3, 2z, 3x^2y^3, 3x^2y^2 \rangle$$

and $\Delta^2(J) = \mathcal{O}_W$. Then

$$\text{Sing}(J, 2) = V(\Delta(J)) = V(x, z) \cup V(y, z).$$

If we choose $Z = V(z)$ and $\mathcal{A} = \langle x^3y^3 \rangle$ then

$$(W, (J, 2), \emptyset) \equiv (Z, (\mathcal{A}, 2), \emptyset).$$

Note, that $\mathcal{A} = \text{Coeff}_Z(J)$ is the coefficient ideal with respect to the hypersurface of maximal contact $Z$.

The advantage of the usage of the auxiliary basic object is that it is a basic object of a dimension one lower than the first basic object, so we can apply induction on the dimension here.

**Lemma 5.3.7** ([36, Lemma 58])**.** *Let $\mathcal{B} = (W, (\mathcal{I}_X, c), E)$ be a basic object and let $\mathcal{A} = (Z, (\mathcal{I}_{\text{new}}, c), E_{\text{new}})$ be an auxiliary basic object. Then the controlled transform with respect to the control c under a blow-up at a center determined by the governing function coincides with the auxiliary basic object constructed from the weak transform of $\mathcal{B}$ under the same blow-up using the strict transform of $Z$ as the hypersurface of maximal contact.*

We have seen that as soon as we find such a hypersurface $Z$ the computation of the coefficient ideal only involves determining $\Delta^i$ (Definition 2.4.9) of the ideal

which has previously been discussed and basic operations on ideals such as taking powers and sums.

Such a hypersurface $Z$ usually does not exist globally (not even in characteristic zero). In an implementation, the choice of the hypersurface involves passing to a suitable open covering such that on each open set $U_j$ is a hypersurface which can be used as $Z$ for each point $w \in U_j$.

The basic idea for finding such a covering is to consider $\Delta^{c-1}(\mathcal{I}_X)$.
Recall that $c$ is the maximal order, so the singular loci of the generators of $\Delta^{c-1}(\mathcal{I}_X)$ are empty. Furthermore, it is possible to express 1 as a combination of the generators of the ideals of these singular loci. We use the complement of those generators appearing with non-zero coefficients as the open covering.

When passing to an open covering, it can enlarge the number of charts which significantly slows down the resolution step. The first idea is to keep the number of open sets as low as possible and recombine them at the end. Unfortunately, the auxiliary objects depend on the chosen hypersurface although the resulting value of the governing function at each point is independent of this choice. So we continue with the algorithm for finding the maximal locus of this governing function in each of these open sets and combine those maximal loci. After passing to the closure and dropping components not meeting the open set $U_j$ each open set $U_j$ provides an ideal $\mathcal{I}_{Y_j}$ describing a candidate for the next center. Then the next center corresponds to the ideal $\bigcap_{j \text{ with maximal } v_j} \mathcal{I}_{Y_j}$.

The calculation of the center is illustrated in Algorithm 4 and Figure 5.2.

**Algorithm 4** center calculation of Bravo, Encinas and Villamayor

**INPUT:** $(W, (\mathcal{I}_X, b), E)$

1: Calculate the center $D = \mathrm{Max}(\mathrm{ord}_{\mathcal{I}_X}, N_E)$
2: **if** $D$ is not regular **then**
3: Calculate a covering $U = \cup_j U_j$ ▷ See Remark A.1.5
4: Centerlist $= \emptyset$
5: **for** $U_j \in U$ **do**
6: Centerlist = Centerlist + calculate_center$(U_j, (\mathrm{Coeff}_{U_j}, b!), E))$
7: D = glue(Centerlist)
8: return(D)

An interesting special case we should mention here is the so called monomial case. We will consider this case if the w$-$ord function is equal to zero at every

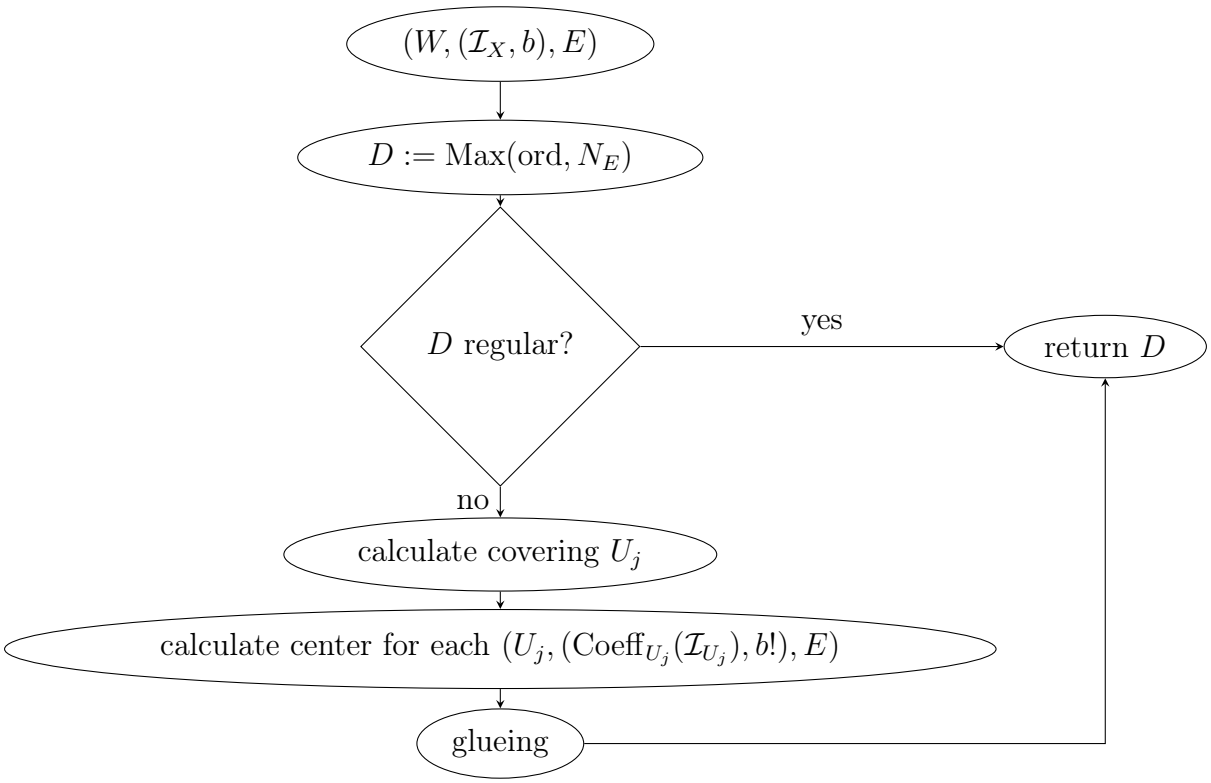



Figure 5.2.: Flow chart of the center calculation of Bravo, Encinas and Villamayor

point. The w$-$ord function for basic objects $(W,(J,b),E)$ equals the order of an idealistic exponent in Definition 2.3.24, where we consider the pair $(J,b)$. For this special case the resolution is defined by means of an upper-semi continuous function without making any induction on the dimension of the basic object. This resolution is purely combinatorial.

First, we have a look on why it is called monomial case.

**Remark 5.3.8** ([16, Remark 15.23(b)])**.** If $\max \mathrm{w-ord} = 0$ then $J_x = \mathcal{I}(H_1)_x^{\alpha_1(x)}\mathcal{I}(H_2)_x^{\alpha_2(x)}\cdots\mathcal{I}(H_k)_x^{\alpha_k(x)}$.

**Definition 5.3.9** ([16, Definition 20.1])**.** Let $(W,(J,b),E)$ be a basic object with $E = \{H_1,\ldots,H_r\}$. We say that the basic object is monomial if for any point $x \in \mathrm{Sing}(J,b)$

$$J_x = \mathcal{I}(H_1)_x^{\alpha_1(x)}\mathcal{I}(H_2)_x^{\alpha_2(x)}\cdots\mathcal{I}(H_k)_x^{\alpha_k(x)},$$

where $\alpha_i\colon H_i \cap \mathrm{Sing}(J,b) \to \mathbb{Z}$ is a locally constant function.

So the ideal $J$ of a monomial basic object is generated by a single monomial.

Now we define the considered upper-semi-continuous function. The maximal locus of this function will determine the center of the following blow-up.

**Definition 5.3.10** ([16, Definition 20.2])**.** Let $(W, (J, b), E)$ be a monomial basic object like in Definition 5.3.9. We define the function

$$\begin{aligned} h\colon \operatorname{Sing}(J,b) &\to \Gamma = \mathbb{Z} \times \mathbb{Q} \times \mathbb{Z}^{\mathbb{N}} \\ h(x) &= \Gamma(J,b)(x) := (-p(x), \omega(x), \ell(x)). \end{aligned}$$

If $x \in \operatorname{Sing}(J, b)$

$$p(x) = \min\{q \mid \exists i_1, \dots i_q \text{ such that } a_{i_1}(x) + \ldots + a_{i_q}(x) \geq b, x \in H_{i_1} \cap \ldots \cap H_{i_q}\},$$

$$\omega(x) = \max\left\{ \frac{a_{i_1}(x) + \ldots + a_{i_q}(x)}{b} \mid q = p(x), x \in H_{i_1} \cap \ldots \cap H_{i_q} \right\}$$

and

$$\begin{aligned} \ell(x) = \max_{<_{\text{lex}}} \Big\{ &(i_1, \dots, i_q, 0, \dots, 0) \mid q = p(x), \\ &\frac{a_{i_1}(x) + \ldots + a_{i_q}(x)}{b} = \omega(x), x \in H_{i_1} \cap \ldots \cap H_{i_q} \Big\}. \end{aligned}$$

Let $C_1, \dots, C_s$ be the irreducible components of $\operatorname{Sing}(J, b)$ at a point $x \in \operatorname{Sing}(J, b)$. Since we want to maximize $h(x)$ to obtain our center, we have to set the maximize $-p(x)$ in order to minimize the value of $p(x)$. This will indicate the minimal codimension of $C_1, \dots C_s$. We will denote this components with minimal codimension with $C'_1, \dots C'_{s'}$.
The second component is $\omega(x) = \frac{b'}{b}$ where $b'$ denotes the maximum order of $J$ along the sets $C'_1, \dots C'_{s'}$. We denote these components of order $\omega(x)$ by $C''_1, \dots C''_{s''}$. With the last component we choose a unique component containing $x$.

An example and an implementation of the center calculation of the monomial case is illustrated in Algorithm 33 and the subsequent example.

## 5.4. Resolution of binomial varieties in characteristic $p \geq 0$: Algorithm of Blanco and Encinas

From now on we leave the general case and we will take advantage of some structural properties of special singularities. In this section we handle the resolution of binomial singularities. Here we give more details than in the general case before. This belongs

to the fact that is was reimplemented for the use in the main algorithm of this thesis. This implementation is explained in Section A.3.

We present the algorithm of Blanco and Encinas, which first was presented in [9] and [10]. This algorithm provides a resolution of singularities for binomial varieties over a field of arbitrary characteristic. The algorithm deals with combinatorial centers of blow-up which preserve the binomial structure of the ideal after blow-up. Furthermore, this lets us ensure the existence of a hypersurface of maximal contact even in characteristic $p > 0$, so we can apply induction on the dimension of the ambient space and follow the main ideas and constructions of Villamayor and Hironaka with small changes.

It is the first algorithm for resolving singularities of binomial ideals without any restriction, i.e., that algorithm can also resolve singularities of monomial ideals and $p$-th powers in a field of characteristic $p > 0$.

Other algorithms for resolution of binomial ideals can only handle toric varieties which coincide with binomial ideals which are prime ideals or binomial ideals without any $p$-th powers, e.g., see the algorithm of Bierstone and Milman [8] for a characteristic-free resolution for reduced binomial ideals with no nilpotent elements and in particular toric ideals or the algorithm of González Pérez and Teissier [44].

While the algorithm of Bierstone and Milman deals with the Hilbert-Samuel function as resolution function, the presented algorithm deals with a modified order function which is called $E$-order.

In practice, the Hilbert-Samuel function is not easy to compute because we would need a stratification according to the initial monomials/forms of the local standard bases at all points (in the sense of the local rings or their completions at all points). So from a computational point of view the algorithm of Blanco and Encinas seems to be more practicable.

In this section let $K$ be an algebraic closed field of arbitrary characteristic and we denote by $\mathbf{W}$ the regular ambient space. At any stage of the resolution process we cover $\mathbf{W} = \cup_{i \in I} U_i$ by affine charts $U_i \cong \mathbb{A}^n_K$ and we will work locally inside these affine charts $U_i$ and consider an open set $W$. At any stage of the resolution proces we can glue charts together.

**Remark 5.4.1** ([9, Remark 1.1])**.** Let $E = \{V_1, \ldots, V_r\}$ be a simple normal crossing

divisor in $W$. $E$ definies a stratification of $W$ by regular closed sets

$$E_\Lambda := \bigcap_{\lambda \in \Lambda} V_\lambda, \text{ where } \Lambda \subseteq \{1, \ldots, r\}.$$

Then, by definition $E_\emptyset = W$ holds and each $E^0_\Lambda = E_\Lambda \backslash ((\bigcup_{j \notin \Lambda} V_j) \cap E_\Lambda)$ is locally closed and $W = \bigcup_\Lambda E^0_\Lambda$.

Hence, for all $\xi \in W$ there is a unique $\Lambda(\xi) \subseteq \{1, \ldots, r\}$ with $\xi \in E^0_{\Lambda(\xi)}$.

**Remark 5.4.2** ([9, Paragraph 1.3])**.** Let $J \subset K[\underline{x}] = K[x_1, \ldots, x_n]$ be a binomial ideal. We fix a monomial ordering in $K[\underline{x}]$ and compute the Gröbner basis of $J$. We know by [32, Proposition 1.1] that the reduced Gröbner basis of a binomial ideal is again generated by binomials.

**Definition 5.4.3** ([9, Definition 1.8])**.** Let $J \subset \mathcal{O}_W$ be a binomial ideal. The binomial equations of $J$ of the form $1 - \mu \underline{y}^{\underline{b}}$ with $\mu \in K$ and $\underline{b} \in \mathbb{Z}^n$ are called hyperbolic equations of $J$.

So we can start with a binomial ideal $J = \langle f_1(\underline{x}), \ldots, f_m(\underline{x}) \rangle \subset K[\underline{x}]$ such that the set of generators $\{f_1(\underline{x}), \ldots, f_m(\underline{x})\}$ is the reduced Gröbner basis of $J$.
After blow-up $W' \to W$ hyperbolic equations appear naturally in the transform of $J$. The points $\xi' \in W'$ outside the exceptional divisor where $1 - \mu \underline{x}^{\underline{b}}$ vanishes satisfies $\underline{x}^{\underline{b}}(\xi') \neq 0$.
We denote each variable $x_i$ that do not vanish anywhere over $V(J) \cap V(1 - \mu \underline{x}^{\underline{b}})$ as $y_i$. And afterwards we work in localized rings of the type $K[\underline{x}, \underline{y}]_{\underline{y}}$.

**Remark 5.4.4** ([9, Remark 1.4 and Remark 1.6])**.** At any stage of the resolution process, we cover $\mathbf{W} = \bigcup_{i \in I} U_i$ by affine charts $U_i \cong \mathbb{A}^n_K$. Inside any chart $U_i$ we consider the open set

$$W = \operatorname{Spec}(K[\underline{x}, \underline{y}]_{\underline{y}}) \subset \mathbb{A}^n_K.$$

The variables $y_i$ are invertible in the local ring $\mathcal{O}_{W,\xi}$, where

$$\xi \in V(1 - \mu \underline{y}^{\underline{b}}) \subset \bigcap_{\{i | B_i \neq 0\}} D(y_i).$$

**Definition 5.4.5** ([9, Definition 1.10])**.** Let $W = \operatorname{Spec}(K[\underline{x}, \underline{y}]_{\underline{y}})$ be the regular ambient space. Let $J$ be an ideal in $W$ and let $E = \{V_1, \ldots, V_r\}$ be a normal crossing divisor in $\mathbb{A}^n_K$. Let $\xi \in W$ be a closed point and let $\Lambda(\xi) \subset \{1, \ldots, n\}$ such that $\xi \in E^0_{\Lambda(\xi)}$. We define the $E$-order of $J_\xi$ in $\mathcal{O}_{W,\xi}$ to the order of the ideal with

respect to the $I(E^0_{\Lambda(\xi)})_\xi$-adic topology

$$E\text{-}\operatorname{ord}_{\mathcal{O}_{W,\xi}}(J_\xi) = \max\{m \in \mathbb{N} \mid J_\xi \subset (I(E^0_{\Lambda(\xi)})_\xi)^m\}.$$

**Definition 5.4.6** ([9, Definition 1.11])**.** Let $J \subset \mathcal{O}_W$ be a binomial ideal and let $\xi$ be a point. The $E$-order function associated to $J$ is

$$\begin{aligned} E\text{-}\operatorname{ord}_J \quad &: W \to \mathbb{N} \\ &\xi \mapsto E\text{-}\operatorname{ord}_J(\xi) := E\text{-}\operatorname{ord}_\xi(J) := E\text{-}\operatorname{ord}_{\mathcal{O}_{W,\xi}}(J_\xi). \end{aligned}$$

The $E$-order of $J$ computes the order of the ideal $J$ along $E \cap W = \{V(x_1), \ldots, V(x_s)\}$. Along the points in $E^0_{\Lambda(\xi)}$ the $E$-order function is constant.

Like in the Villamayor-setting (see Section 5.3), where we consider basic objects, we consider in the binomial setting so called binomial basic objects along $E$. First we define the affine version of this object before we are able to define the non affine version.

**Definition 5.4.7** ([9, Definition 1.15])**.** An affine binomial basic object along $E$ (short: BBOE) is a tuple $B = (W, (J, c), H, E)$ where

- $W = \operatorname{Spec}(K[\underline{x}, \underline{y}]_{\underline{y}}) \subset \mathbb{A}^n_K$,
- $E$ is a set of normal crossing regular hypersurfaces in $\mathbb{A}^n_K$ such that

  $$E = \{V(x_1), \ldots, V(x_s), V(y_1), \ldots, V(y_{n-s})\}.$$

  Moreover,

  $$E \cap \operatorname{Spec}(K[\underline{x}, \underline{y}]_{\underline{y}}) = \{V(x_1), \ldots, V(x_s)\}$$

  holds,
- $J$ is a binomial ideal and $c$ is a positive integer number and
- $H \subset E$ is a set of normal crossing regular hypersurfaces in $W$.

**Remark 5.4.8.** Note, that we have here an additional element $H$ in comparison to the basic objects of Section 5.3. We discuss the reason later in this section.

Now we define the non affine version.

**Definition 5.4.9** ([9, Definition 1.16])**.** A non affine binomial basic object along $E$ is a tuple $\boldsymbol{B} = (\boldsymbol{W}, (\mathcal{J}, c), H, E)$ which is covered by affine BBOEs where

- $\boldsymbol{W}$ is the regular ambient space over a field $K$ of arbitrary characteristic,
- $E$ is a set of normal crossing regular hypersurfaces in $\boldsymbol{W}$,
- $(\mathcal{J}, c)$ is a binomial pair, i.e., $\mathcal{J} \subset \mathcal{O}_{\boldsymbol{W}}$ is a coherent sheaf of binomial ideals with respect to $E$ satisfying $\mathcal{J}_\xi \neq 0$ for all $\xi \in \boldsymbol{W}$ and $c$ is a positive integer number and
- $H \subset E$ is a set of normal crossing regular hypersurfaces in $\boldsymbol{W}$.

Now we can define the considered $E$-singular locus. Along a normal crossing divisor $E$ it is analogous to the usual definition of the singular locus.

**Definition 5.4.10** ([9, Definition 1.18])**.** Let $J \subset \mathcal{O}_W$ be a binomial ideal and let $c$ be a positive integer. The $E$-singular locus of $J$ with respect to $c$ is

$$E\text{-}\operatorname{Sing}(J, c) = \{\xi \in W \mid E\text{-}\operatorname{ord}_\xi(J) \geq c\}.$$

Since the $E$-order is upper semi-continous, the level sets and in particular the $E$-singular locus is a closed subset of $W$.

We have seen the notion of idealistic exponents in Section 2.3.3 and there the definition of the singular locus does not depend on the generator of an idealistic exponent. So we have to show that the $E$-order is well-defined (Definition 5.4.5) in the class of idealisitc exponents. We have to define equivalence in this setting first.

**Definition 5.4.11** ([9, Definition 1.20])**.** Let $W$ be a regular scheme and let $J_1, J_2$ be coherent ideal sheafs in $W$ and let $c_1, c_2 \in \mathbb{N}_0$. The pair $(J_1, c_1)$ is equivalent to the pair $(J_2, c_2)$, if for every morphism $h\colon \operatorname{Spec}(A) \to W$

$$\frac{\nu(J_1 A)}{c_1} = \frac{\nu(J_2 A)}{c_2},$$

where $(A, \nu)$ is a valuation ring and $J_i A$ is the ideal generated by $J_i$ by means of $h$ in $A$ for $i = 1, 2$.

**Remark 5.4.12.** Let $B_1 = (W, (J_1, c_1), H, E)$ and $B_2 = (W, (J_2, c_2), H, E)$ be binomial basic objects along $E$ that define the same binomial basic object, then for every point $\xi \in \operatorname{Sing}(J, c) = \operatorname{Sing}(J_0, c_0)$

$$\frac{\operatorname{ord}_\xi(J_1)}{c_1} = \frac{\operatorname{ord}_\xi(J_2)}{c_2}$$

holds. So it suffices to consider only the pair $(J, c)$. For more details, see [34] or [56].

**Definition 5.4.13** ([9, Definition 1.22])**.** Let $W = \mathrm{Spec}(K[x_1, \ldots, x_s, y_1, \ldots, y_{n-s}]_{\underline{y}})$ .Let

$$(W, (J, c), H, E = \{V(x_1), \ldots, V(x_s), V(y_1), \ldots, V(y_{n-s})\} = \{V_1, \ldots, V_n\})$$

be an affine binomial basic object along $E$. Furthermore, let $H = \{H_1, \ldots, H_r\} \subset E$ be a normal crossing divisor with $H_i = V_j$ for $1 \leq j \leq s$ and $1 \leq i \leq r$. We define a transformation of the binomial basic object $(W, (J, c), H, E) \leftarrow (W', (J', c), H', E')$ by means of the blow-up $W \xleftarrow{\pi} W'$ in the center $Z$ contained in the $E$-$\mathrm{Sing}(J, c)$ with

- $W'$ is the strict transform of $W$,
- $H' = \{H'_1, \ldots H'_r, Y'\}$, with $H'_i$ is the strict transform of $H_i$ for all $1 \leq i \leq r$,
- $E' = \{V'_1, \ldots V'_n, Y'\}$, with $E'_i$ is the strict transform of $E_i$ for all $1 \leq i \leq n$,
- $J' = I(Y')^{\theta - c} \cdot J^*$ is the controlled transform of $J$ (with control $c$), where $J^*$ is the total transform of $J$ and

where $Y'$ is the exceptional divisor in $W'$ and $\theta = \max E\text{-}\,\mathrm{ord}(J)$.

A combinatorial center is given by an intersection of some coordinate hypersurfaces defined by some variables $x_i$.

**Definition 5.4.14** ([11, Definition 5.15])**.** A sequence of transformations of binomial basic objects

$$(W^{(0)}, (J^{(0)}, c), H^{(0)}, E^{(0)}) \leftarrow (W^{(1)}, (J^{(1)}, c), H^{(1)}, E^{(1)}) \leftarrow \cdots \leftarrow (W^{(N)}, (J^{(N)}, c), H^{(N)}, E^{(N)})$$

is called an $E$-resolution of $(W^{(0)}, (J^{(0)}, c), H^{(0)}, E^{(0)})$ or for shorthand an $E$-resolution of $(J^{(0)}, c)$, if $E$-$\mathrm{Sing}(J^{(N)}, c) = \emptyset$.

**Lemma 5.4.15** ([9, Corollary 2.5])**.** *Let $J \subset \mathcal{O}_W$ be a binomial ideal. Then*

$$E\text{-}\,\mathrm{ord}_\xi(J) \leq \mathrm{ord}_\xi(J),$$

*for every $\xi \in W$.*

*Proof.* See [9, Corollary 2.5]. □

**Lemma 5.4.16** ([9, Proposition 2.6]). *Let $J \subset \mathcal{O}_W$ be a binomial ideal and let $\xi \in W$. Then*

$$E\text{-}\operatorname{ord}_J \colon W \to (\mathbb{Z}, \leq), \xi \mapsto E\text{-}\operatorname{ord}_J(\xi) := E\text{-}\operatorname{ord}_\xi(J)$$

*is an upper semi-continuous function.*

*Proof.* See [9, Proposition 2.6]. □

We have seen in Section 5.2 and Example 5.2.8 that Hironaka's proof for the resolution of singularities fails in characteristic $p > 0$ in general because of the non-existance of hypersurfaces of maximal contact in positive characteristic. The $E$-order is a suitable resolution function since it can deal with the existence of hypersurfaces of maximal contact in positive characteristics in the binomial case.

**Definition 5.4.17** ([9, Definition 2.13]). Let $J \subset \mathcal{O}_W$ be a binomial ideal. Let $\xi \in W$ be a point such that $E\text{-}\operatorname{ord}_\xi(J) = \max E\text{-}\operatorname{ord}(J) = \theta$. A regular hypersurface $V$ is called hypersurface of maximal contact along $E$ (or shorthand: hypersurface of $E$-maximal contact) for $J$ at $\xi$

- $\xi \in V$,
- $E\text{-}\operatorname{Sing}(J, \theta) \subseteq V$ and their transforms under blow-up along a combinatorial center $Z \subseteq V$ also satisfy $E\text{-}\operatorname{Sing}(J', \theta) \subseteq V'$, where $J'$ denotes the controlled transform of $J$ and $V'$ denotes the strict transform of $V$.

Note, that a hypersurface of $E$-maximal contact fulfills the useful properties for binomials like a hypersurface of maximal contact in the characteristic zero case, i.e., such a hypersurface always exists and it makes the descend in dimension possible.

**Theorem 5.4.18** ([9, Corollary 2.17]). *Let $f(\underline{x}, \underline{y}) = \underline{y}^\gamma \underline{x}^\alpha - \mu \underline{x}^\beta \in K[\underline{x}, \underline{y}]_{\underline{y}}$ be a binomial equation. Let $\underline{a} \in \operatorname{Spec}(K[\underline{x}, \underline{y}]_{\underline{y}})$ be a point with $a_i = 0$ for all $i$ with $\alpha_i > 0$, $y^\gamma(a) \neq 0$ and $E\text{-}\operatorname{ord}_a(f) = |\alpha|$. If $|\alpha| > 0$ then*

$$E\text{-}\operatorname{Sing}(J, \theta) \subset \bigcap_{\{i | \alpha_i > 0\}} V(x_i).$$

*Proof.* See [9, Theorem 2.16] + [9, Corollary 2.17]. □

Hence, the hypersurface of maximal contact is always given by coordinate equations. That is the reason why the centers of the blow-ups are always combinatorial in this setting. It remains to show that these hypersurfaces always exist.

**Lemma 5.4.19.** *Let $B = (W, (J, c), H, E)$ be an affine binomial basic object along $E$. Then there is a hypersurface of maximal contact along $E$ for $J$.*

*Proof.* See [9, Remark 3.10, Lemma 3.11 and Lemma 3.31]. □

## 5.4.1. Induction on the dimension

Like in the Hironaka-setting we discussed in Section 5.2 we will apply induction of the dimension to the ambient space $W$. So given a BBOE (like for basic objects in the Villamayor-setting) we want to construct ideals $J_i$ defined in the local flags $W = W_n \supseteq W_{n-1} \supseteq \ldots \supseteq W_i \supseteq \ldots \supseteq W_1$ and the BBOEs $(W_i, (J_i, c_{i+1}), H_i, E_i)$ in dimension $i$ where each $E_i = W_i \cap E$.

If $E\text{-}\operatorname{Sing}(J_i, c_{i+1}) \neq \emptyset$, we factorize the ideal $J_i = M_i \cdot I_i$ such that $M_i$ is defined by a normal crossing divisor $D_i$ supported by the current exceptional locus.

We also need the companion ideal $P_i$.

**Definition 5.4.20** ([9, Definition 3.2])**.** Let $J_i = M_i \cdot I_i$ be an ideal at $\xi \in W_i$ in $W_i$. Let $\theta_i = E\text{-}\operatorname{ord}_\xi(I_i)$. Then the companion ideal $P_i$ of $J_i$ at $\xi$ with respect to the critical value $c_{i+1}$ satisfying $E\text{-}\operatorname{ord}_\xi(J_i) \geq c_{i+1}$ is the ideal

$$P_i = \begin{cases} I_i & , \text{ if } \theta_i \geq c_{i+1} \\ I_i + M_i^{\frac{\theta_i}{c_{i+1}-\theta_i}} & , \text{ if } 0 < \theta_i < c_{i+1}. \end{cases}$$

If $\theta_i = 0$ but $\operatorname{ord}_\xi(I_i) \neq 0$ then there are some hyperbolic equations in the ideal $I_i$. Therefore $I_i \neq 1$ but we will treat it like $I_i \equiv 1$ since the ideal $J_i$ behaves like a monomial ideal with respect to the $E$-order. So $J_i \equiv M_i$.

**Remark 5.4.21** ([9, Definition 3.6])**.** The critical value $c_i$ corresponding to the dimension $i-1$ is $c_i = E\text{-}\operatorname{ord}_\xi(P_i)$.

We have already seen the role of the coefficient ideal (Definition 5.3.4) in the characteristic zero setting. In the binomial setting we will use the coefficient ideal along $E$:

**Definition 5.4.22** ([9, Definition 3.7])**.** Let $P$ be an ideal in $W$, let $V \subset W$ be a regular hypersurface and let $\xi \in V$ be a point. Let $\{\underline{z}, \underline{w}\}$ be a regular system of parameters of $\mathcal{O}_{W,\xi}$ and let $\{\underline{w}\}$ be a regular system of parameters of $\mathcal{O}_{V,\xi}$ such that $V(\underline{z})$ defines $V$ in $W$. Let for all $f \in P$, $f = \sum_\alpha a_{f,\alpha}\underline{z}^\alpha$ be the Taylor expansion with

respect to the equation defining $V$ so that $a_{f,\alpha} \in \mathcal{O}_{V,\xi}$ and this equality holds after passing through the completion.
The coefficient ideal of $P$ along $E$ at $\xi$ with respect to $V$ is the ideal

$$E\text{-Coeff}_V(P) = \langle E\text{-Coeff}_V(f) \mid f \in P \rangle = \sum_{f \in P, |\alpha| < c} (a_{f,\alpha})^{\frac{c}{c-|\alpha|}},$$

where $c$ is the suitable critical value.

Note, that the computation of the coefficient ideal along $E$ commutes with the blow-up (see [9, Section 3.5] for more information).

**Remark 5.4.23** ([9, Remark 3.8])**.** The pair $(E\text{-Coeff}_V(P), c!)$ with integer exponents, i.e., take $\frac{c!}{c-|\alpha|}$ instead of $\frac{c}{c-|\alpha|}$ is equivalent to the pair $(E\text{-Coeff}_V(P), c)$ with rational exponents. So we can avoid rational exponents in Definition 5.4.22 like in Remark 5.3.5.

**Proposition 5.4.24** ([9, Proposition 3.9])**.** *Let $P$ be an ideal in $W$, let $V \subset W$ a regular hypersurface. Then*

$$E\text{-}\operatorname{ord}_\xi(E\text{-}\mathit{Coeff}_V(P)) \leq E\text{-}\operatorname{ord}_\xi(P)$$

*for every $\xi \in V$.*

*Proof.* See [9, Proposition 3.9]. □

**Remark 5.4.25.** The coefficient ideal of $P$ along $E$ has the same properties concerning the process of resolution of singularities (e.g. commutativity, stability,...) like the coefficient ideal which we discussed in Section 5.2. The proofs are the same or at least similar enough.

**Definition 5.4.26** ([9, Definition 3.14])**.** Let $P$ be an ideal in a hypersurface $V = V(x_1) \subset W = \operatorname{Spec}(K[\underline{x}, \underline{y}]_{\underline{y}})$. The ideal $P$ is called bold regular along $E$ or $E$-bold regular if $P$ is of the form $\langle \underline{y}^{\underline{\gamma}}(1 - \mu \underline{y}^{\underline{\delta}}) x_1^{\alpha_1} \rangle$ with $\mu \in K$, $\gamma, \delta \in \mathbb{Z}^n, \alpha_1 \in \mathbb{Z}_{>0}$.

Note, that $E\text{-}\operatorname{ord}(\underline{y}^{\underline{\gamma}}(1 - \mu \underline{y}^{\underline{\delta}})) = 0$ for all $\xi \in V \cap \{\eta \in W \mid \underline{y}^{\underline{\gamma}}(\eta) \neq 0, \underline{y}^{\underline{\delta}}(\eta) \neq 0\}$.

**Proposition 5.4.27** ([9, Proposition 3.16])**.** *Let $P \neq 0$ be an ideal in $W$. Let $V \subset W$ be a hypersurface of $E$-maximal contact for $P$ in a neighborhood of $\xi \in V$ and let $c = E\text{-}\operatorname{ord}_\xi(P)$. Then $E\text{-}\mathit{Coeff}_V(P) = 0$ if and only if $P$ is $E$-bold regular or 1.*

*Proof.* See [9, Proposition 3.16]. □

Now we can discuss how to apply the induction on dimension.

**Definition 5.4.28** ([9, Definition 3.17])**.** Let $P_i$ be an ideal in $W_i$. Let $V \subset W_i$ be a hypersurface of $E$-maximal contact for $P_i$ at $\xi \in V$. Let $c_i = E\text{-}\operatorname{ord}_\xi(P_i)$. The junior ideal of $P_i$ in $V$ is the ideal

$$J_{i-1} = \begin{cases} E\text{-}\operatorname{Coeff}_V(P_i) & \text{, if } E\text{-}\operatorname{Coeff}_V(P_i) \neq 0 \\ 1 & \text{, if } E\text{-}\operatorname{Coeff}_V(P_i) = 0. \end{cases}$$

By construction the junior ideal $J_i$ in $V$ can be expressed in terms of binomials, i.e., it is locally generated by binomials or powers of binomials. So we can apply induction on the dimension and we remain in the binomial setting.

Now we can discuss the procedure which constructs the center of the blow-up.

**Construction 5.4.29** ([9, Remark 3.19])**.**

- Choose a hypersurface of $E$-maximal contact for $P_i$ in $W_i$ (like in [9, Theorem 2.16]).
- Compute the junior ideal $J_{i-1}$ inside this hypersurface
- Set $Z = \bigcap_{i\in I} V(x_i)$ with $I \subseteq \{1, \ldots, n\}$.

**Definition 5.4.30** ([9, Definition 3.20])**.** Let $(W, (J, c), H, E)$ be a binomial basic object along $E$. For all points $\xi \in E\text{-}\operatorname{Sing}(J, c)$ we define the resolution function $E\text{-}\operatorname{inv}_{J,c}$ with its $n$ components with lexicographic order. It will be of one of the following types:

$$t(\xi) = \begin{cases} \left(\frac{E\text{-}\operatorname{ord}_\xi(I_n)}{c_{n+1}}, \frac{E\text{-}\operatorname{ord}_\xi(I_{n-1})}{c_n}, \ldots, \frac{E\text{-}\operatorname{ord}_\xi(I_{n-r})}{c_{n-r+1}}, \infty, \infty, \ldots, \infty\right) & (a) \\ \left(\frac{E\text{-}\operatorname{ord}_\xi(I_n)}{c_{n+1}}, \frac{E\text{-}\operatorname{ord}_\xi(I_{n-1})}{c_n}, \ldots, \frac{E\text{-}\operatorname{ord}_\xi(I_{n-r})}{c_{n-r+1}}, \Gamma(\xi), \infty, \ldots, \infty\right) & (b) \\ \left(\frac{E\text{-}\operatorname{ord}_\xi(I_n)}{c_{n+1}}, \frac{E\text{-}\operatorname{ord}_\xi(I_{n-1})}{c_n}, \ldots, \frac{E\text{-}\operatorname{ord}_\xi(I_{n-r})}{c_{n-r+1}}, \ldots, \frac{E\text{-}\operatorname{ord}_\xi(I_1)}{c_2}\right) & (c) \end{cases},$$

where $c_{i+1} = \max E\text{-}\operatorname{ord}(P_{i+1})$ is the critical value in dimension $i$.

(a) If $J_i = \langle 1 \rangle$, for $i < n$ we define $(t_{i-1}(\xi), \ldots, t_1(\xi)) = (\infty, \ldots, \infty)$ in order to preserve the number of components.

(b) If $E\text{-}\operatorname{ord}_\xi(I_i) = 0$ for some $i < n$ then we set $t_i(\xi) = \Gamma(\xi)$, where $\Gamma(\cdot)$ is the resolution function of the monomial case see [34] or the end of Section 5.3, and $t_j(\xi) = \infty$, for all $j = 1, \ldots, i-1$.

(c) Otherwise.

To compute the maximal value of $t$ it suffices to look at the points of maximal $E$-order at each dimension $i$. In this case $c_{i+1} = E\text{-}\operatorname{ord}_\xi(I_{i+1})$.

**Definition 5.4.31.** We define the maximal $E$-invariant locus as

$$E\text{-}\underline{\operatorname{Max}}(t) = \{\xi \in E\text{-}\operatorname{Sing}(J,c) \mid E\text{-}\operatorname{inv}_{(J,c)}(\xi) = \max t\},$$

where $\max t$ is the maximal value of $t$.

**Remark 5.4.32** ([9, Lemma 3.24])**.** $E$-$\underline{\operatorname{Max}}(t)$ is a closed set and $t : W \to (\mathcal{I}, \leq), \xi \mapsto t(\xi)$ is a upper semi-continous function, where $\mathcal{I}$ is a totally ordered set with lexicographical order, $\xi \in W$ is a point and $J \subset \mathcal{O}_W$ is a binomial ideal. Furthermore, the function $t$ drops lexicographically after blow-ups (see [9, Section 3.6]) and by construction, $E$-$\underline{\operatorname{Max}}(t) = Z = \bigcap_{i\in I} V(x_i)$ so it is the center of the next blow-up.

We now discuss the role of $H$ in the BBOE. It is necessary to take into account the exceptional divisors coming from the previous blow-ups and their transforms. When the higher coordinates of the $E$-resolution function remain constant under the blow-up the strict transforms of the hypersurface of $E$-maxinal contact from the previous step are again hypersurfaces of $E$-maximal contact. Then it is indispensable to consider these strict transforms at the same positions as their ancestors. So we have to determine the set of permissible hypersurfaces.

**Notation 1** ([9, 3.35])**.** Let $H_i^{(k)} \neq \emptyset$ be the exceptional divisor at dimension $i$ at the $k$-th step of the resolution process.
Let $\xi^{(k_0)} \in W^{(k_0)}, \xi^{(k-1)} \in W^{(k-1)}$ and $\xi^{(k)} \in W^{(k)}$ be points satisfying

$$\pi_k(\xi^{(k)}) = \xi^{k-1}, \ldots, \pi_{k_0+1}(\xi^{(k_0+1)}) = \xi^{(k_0)},$$

where the $E$-resolution function is maximal and $k_0 \leq k$.
If $t_i^{(k)} > 0$ let $k_0$ be the smallest index such that

$$\max t_i^{(k_0-1)} > \max t_i^{(k_0)} = \ldots = \max t_i^{(k)}.$$

**Definition 5.4.33** ([9, Definition 3.36])**.** Let $H_i^{(k)-}$ be the set of hypersurfaces of $H_i^{(k)}$ which are strict transforms of $H_i^{(k_0)}$ with the Notation 1. We set

$$H_i^{(k)} = H_i^{(k)+} \cup H_i^{(k)-}$$

where the union is disjoint. Elements in $H_i^{(k)-}$ are called permissible hypersurfaces in the sense that they are the only hypersurfaces of $E$-maximal contact that can

be considered in practice to apply induction on the dimension at this step of the $E$-resolution.

Note, that for $H = \emptyset$, every hypersurface of $E$-maximal contact is permissible.

### 5.4.2. Algorithm

Let $(W, (J, c), H, E)$ be a basic binomial object along $E$ and write $W = W_n$, where $\dim(W_n) = n$.

**Construction 5.4.34** ([9, Algorithm 4.4])**.** If $E\text{-}\operatorname{Sing}(J, c) \neq \emptyset$ and if we assume $(t_n, \ldots, t_{i+1})$ and $J_n, \ldots, J_{i+1}, J_i$ are given we can construct $t_i$ and $J_i$ as follows:

1. Factorize $J_i = M_i \cdot I_i$, where $M_i$ has support in $D_i$.
2. Compute $\max E\text{-}\operatorname{ord}(I_i)$ and let $\xi \in \underline{\operatorname{Max}} E\text{-}\operatorname{ord}(I_i)$ be a point.
   - If $E\text{-}\operatorname{ord}_\xi(I_i) > 0$, then $t_{i+1}(\xi) = \frac{E\text{-}\operatorname{ord}_\xi(I_i)}{E\text{-}\operatorname{ord}_\xi(P_{i+1})}$. Compute the companion ideal $P_i$ and go to step 3.
   - If $E\text{-}\operatorname{ord}_\xi(I_i) = 0$ for all $\xi \in W$ apply $\Gamma$ to $M_i$, then $(t_i(\xi), t_{i-1}(\xi), \ldots, t_1(\xi)) = (\Gamma(\xi), \infty, \ldots, \infty)$ and stop.
3. Set $c_i = \max E\text{-}\operatorname{ord}_\xi(P_i)$. Compute a hypersurface $V$ of $E$-maximal contact for $P_i$ in a neighborhoof of $\xi$.
4. Compute $E\text{-}\operatorname{Coeff}_V(P_i)$ with respect to $V \in H_i^-$ in order to construct $J_{i-1}$ in $W_{i-1}$ such that $\dim(W_{i-1}) = i - 1$.
   - If $E\text{-}\operatorname{Coeff}_V(P_i) = 0$, then $P_i$ is bold regular, $J_{i-1} = 1$ and $(t_{i-1}(\xi), \ldots, t_1(\xi)) = (\infty, \ldots, \infty)$, stop.
   - If $E\text{-}\operatorname{Coeff}_V(P_i) \neq 0$ set $J_{i-1} = E\text{-}\operatorname{Coeff}_V(P_i)$ and go to step 1.

Algorithm 5 and Figure 5.3 illustrate the induction on the dimension of the algorithm of Blanco and Encinas.

**Remark 5.4.35** ([9, Remark 4.5])**.** If $E\text{-}\operatorname{ord}_\xi(I) = 0$ for all $\xi \in W$ but there is a point $a \in W$ such that $\operatorname{ord}_a(I) \neq 0$ then there are hyperbolic equations in $I$.
In dimension $< n$ these ideals $I$ are considered as $I = 1$ and with respect to the $E$-order the ideal $J$ behaves like a monomial ideal.
In dimension $n$ we apply $\Gamma$ to $J_n$ and stop when $E\text{-}\operatorname{Sing}(J_n, c) = \emptyset$.

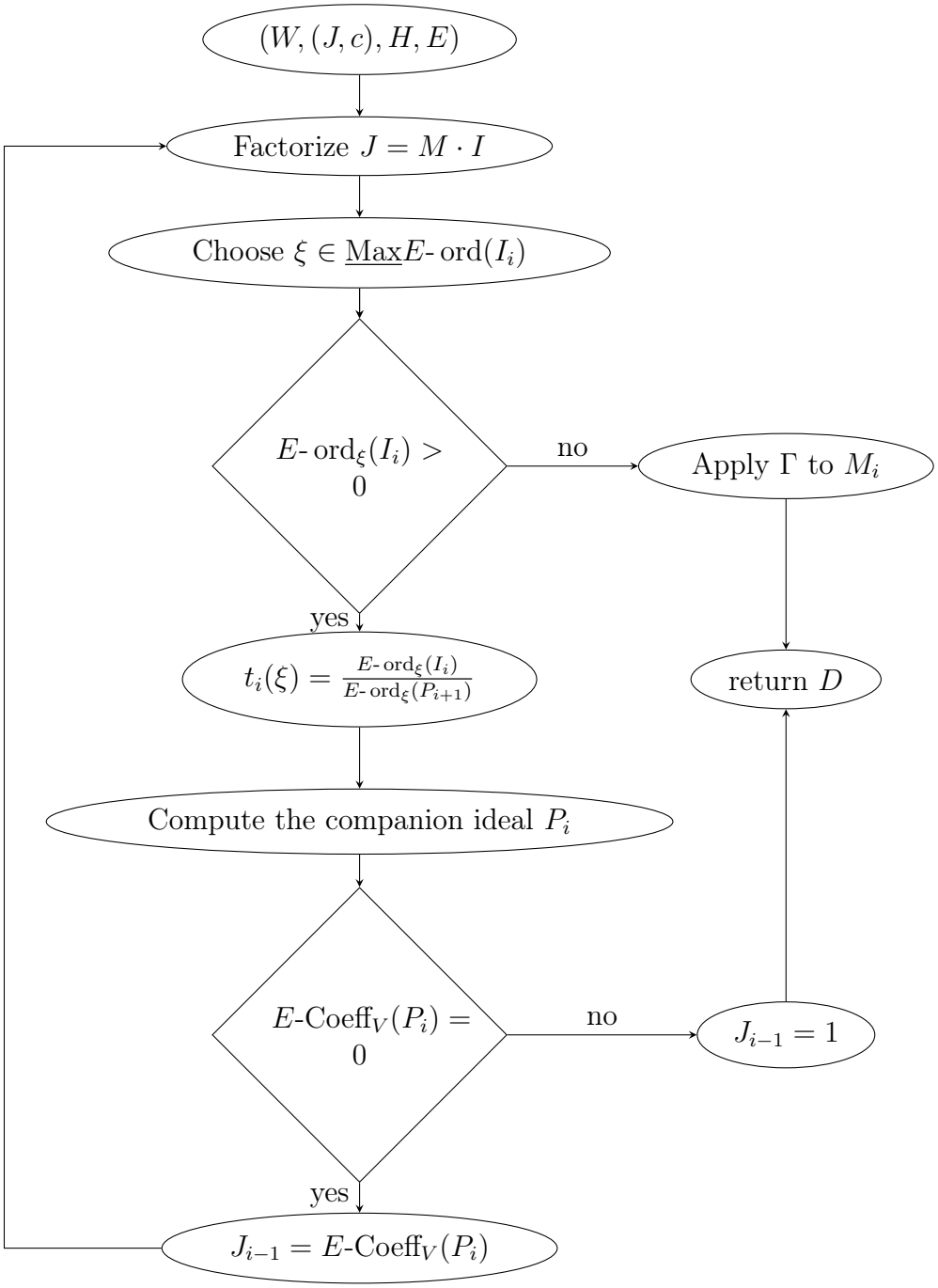



Figure 5.3.: Flow chart of the induction on dimension

**Algorithm 5** Algorithm of Blanco and Encinas for induction on the dimension

**INPUT:**
**OUTPUT:**

1: Factorize $J_i = M_i \cdot I_i$
2: Compute $\max E\text{-}\operatorname{ord}(I_i)$
3: Choose a point $\xi \in \underline{\operatorname{Max}} E\text{-}\operatorname{ord}(I_i)$
4: **if** $E\text{-}\operatorname{ord}_\xi(I_i) > 0$ **then**
5: $\quad t_i(\xi) = \frac{E\text{-}\operatorname{ord}_\xi(I_i)}{E\text{-}\operatorname{ord}_\xi(P_{i+1})}$
6: $\quad$ Compute the companion ideal $P_i$
7: **else if** $E\text{-}\operatorname{ord}_\xi(I_i) = 0$ for all $\xi \in W$ **then**
8: $\quad$ Apply $\Gamma$ to $M_i$ and stop.
9: **if** $E\text{-Coeff}_V(P_i) = 0$ **then**
10: $\quad J_{i-1} = 1$ and $(t_{i-1}(\xi), \ldots, t_1(\xi)) = (\infty, \ldots, \infty)$ ▷ $P_i$ is bold regular
11: **else if** $E\text{-Coeff}_V(P_i) \neq 0$ **then**
12: $\quad J_{i-1} = E\text{-Coeff}_V(P_i)$
13: $\quad$ Go to 1.

By blowing up only combinatorial centers we obtain a locally monomial ideal as output. We can apply some known algorithm to complete the resolution or alternatively, we can apply the same algorithm again (see Remark 5.4.39). The latter yields a log-resolution of binomial ideals and an embedded desingularization of the corresponding binomial variety.

Let $(W^{(0)}, (J^{(0)}, c), H^{(0)}, E^{(0)})$ be an affine binomial basic object and let $\{g_1, \ldots, g_m\}$ be a Gröbner basis of $J^{(0)}$ in $W^{(0)}$. After applying Construction 5.4.34, we have in each chart $E\text{-}\operatorname{Sing}(J^{(r)}, c) = \emptyset$ for some index $r$, where $J^{(r)} = J_n^{(r)} = M_n^{(r)} \cdot I_n^{(r)}$. The Algorithm terminates when $E\text{-}\operatorname{ord}(I_n^{(r)}) = 0$. If $I_n^{(r)} = 1$ the resolution process is finished. If $I_n^{(r)} \neq 1$, we have to modify the part of the singular locus which is included in the hyperbolic hypersurfaces which contain $V(I_n^{(r)})$.

Let $\xi \in W^{(r)}$ be a point. In a neighborhood of $\xi$ set $I_n^{(r)} = \langle f_1, \ldots, f_m \rangle$ where $f_j$ are the transforms of the generators of $g_j$ for $1 \leq j \leq m$ by the sequence of blow-ups. At $W^{(r)}$ we define

$$\tilde{I}_n^{(r)} = (\operatorname{Nhyp}(I_n^{(r)}))_{\underline{y}}(\langle f_i \mid 1 \leq i \leq m, E\text{-}\operatorname{ord}_\xi(f_i) \neq 0 \forall \xi \in W^{(r)} \rangle)_{\underline{y}} \subset K[\underline{x}, \underline{y}]_{\underline{y}},$$

where $\operatorname{Nhyp}(I_n^{(r)}))$ is the ideal generated by the non-hyperbolic generators of $I_n^{(r)}$. It is necessary to rename the variables $\underline{x}$ appearing in the hyperbolic equations of $I_n^{(r)}$ as $\underline{y}$ before passing through the localization. We have already seen this idea in the computation of the locus of refined order in Section 2.4.2. If the maximal $E$-order

of $J$ is one, the scheme corresponding to $J$ can locally be embedded into a regular hypersurface on an open set and the equation of this hypersurface can be added to the generators of the ideal of the ambient space. Then by construction $\tilde{I}_n^{(r)} \neq 0$ implies $E\text{-}\operatorname{ord}_\xi(\tilde{I}_n^{(r)}) > 0$ for all $\xi \in W^{(r)}$. Finally, we can resolve the binomial pair $(\tilde{I}_n^{(r)}, c)$ using Construction 5.4.34.

**Remark 5.4.36** ([10, Remark 2.2])**.** The construction of $\tilde{I}$ depends on the choice of the generators of $I$. Therefore we have to fix a Gröbner basis of $J^{(0)}$ at the beginning of the resolution process. In the toric case one can add the hyperbolic embedding dimension to the invariant. The algorithm is more or less the same. See [11] for more details.

**Construction 5.4.37** ([10, Algorithm 2.4])**.** Let $J \subset \mathcal{O}_W$ be a binomial ideal without hyperbolic equations with respect to a normal crossing divisor. We fix a reduced Gröbner basis of $J$ and consider $J = M \cdot I$.
At the beginning we assume $\mathcal{O}_W = K[\underline{x}]$, $E = \{V(x_1), \ldots, V(x_n)\}$, $H = \emptyset$ and $J = I$.

1. Apply Construction 5.4.34 to $(W, (J, \max E\text{-}\operatorname{ord}(J)), H, E)$ and obtain $J' = M' \cdot I'$ with $E\text{-}\operatorname{Sing}(J', c) = \emptyset$.

2. If $\max E\text{-}\operatorname{ord}(I') = 0$
   - If $I' = 1$ finish and $J'$ is principal.
   - If $I' \neq 1$, compute $\tilde{I}$ in $K[\underline{x}, \underline{y}]_{\underline{y}}$.
     - If $\tilde{I} \neq 0$, set $J = \tilde{I}$ and go to step 1.
     - If $\tilde{I} = 0$ finish and the ideal $I'$ is given by hyperbolic equations.

3. If $\max E\text{-}\operatorname{ord}(I') > 0$ set $J = J'$ and go to step 1.

Algorithm 6 and Figure 5.4 illustrate the principalization of the algorithm of Blanco and Encinas.

**Remark 5.4.38.** Construction 5.4.37 provides a principalization of a binomial ideal.

We still have to resolve the establish normal crossings.

**Remark 5.4.39** ([10, Remark 2.22])**.** To establish normal crossings we can apply:

1. The algorithm of Goward [45].

2. In characteristic 0: The algorithm of Villamayor (see Section 5.3) adapted to the case of an ideal generated by monomials which works over a field of arbitrary characteristic.

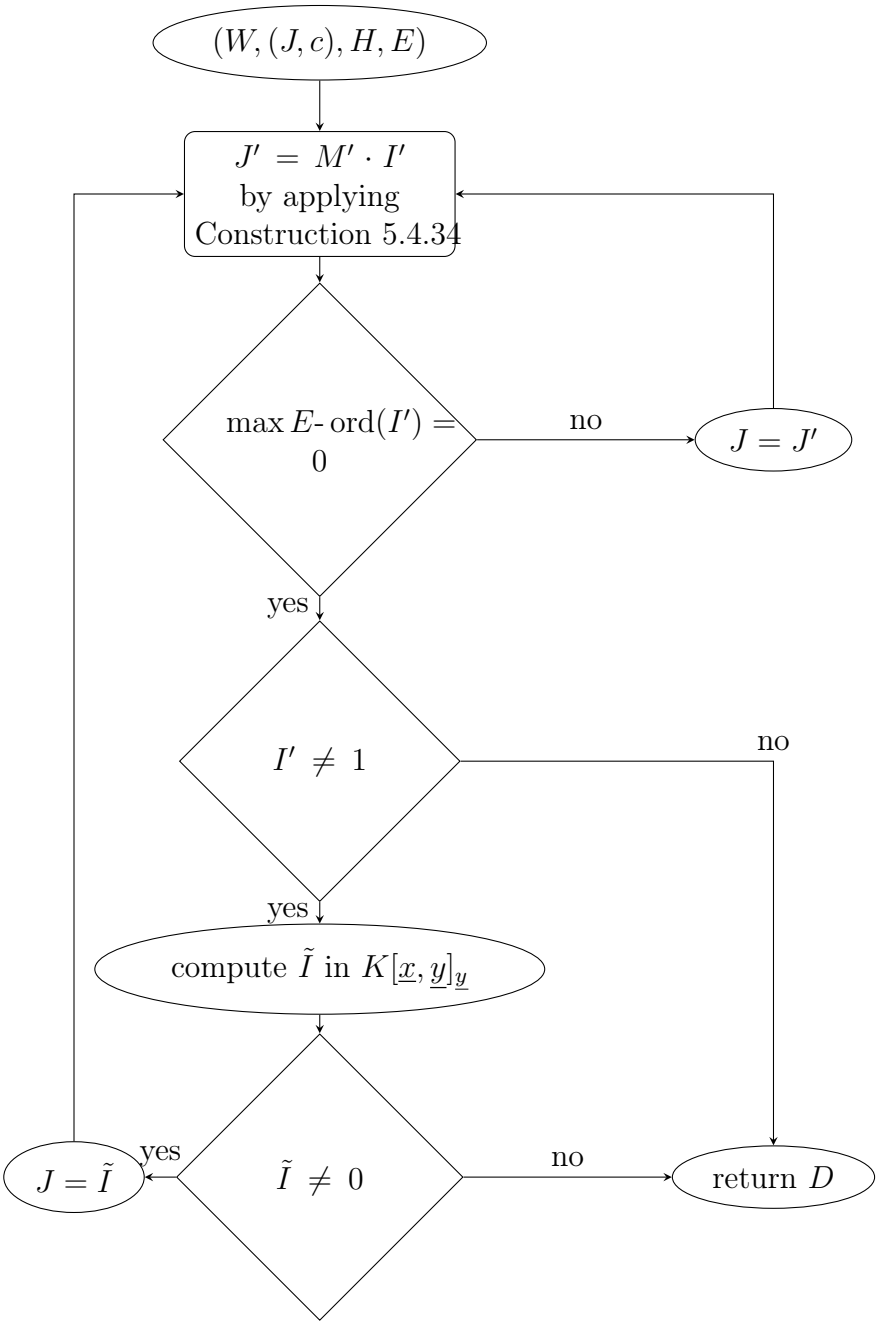



Figure 5.4.: Flow chart of the principalization of Blanco

**Algorithm 6** Algorithm of Blanco and Encinas (principalization)

**INPUT:** $(W, (J, c), H, E)$
**OUTPUT:**

1: $J' = M' \cdot I'$ ▷ Apply Algorithm 5; $E\text{-}\operatorname{Sing}(J', c) = \emptyset$
2: **if** $\max E\text{-}\operatorname{ord}(I') = 0$ **then**
3: **if** $I' = 1$ **then**
4: finish ▷ $J'$ is principal
5: **else if** $I' \neq 1$ **then**
6: compute $\tilde{I}$ in $K[\underline{x}, \underline{y}]_{\underline{y}}$
7: **if** $\tilde{I} \neq 0$ **then**
8: set $J = \tilde{I}$
9: Go to 1.
10: **else if** $\tilde{I} = 0$ **then**
11: finish ▷ $I'$ is given by hyperbolic equations.
12: **else if** $\max E\text{-}\operatorname{ord}(I') > 0$ **then**
13: set $J = J'$
14: Go to 1.

3. The algorithm of Bierstone and Milman [8] to resolve ideals generated by monomials.

4. Construction 5.4.34 of $E$-resolution of BBOE.

Another algorithm we can and will use in our main algorithm in Section 8 is the algorithm of Hu for simple arrangements (see Section 5.5).

**Remark 5.4.40** ([10, Remark 3.5 and Remark 3.6])**.** The several charts of affine BBOE glue together and form a unique resolution of singularities of a non affine BBOE.
Every center of the sequence of blow-up is compatible with the centers at other charts.
So the algorithm for resolution of binomial varieties of Blanco and Encinas is a global one.

## 5.5. Desingularization of arrangements of smooth subvarieties: Algorithm of Hu

The goal of this thesis is to resolve determinantal ideals for which there is a matrix with at most binomial entries. For this goal it is not enough to use the resolution of binomial ideals for a principalization step. We discuss this problem in Section 8.

We also need a procedure establishing normal crossings.

The algorithm of Hu is an alternative algorithm to the previously mentioned approaches in Remark 5.4.39.

The problem of forcing normal crossings to of several principalized hypersurfaces, is connected to the desingularization of arrangements of smooth subvarieties, which is discussed in [60].

**Definition 5.5.1.** An arrangement of smooth subvarieties of a nonsingular algebraic variety $X$ is a finite set $\mathcal{D} = \{D_i\}_i$ of closed irreducible subvarieties such that

1. $D_i$ is smooth;
2. $D_i$ and $D_j$ meet cleanly, i.e., the scheme-theoretic intersection $D_i \cap D_j$ is smooth and $T(D_i \cap D_j) = T(D_i) \cap T(D_j)$ holds for their tangent spaces;
3. $D_i \cap D_j = \emptyset$ or a disjoint union of $D_\ell$.

For our main resolution algorithm, we want to use the algorithm described in the theorem below:

**Theorem 5.5.2** ([60], Theorem 1.1)**.** *Let $X^0$ be an open subset of a nonsingular algebraic variety $X$. Assume that $X \backslash X^0$ can be decomposed into a union $\bigcup_{i\in I} D_i$ which is an arrangement of smooth subvarieties.*
*Then the set $\mathcal{D} = \{D_i\}_i$ is a partially ordered set.*
*Let $k$ be the rank of $D$. Then there is a sequence of well-defined blow-ups*

$$Bl_{\mathcal{D}}(X) \to Bl_{\mathcal{D}_{\leq k-1}}(X) \to \cdots \to Bl_{\mathcal{D}_{\leq 0}}(X) \to X,$$

*where $Bl_{\mathcal{D}_{\leq 0}}(X) \to X$ denotes the blow-up of $X$ along $D_i$ of rank $0$. Inductively $Bl_{\mathcal{D}_{\leq r}}(X) \to Bl_{\mathcal{D}_{\leq r-1}}(X)$ denotes the blow-up of $Bl_{\mathcal{D}_{\leq r-1}}(X)$ along the proper transforms of $D_j$ of rank $r$, such that*

1. *$Bl_{\mathcal{D}}(X)$ is smooth;*
2. *$Bl_{\mathcal{D}}(X \backslash X^0) = \bigcup_{i\in I} \tilde{D}_i$ is a divisor with normal crossings;*
3. *$\tilde{D_{i_1}} \cap \tilde{D_{i_2}} \cap \ldots \cap \tilde{D_{i_n}}$ is non-empty if and only if $\tilde{D_{i_1}} \ldots \tilde{D_{i_n}}$ form a chain in the partially ordered set $\mathcal{D}$. Consequently $\tilde{D}_i$ and $\tilde{D}_j$ meet cleanly if and only if $\tilde{D}_i$ and $\tilde{D}_j$ are comparable.*

Obviously, there are only finitely many $D_i$ so the following algorithm ends after finitely many steps. Complexity aspects concerning our case of use are discussed in Section A.4.

For the proof of Theorem 5.5.2, we need two additional Lemma.

**Lemma 5.5.3** ([60], Lemma 2.1)**.** *Let $U$ and $V$ be smooth closed subvarieties of a smooth variety $W$ such that $U$ and $V$ meet cleanly. Then*

1. *the proper transform of $U$ and $V$ in $B\ell_{U\cap V}(W)$ are disjoint.*
2. *the proper transformation of $V$ in $B\ell_U(W)$ is isomorphic to $B\ell_{U\cap V}(V)$.*
3. *if $Z$ is a smooth subvariety of $U \cap V$, then the proper transforms of $U$ and $V$ in $B\ell_Z(W)$ meet cleanly.*

**Lemma 5.5.4** (Flag Blow-up Lemma; [74])**.** *Let $V_0^1 \subset V_0^2 \subset \ldots \subset V_0^s \subset W$ be a flag of smooth subvarieties in a smooth algebraic variety $W_0$.*
*We define inductively for $k = 1, \ldots, s$:*
*$W_k$ is the blow-up of $W_{k-1}$ along $V_{k-1}^k$, $V_k^k$ is the exceptional divisor in $W_k$ and $V_k^i$ is the proper transform $V_{k-1}^i$ in $W_k$, for $k \neq i$.*
*Then the preimage of $V_0^s$ in the variety $W_s$ is a normal crossing divisor $V_s^1 \cup \ldots \cup V_s^s$.*

*Proof.* Let $\pi\colon B\ell_Z(W) \to W$ be a blow-up of a smooth algebraic variety $W$ along a smooth center $Z$, if $\tilde{V}$ is the proper transform of a smooth variety $V \supset Z$, then in terms of ideal sheaves

$$\mathcal{I}(\pi^{-1}(V)) = \mathcal{I}(\mathcal{V}) \cdot \mathcal{I}(E).$$

If we apply this in each step this yields

$$\mathcal{I}(\pi_s^1(V_0^s)) = \mathcal{I}(V_s^1) \times \ldots \times \mathcal{I}(V_s^s),$$

where $\pi_s\colon W_s \to W_0$ is the composition of the stated blow-ups. □

**Definition 5.5.5.** $\mathcal{D}_{<D_i}$ denotes the partially ordered subset of the elements less than $D_i$.

*Proof of Theorem 5.5.2.* Let $B\ell_{\mathcal{D}_{\leq 0}}(X)$ be the blow-up of $X$ along the disjoint smooth subvarieties of $D_i$ of rank 0 and let $D_j^1$ be the proper transform of $D_j$ of rank $\geq 1$. Lemma 5.5.3(1) shows that the proper transforms $D_j^1$ of $D_j$ of rank 1 are disjoint in $B\ell_{\mathcal{D}_{\leq 0}}$. Lemma 5.5.3(2) and (3) show that all $D_j^1$ are smooth and intersect cleanly or are disjoint. $D_i \cap D_j = \bigcup_\ell D_\ell$ implies $D_i^1 \cap D_j^1 = \bigcup_{\mathrm{rank}(D_\ell)>0} D_\ell^1$ otherwise $D_i \cap D_j = \emptyset$ and therefore $D_i^1 \cap D_j^1 = \emptyset$ holds.

Hence $\mathcal{D}^1 = \{D_j^1 = B\ell_{\mathcal{D}_{<D_j}}(D_j) \mid \operatorname{rank}(D_j) \geq 1\}$ is an arrangement of subvarieties in $B\ell_{\mathcal{D}_{\leq 0}}(X)$ and $\mathcal{D}^1_{\leq 0} = \{D_j^1 = B\ell_{\mathcal{D}_{<D_j}}(D_j) \mid \operatorname{rank}(D_j) = 1\}$.

We repeat the first step by computing the blow-up

$$B\ell_{\mathcal{D}_{\leq 1}}(X) = B\ell_{\mathcal{D}^1_{\leq 0}}(B\ell_{\mathcal{D}_{\leq 0}}(X)) \to B\ell_{\mathcal{D}_{\leq 0}}(X).$$

Let $\mathcal{D}^2$ be the proper transform of $D_j$ in $B\ell_{\mathcal{D}_{\leq 1}}(X)$ of rank $\geq 2$.
Analogous to the argumentation above, Lemma 5.5.3 shows that

$$\mathcal{D}^2 = \{D_j^1 = B\ell_{(\mathcal{D}_{<D_j})_{\leq 1}}(D_j) \mid \operatorname{rank}(D_j) \geq 2\}$$

is an arrangement of subvarieties and

$$\mathcal{D}^2_{\leq 0} = \{D_j^1 = B\ell_{(\mathcal{D}_{<D_j})_{\leq 1}}(D_j) \mid \operatorname{rank}(D_j) = 2\}.$$

Again, we repeat this step and we obtain

$$B\ell_{\mathcal{D}_{\leq 2}}(X) = B\ell_{\mathcal{D}^2_{\leq 0}}(B\ell_{\mathcal{D}_{\leq 1}}(X)) \to B\ell_{\mathcal{D}_{\leq 1}}(X).$$

We have $\operatorname{rank}(\mathcal{D}^2) = \operatorname{rank}(\mathcal{D}^1) - 1 = \operatorname{rank}(\mathcal{D}) - 2$.
We repeat this step until the subvarieties in the rank 0 are partially ordered sets $\mathcal{D}^k$ is blown up. Altogether we have the iterated blow-up along smooth disjoint centers

$$B\ell_{\mathcal{D}}(X) = B\ell_{\mathcal{D}^k_{\leq 0}}(B\ell_{\mathcal{D}^{k-1}_{\leq 0}}(\cdots B\ell_{\mathcal{D}^1_{\leq 0}}(B\ell_{\mathcal{D}_{\leq 0}}(X))\cdots)).$$

This proofs statement (1).
To show statement (3), we assume that $D_i \cap D_j \neq \emptyset$, i.e., that $D_i$ and $D_j$ are incomparable. Lemma 5.5.3(1) again, shows that their proper transform is disjoint at the stage $B\ell_{\mathcal{D}_{\leq r}}(X) \to X$ with $r := \max\{\operatorname{rank}(D_\ell) \mid D_\ell \subset D_i \cap D_j\}$. Hence, $\tilde{D_{i_1}} \cap \tilde{D_{i_2}} \cap \ldots \cap \tilde{D_{i_n}}$ is non-empty if and only if $D_{i_1} \ldots D_{i_n}$ form a chain in the partially orderet set $\mathcal{D}$.
To show statement (2), we use the Flag Blow-up Lemma (Lemma 5.5.4). We can apply it since the proof of statement (3) yields that for any chain $D_{i_1} < \ldots < D_{i_n}$ blow-up the proper transform of any $D_j$ which is not in the chain is irrelevant to the intersection $\tilde{D_{i_1}} \cap \tilde{D_{i_2}} \cap \ldots \cap \tilde{D_{i_n}}$. □

## 5.6. Resolution of surface singularities in arbitrary characteristic: Algorithm of Cossart, Jannsen and Saito

We also want to consider the CJS-Algorithm, which is originally stated in [20]. It is an algorithm for the embedding resolution and it is known, that it works correctly and terminates after finitely many steps in characteristic $0, p > 0$ and mixed characteristic, if the excellent noetherian scheme is of dimension 2. It is an open question if it resolves higher dimensional schemes, too. So the idea is to give an environment for testing and to find special examples in order to study this approach in an experimental manner.
The chosen upper semi-continuous function in the original variant is the Hilbert-Samuel function. The variant of Frühbis-Krüger, Ristau and Schober [35] replaces the measure function by a refinement of the order function (see Section 2.4.2). This is due to the fact that computing the locus of maximal order it is more efficient than the maximal Hilbert-Samuel stratum. Furthermore, their variant of the CJS algorithm is a generalization and also an embedded desingularization for arithmetic surfaces, so the ground field of characteristic zero in the original variant can be replaced by any field or principal Dedekind ring, e.g., $\mathbb{Z}$. Moreover, the approach of [35] in the arithmetic case makes a distinction between vertical and horizontal components. Both variants may lead to different resolution processes [35, Example 4.16].

We have already seen that it is standard technique in resolution of singularities to store the information on the exceptional divisor. While Hironaka-style resolutions are using basic objects the CJS algorithm uses boundarys to encode the information about it.

In this section, we assume that $X$ is a reduced excellent Noetherian scheme of dimension two, embedded in some excellent regular scheme $Z$.

**Definition 5.6.1** ([35, Definition 2.9])**.** A boundary on $Z$ is a set $\mathcal{B} = \{B_1, ..., B_s\}$ containing regular divisors on $Z$ such that $\operatorname{div}(\mathcal{B}) = B_1 \cup ... \cup B_s$ is a simple normal crossing divisor.

With this boundary we are able to define weakly permissible centers.

**Definition 5.6.2** ([35, Definition 2.9])**.** Let $\mathcal{B}$ be a boundary on $Z$. A closed subscheme $D \subset X$ is called $\mathcal{B}$-weakly permissible, if $D$ is regular contained in

Max-$\nu(X)$ and has transversal intersections with $\operatorname{div}(\mathcal{B})$.
We also say that the corresponding blow-up $\pi\colon Z' := B\ell_D(Z) \to Z$ with center $D$ is a $\mathcal{B}$-permissible blow-up.
The transform $\mathcal{B}'$ of the boundary $\mathcal{B}$ under $\pi$ is the union of the strict transforms of the components of $\mathcal{B}$ and the exceptional divisors $E'_\pi$ of $\pi$.

As already mentioned in the construction of the Hironaka invariant (Section 5.2) we have to divide the set of boundary components into two sets: The set of old components and the set of new components. Since we want to consider these old components in our calculation of center, too, we need a new definition of our order and the locus of our maximal order. The following example illustrates the reason why.

**Example 5.6.3** ([35, page 7])**.** Assume that the singular locus of a scheme is defined by $X := V(x^2 - y^2z^2)$. Then

$$\text{max-}\nu(X) = \underbrace{V(x,y)}_{=:L_1} \cup \underbrace{V(x,z)}_{=:L_2}.$$

There is a priori no way to distinguish them. We need to blow-up their intersection $V(x,y,z)$. In the $z$-chart, $V(x',z')$ is contained in the exceptional divisor and $V(x',y')$ is the strict transform of $L_1$. So with this idea, we can distinguish the lines and pick one of them as the next center. The CJS algorithm will choose the older component $L'_1$ in this case.

**Construction 5.6.4.** Let $\pi\colon Z' \to Z$ be a blow-up in a $\mathcal{B}$-weakly permissible center $D$. We denote by $X'$ and $\mathcal{B}' = \{B'_1, \ldots, B'_s, E'_\pi\}$ the strict transforms of $X$ and $\mathcal{B}$ under the blow-up $\pi$ where $E'_\pi$ denotes the exceptional divisor of $\pi$.

- If max-$\nu(X') <$ max-$\nu(X)$, the invariant has improved and we define all components of $\mathcal{B}'$ to be old.
- If max-$\nu(X') =$ max-$\nu(X)$, we define $E'_\pi$ to be a new component of the boundary $\mathcal{B}$.

Furthermore, we define the set of old components of the boundary as $O(X)$ and the set of new components is written as $N(X)$.

**Definition 5.6.5** ([35, Definition 2.10])**.** Let $\mathcal{B}$ be a boundary on $Z$. We define

$$\begin{aligned} \nu^O_{\text{ref}} = \nu^O_{\text{ref X,Z}} : X &\to (\mathbb{N}^3, \leq_{lex}), \\ x &\mapsto (\nu_{\text{ref}}(x), |O(X)|). \end{aligned}$$


Figure 5.5.: Horizontal and vertical components of the locus of maximal order in the arithmetic case

For $\tilde{A} \in \mathbb{N}^3$ we define

$$\mathcal{V}^O_{\geq \tilde{A}}(X) := \{x \in X \mid \nu^O_{\text{ref}}(x) \geq \tilde{A}\},$$

and

$$\max -\nu^O(X) := (\alpha, \delta, \sigma) := \max\{\nu^O_{\text{ref}}(x) \mid x \in X\}$$

and the locus of maximal log-refined order

$$\text{Max-}\nu^O(X) := \begin{cases} \mathcal{V}_{\geq \max -\nu^O(X)}(X), & \text{if } \delta > 1 \text{ or } \sigma > 0, \\ X, & \text{if } \delta = 1 \text{ or } \sigma = 0. \end{cases}$$

**Remark 5.6.6.** The function $\nu^O_{\text{ref}} \colon X \to (\mathbb{N}^3, \leq)$ is upper semi-continuous ([35, Proposition 2.12]).

When we consider the locus of maximal order in the arithmetic case, we are in the setting of Figure 5.5.

On the one hand, we have components of the locus of maximal order which can be seen in every prime. This corresponds to the horizontally aligned line in Figure 5.5. That is why we call such components horizontal components. On the other hand, there are components that can only be seen above a single prime. In the picture above, these are the components above 2 and 7. These lie vertically and are therefore called vertical components of the locus of maximal order

In the geometric case[1], we do not have these vertical components above the individual primes.

**Definition 5.6.7** ([35, Definition 2.13])**.** We distinguish the irreducible components of Max-$\nu^O(X)$ as follows:

- Geometric case: If $Z$ contains a field, then all irreducible components are defined to be vertical.
- Arithmetic case: Assume that $Z$ does not contain a field. Let $Y \subset$ Max-$\nu^O(X)$ be an irreducible component.
  - We say $Y$ is a horizontal component of Max-$\nu^O(X)$, if

$$Y \times_{\mathrm{Spec}(\mathbb{Z})} \mathrm{Spec}(\mathbb{Q}) \neq \emptyset.$$

  - We say $Y$ is a vertical component of Max-$\nu^O(X)$, if

$$Y \times_{\mathrm{Spec}(\mathbb{Z})} \mathrm{Spec}(\mathbb{Q}) = \emptyset.$$

We denote by Max-$\nu^O_{\mathrm{hor}}$ the set of horizontal components of Max-$\nu^O(X)$ and by Max-$\nu^O_{\mathrm{vertical}}$ the set of the vertical ones.

Such a prime like 2 or 7 in Figure 5.5 is called a bad prime:

**Definition 5.6.8** ([35, Notation 4.5])**.** Let $Z \subseteq \mathrm{Spec}(\mathbb{Z}[\underline{x}])$ be an equidimensional, regular closed subscheme and let $X \subset Z$ be a non-empty closed subscheme of $Z$ then we call a prime $p \in \mathbb{Z}$ bad prime for $X$, if the following conditions hold.

- Max-ord$(X)_{\mathrm{hor}} = \emptyset$
- there is an irreducible component $W \subseteq$ Max-ord$(X)$ such that $W \subset V(p)$.

If $p$ is not a bad prime, we say that $p$ is a good prime for $X$.

**Definition 5.6.9** ([35, Definition 2.15])**.** Let $\pi\colon Z' \to Z$ the blow-up with a regular irreducible center $D$. Let $Y' \subset Z'$ be an irreducible subscheme contained in the exceptional divisor $E_\pi$.
We say $Y'$ dominates $D$, if $\pi(Y')$ is dense in $D$.
In the case that $D$ is not irreducible, we say $Y'$ dominates $D = D_1 \cup \ldots \cup D_s$ with irreducible components $D_i$ for $1 \leq i \leq 1$, if there is an irreducible component $D_i$ which is dominated by $Y$.

[1]The case, where the ambient space $Z$ contains a field.

**Example 5.6.10.** Let $X = V(f_1, \ldots, f_r)$, where $f_i = \underline{x}^{A_i} - \rho \underline{x}^{B_i}$. We know:

- The centers, since being Max-$\nu^O(\cdot)$-components have the form $V(g_1, ..., g_t)$, where $g_\ell \in \{1 - \underline{x}^{B_\ell}, x_j - \underline{x}^{B_\ell}(\text{with } B_{\ell_j} = 0), \underline{x}\}$ and irreducible components of Max-$\nu^O(X)$ can be written as $V(g'_{i_1}, g'_{i_2}, \ldots, g'_{i_s})$, where the $g'_{i_j}$ are of the same form as the $g_i$.
- Exceptional divisors can be written as $V(x_i)$.

Let $Y' := \langle g'_{i_1}, g'_{i_2}, \ldots, g'_{i_s} \rangle$ be an ideal of an irreducible component of Max-$\nu^O(X')$. We assume that an irreducible component of the locus of maximal log-refined order of $X$ was $Y = V(g_1, \ldots, g_t)$. If there exists a $j$ with $i_j \notin \{1, \ldots, t\}$ such that $Y'$ dominates $Y$. Otherwise $Y'$ is not dominating $Y$.

The variant of [35] gives precedence to the horizontal components and considers the vertical components only when no more horizontal components are left. This idea goes back to Hironaka. Now we can state the algorithm in the variant of [35]:

**Construction 5.6.11** ([35, Construction 2.16])**.** Let $X$ be a reduced noetherian excellent scheme of dimension two, which is embedded in a regular scheme $Z$. Let $\mathcal{B}$ be a boundary on $Z$.

- Horizontal case:
  Let $Y_0 := \text{Max-}\nu^O_{\text{hor}}(X) \neq \emptyset$.
  If $X$ is regular and has only transversal crossings with the boundary, the resolution is finished.
  Since $X$ is reduced, we have $\dim(Y_0) < \dim(X)$, i.e., we can inductively find a sequence of $\mathcal{B}$ weak permissible blow-ups

$$\begin{aligned} Z = Z_0 \leftarrow \quad \cdots \leftarrow \quad Z_{m_0} =: Z' \\ \mathcal{B} = \mathcal{B}_0 \leftarrow \quad \cdots \leftarrow \quad \mathcal{B}_{m_0} =: \mathcal{B}' \end{aligned}$$

  such that the strict transform $Y'_0$ of $Y_0$ in $Z'$ $\mathcal{B}$ is a weak permissible center for $X_{m_0}$. We get this sequence of blow-ups by applying the CJS algorithm recursively with $Y_0$ instead of $X$. The center of the next blow-up is $D_{m_0} := Y'_0$.

$$\begin{aligned} Z' =: Z_{m_0} \quad \leftarrow Bl_{D_{m_0}}(Z_{m_0}) =: \quad Z_{m_0+1} \\ \mathcal{B}' =: \mathcal{B}_{m_0} \qquad\qquad\qquad\quad \mathcal{B}_{m_0+1} \end{aligned}$$

  Let $X_{m_0}$ be the strict transform of $X_0 := X$ (with respect to the whole sequence of blow-ups).
  We distinguish the following cases:

1. max-$\nu^O(X_{m_0+1}) <$ max-$\nu^O(X_0)$: The singularity has improved. The construction starts with the the new boundary (strict transforms of the exceptional divisors of the sequence of blow-ups) and $X_{m_0+1}$ plays the role of $X$.

2. max-$\nu^O(X_{m_0+1}) =$ max-$\nu^O(X_0)$: The singularity has not improved. Calculate a decomposition

$$Y_{m_0+1} := \text{Max-}\nu^O(X_{m_0+1}) = Y_{m_0+1}^{(0)} \cup Y_{m_0+1}^{(1)} \cup \ldots \cup Y_{m_0+1}^{(m_0+1)},$$

where

a) It holds for all $1 \leq i \leq m_0$ that $Y_{m_0+1}^{(i)}$ is an irreducible component of $Y_{m_0+1}$, which appears after the $i$-th blow-up at the first time, also called label $i$ components.

b) $Y_{m_0+1}^{(0)}$ are the irreducible components, which dominate the center $D_{m_0}$ of the last blowing up. These components inherit the label of $D_{m_0}$.

c) $Y_{m_0+1}^{(m_0+1)}$ are the irreducible components lying over $D_{m_0}$, but which do not dominate this center. This components are also called label $m_0+1$ components.

We set $k := \min\{j \in \{0, \ldots, m_0+1\} \mid Y_{m_0+1}^{(j)} \neq \emptyset\}$ and we start this construction again, where $Y_{m_0+1}^{(k)}$ plays the role of $Y_0$.

- Vertical case:
  Assume Max-$\nu^O_{\text{hor}}(X) = \emptyset$. Then we set $Y_0 :=$ Max-$\nu^O(X) =$ Max-$\nu_{\text{vertical}}(X)$ and proceed analogous to the horizontal case using Max-$\nu^O$ instead of Max-$\nu^O_{\text{hor}}$.

Figure 5.6 illustrates Construction 5.6.11.

More information on the flowchart and an implementation of max-$\nu^O$ (Section A.1.2) and of the CJS algorithm (see Section A.2) is found in the appendix.

**Theorem 5.6.12** ([35, Proposition 2.18])**.** *Let $X$ be a reduced noetherian scheme of dimension at most 2 embedded in a regular scheme $Z$. Suppose that one of the following conditions holds*

1. *$X$ has at most dimension 1, or*

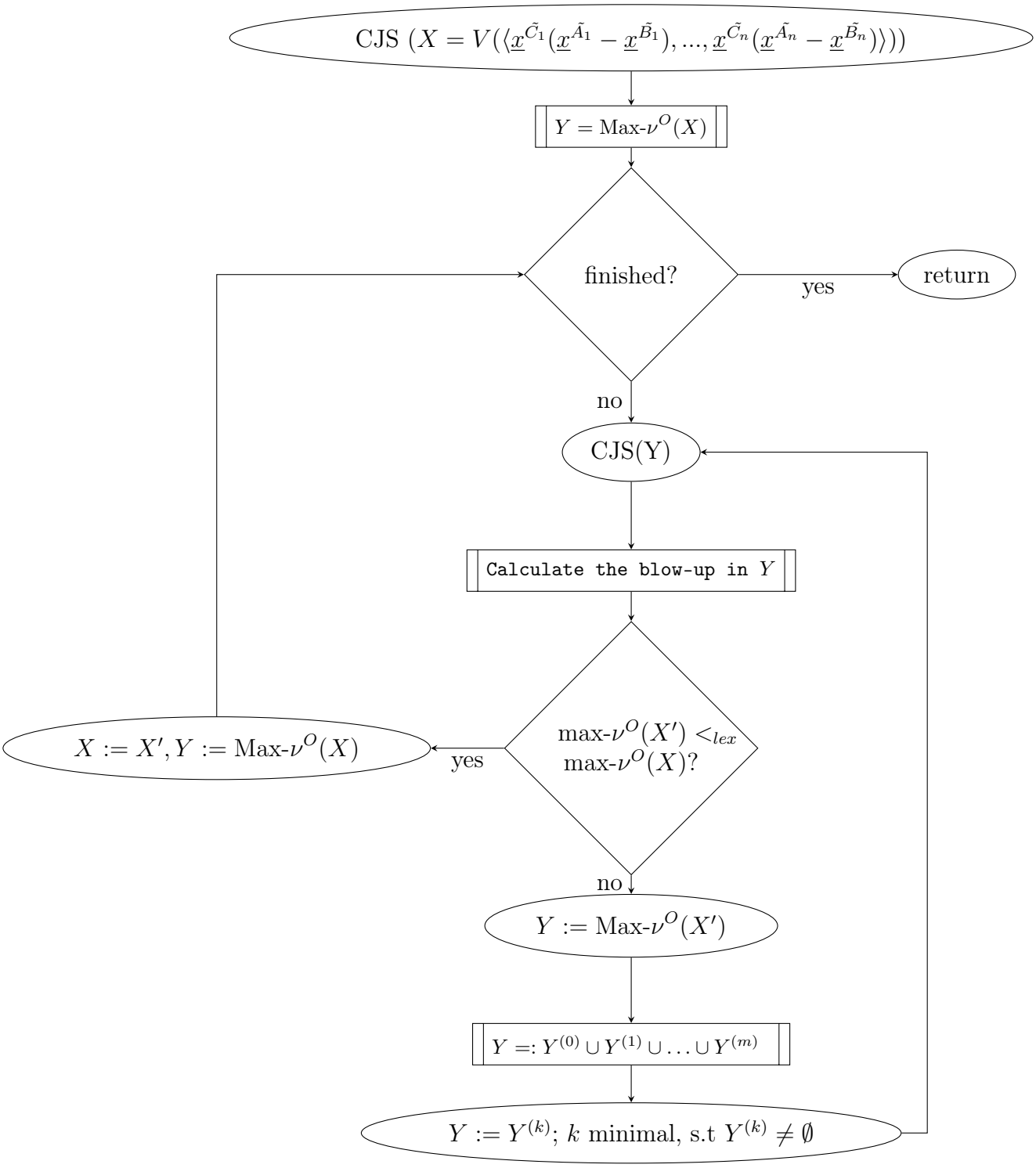



Figure 5.6.: Flow chart of the CJS algorithm

2. *$X$ has dimension 2 and there is a covering $Z = \bigcup_{i=1}^{m} U_i$ such that $X \cap U_i$ is isomorphic to a hypersurface.*

*Then the sequence of blow-ups in regular centers constructed in Construction 5.6.11 provides a desingularization of $X$. In particular, after finitely many blow-ups the strict transform is regular and transversal to the exceptional locus which is a normal crossing divisor.*

*Proof.* See [35, Proposition 2.18]. □

The formulation of the CJS algorithm allows arbitrary dimensional schemes instead of two-dimensionals. For both variants, it is an open question if the algorithm terminates or not, even in the characteristic zero case. The newly created components in the locus of maximal singularity, i.e., the label 1 or higher labeled components are well-behaved in the dimension two case but there is less control in a higher dimensional case. See [25, Example 2.6] for an example why one has to be careful in a higher dimensional case.
Another problem in higher dimension is to obtain a similar control on the characteristic polyhedron or an alternative invariant measuring for the improvement. For that reason, an implementation of the CJS algorithm is found in the appendix in Section A.2.

# 6. Determinantal Singularities

The class of determinantal singularities generalizes the class of complete intersections and has more structure than arbitrary singularities because of the underlying matrix structure. For references providing more on the background of determinantal singularities, we refer to the survey of Frühbis-Krüger and Zach [38], the book of Bruns and Vetter [18], or the book of Harris [48, Lecture 9]. Examples of more specialized approaches are the studies of symmetric determinantal singularities, which can be found in the articles of Gaffney and Molino [39] and [40] and the studies of skew-symmetric determinantal singularities by Bruce, Goryunov and Haslinger [17], for example.

In this section, we focus on resolution of generic skew-symmetric resp. generic symmetric determinantal singularities, generalizing results of [71] and [75]. An implementation of the presented algorithms is described in Section A.5.

A resolution of more general determinantal singularities, namely such determinantal singularities for which there exists a procedure which principalizes the entries (e.g. binomial entries), can be found in Chapter 8.

Throughout this chapter, we fix $m, r \in \mathbb{Z}_+$ with $r \leq m$. Let $R_0$ be a regular ring (e.g., $R_0 = \mathbb{C}, \mathbb{F}_q, \mathbb{Z}, \mathbb{Z}[T]_{\langle 2,T \rangle}, \ldots$) and let

$$R := R_0[x_{i,j} \mid i,j \in \{1,\ldots,m\}]$$

be the polynomial ring over $R$ with $m^2$ independent variables.

## 6.1. Generic case of determinantal singularities

First, we want to consider the approach of [71, Section 5], which we briefly illustrated in Chapter 4.

**Example 6.1.1.** [[71, Observation 5.3]] Let $R_0$ be a regular ring, $n \in \mathbb{Z}_+$ and let $R = R_0[x_{i,j} \mid 1 \leq i,j \leq n]$ be the polynomial ring in $n^2$ independent variables.
Let $M$ be a generic square matrix with homogeneous entries of degree 1. Without loss of generality, and maybe after some transformations of coordinates, we can

assume that $M = (x_{i,j})_{1 \leq i,j \leq n}$. Furthermore, we want to consider the determinantal singularity $V(f)$ generated by a single equation $f := \det(M)$.
We blow-up with center $V(x_{i,j} \mid 1 \leq i, j \leq n)$. For readability and by symmetry, we only consider the $X_{1,1}$-chart. The other charts are analogous and only the position of the 1-entry differs.
The strict transform $f'$ of $f$ is given by

$$f' = \det \begin{pmatrix} 1 & x'_{1,2} & \dots & x'_{1,n} \\ x'_{2,1} & x'_{2,2} & \dots & x'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{n,1} & x'_{n,2} & \dots & x'_{n,n} \end{pmatrix}.$$

Since the addition of rows does not change the determinant, we can add $-x_{1,j}$-times row 1 to row $j$, for $2 \leq j \leq n$. We get

$$f' = \det \begin{pmatrix} 1 & x'_{1,2} & \dots & x'_{1,n} \\ 0 & x'_{2,2} - x'_{2,1}x'_{1,2} & \dots & x'_{2,n} - x'_{2,1}x'_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x'_{n,2} - x'_{n,1}x'_{1,n} & \dots & x'_{n,n} - x'_{n,1}x'_{1,n} \end{pmatrix}.$$

By the Laplace Expansion of the determinant we get

$$f' = 1 \cdot \det \begin{pmatrix} x'_{2,2} - x'_{2,1}x'_{1,2} & \dots & x'_{2,n} - x'_{2,1}x'_{1,n} \\ \vdots & \ddots & \vdots \\ x'_{n,2} - x'_{n,1}x'_{1,n} & \dots & x'_{n,n} - x'_{n,1}x'_{1,n} \end{pmatrix}.$$

Since all variables are independent, all entries of the former matrix are simultaneously locally monomial and we can apply a change of variables to obtain variables $y_{i,j} := x_{i,j} - x_{i,1}x_{1,j}$, for $1 \leq i, j \leq n$.

$$f' = 1 \cdot \det \begin{pmatrix} y_{2,2} & \dots & y_{2,n} \\ \vdots & \ddots & \vdots \\ y_{2,n} & \dots & y_{n,n} \end{pmatrix}.$$

Although we perform a coordinate change, the blowing up globally glues together. See Construction 6.1.2 for more details.
By induction we can introduce new variables $z_{n-1,n-1}, z_{n-1,n}, z_{n,n-1}, z_{n,n}$, after $n-2$

blow-ups such that

$$f^{(n-2)} = \det \begin{pmatrix} z_{n-1,n-1} & z_{n-1,n} \\ z_{n,n-1} & z_{n,n} \end{pmatrix}.$$

And therefore

$$f' = z_{n-1,n-1}z_{n,n} - z_{n,n-1}z_{n-1,n}.$$

So we can reduce the problem of resolving a determinantal singularity of a square generic matrix to the problem of resolving a singularity generated by a single binomial.
This problem is solved in, e.g., [8], [11], [71] and [42]. But this special situation is so easy that it suffices to blow-up ones more in the origin to resolve the singularity.

Now we generalize this procedure to resolve generic matrices, which are not necessarily square matrices (see for example [71, Theorem 5.4]).

**Construction 6.1.2** (Resolution of determinantal singularities generated by generic matrices [71, Theorem 5.4])**.** Let $R_0$ be a regular ring. Let $r, k \in \mathbb{Z}_+$ and let $S = R_0[x_{i,j} \mid 1 \leq i \leq r, 1 \leq j \leq k]$ be the polynomial ring in $r \cdot k$ independent variables. Let $M_{r,k}$ be a generic matrix with homogeneous entries of degree 1. Without loss of generality, and maybe after some transformations of coordinates, we can assume that $M_{r,k} = (x_{i,j})_{1\leq i\leq r, 1\leq j\leq k}$. Furthermore, we want to consider the determinantal singularity $\mathcal{J}_{r,k,n}$ which is the ideal generated by the $n$-minors of $M$, where $n \leq \min\{r, k\}$,

$$\mathcal{J}_{r,k,n} = \langle f_{I,J} \mid I \subset \{1, \ldots, r\}, J \subset \{1, \ldots, k\}, \#I = \#J = n\rangle,$$

where $f_{I,J} = \det(M_{I,J})$ and $M_{I,J}$ denotes the $n \times n$ submatrix of $M_{r,k}$ determined by $I$ and $J$.
$f_{I,J}$ is homogeneous of degree $n$, every monomial in $f_{I,J}$ is a product of $n$ different variables and every $x_{i,j}$ such that $i \in I$ and $j \in J$ appears in $f_{I,J}$. Hence, $(f_{I,J})_{I,J}$ is a standard basis for $\mathcal{J}_{r,k,n}$ at the origin. See [66, Theorem 1] for more details.

After blowing up in $V(x_{i,j} \mid 1 \leq i \leq r, 1 \leq j \leq k)$, the ideal generated by the 1-minors of the strict transform $M'$ of $M$ equals $\langle 1\rangle$ in each chart (after a change of rows and columns), i.e.,

$$M' = \begin{pmatrix} 1 & x'_{1,2} & \ldots & x'_{1,k} \\ x'_{2,1} & x'_{2,2} & \ldots & x'_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{r,1} & x'_{r,2} & \ldots & x'_{r,k} \end{pmatrix}.$$

So the relevant ideal (lowest minor size such that the ideal generated by the $\ell$-minors does not equal $\langle 1\rangle$) is the ideal generated by the 2-minors. The ideals generated by $\ell$-minors for each $\ell = 2, \ldots, \max\{r, k\}$ do not change by applying elementary row operations[1].

By applying the elementary operations $\mathrm{row}_j \mapsto \mathrm{row}_j - x'_{j,1} \cdot \mathrm{row}_1$, for $2 \leq j \leq r$ and $\mathrm{col}_\ell \mapsto \mathrm{col}_\ell - x'_{1,\ell} \cdot \mathrm{col}_1$, for $2 \leq \ell \leq k$ we get

$$\tilde{M'} := \left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & x'_{2,2} - x'_{2,1}x'_{1,2} & \ldots & x'_{2,k} - x'_{2,1}x'_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x'_{r,2} - x'_{r,1}x'_{1,2} & \ldots & x'_{r,k} - x'_{r,1}x'_{1,k} \end{array}\right).$$

We call the highlighted $(r-1) \times (k-1)$-dimensional submatrix $M'_1$.

Obviously, the ideal generated by the $\ell$-minors of $M'$ is the same as the ideal generated by the $(\ell-1)$-minors of $M'_1$. By a coordinate change, which we can apply because the variables are independent, we can assume

$$M'_1 = \begin{pmatrix} y_{2,2} & \cdots & y_{2,k} \\ \vdots & \ddots & \vdots \\ y_{r,2} & \cdots & y_{r,k} \end{pmatrix},$$

where $y_{i,j} = x'_{i,j} - x'_{i,1}x'_{1,j}$. Hence $\langle y_{i,j} \mid 2 \leq i \leq r, 2 \leq j \leq k\rangle$ is the strict transform of the ideal $\mathcal{J}_{r,k,2}$ defined by the 2-minors of $M_{r,k}$. By a suitable choice of local coordinates, the strict transforms in the charts are global strict transforms of the ideal generated by the $\ell$-minors. The variety did not change, only the presentation has changed. We are now in the same situation as before blow-up $M$.

Like in the quadratic setting, we can blow-up in the center $V(y_{i,j} \mid 2 \leq i \leq r, 2 \leq j \leq k)$ and we can repeat this procedure until we are left with the 2-minors as maximal minors.
Then, we only have to resolve the ideal generated by the 2-minors of the strict transform of $M$.
The 2-minors of a generic matrix have binomial generators and we have multiple possibilities to resolve it, e.g., by the algorithm of Bierstone and Milman [8] or by the algorithm of Blanco and Encinas [9] and [10]. Another possibility is to continue the process described in Construction 6.1.2, namely blowing up with the ideal generated

[1] with exception of the multiplication of a scalar with a row, which we do not apply here

by the entries of the matrix as center and obtaining a one-entry in every chart, then repeating the row- and column transformations.

This result of Schober [71] generalized a result by Vainsencher [75] who proved this for generic determinantal singularities, if $R_0$ contains an algebraic closed field.

**Example 6.1.3.** Let

$$M = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \end{pmatrix}.$$

Let $I_j$ be the ideal generated by the $j$-minors of $M$. Then

$$I_1 = \langle x_{i,j} \mid 1 \leq i \leq 3, 1 \leq j \leq 4 \rangle \neq \langle 1 \rangle.$$

So our first center is $I_1$. Following Construction 6.1.2 after blow-up in $I_1$, we have

$$\begin{aligned} I_1' &= \langle 1 \rangle \\ I_2' &= \langle y_{i,j} \mid 2 \leq i \leq 3, 2 \leq j \leq 4 \rangle \neq \langle 1 \rangle. \end{aligned}$$

In the $X_{1,1}$-chart we have

$$M_1' = \begin{pmatrix} y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,2} & y_{3,3} & y_{3,4} \end{pmatrix}.$$

The situation in the other 11 charts is analogous. Hence it suffices to focus on the $X_{1,1}$-chart. There we are in the same situation as above. The next center is $I_2'$ the ideal generated by the strict transform of the 2-minors of $M$, which equals the ideal generated by the 1-minors of $M_1'$. So in each of the 12 charts of the first blow-up, we get 6 new charts by applying the second blow-up. In the $Y_{2,2}$-chart we have

$$M_2'' = \begin{pmatrix} z_{3,3} & z_{3,4} \end{pmatrix}$$

and $V(z_{3,3}, z_{3,4})$ is already resolved. The situation in the other charts is still analogous. Altogether we have $12 \cdot 6 = 72$ final charts and the whole process works with $12 \cdot 6 + 12 + 1 = 85$ charts. The 12 intermediate charts exist after the first blow-up and the initial chart is the additional chart in the end of the calculation. until $I_i''' = \langle 1 \rangle$ for all $i = 1, 2, 3$.

The whole complexity analysis of the exact number of total and final charts can be found in the appendix in Section B.2.

It is a natural question to ask whether a resolution exists for non-generic determinantal singularities, i.e., where the entries $x_{i,j}$ are replaced by arbitrary elements in $R$.

If $R_0$ contains a field of characteristic zero, there exists a resolution by Hironaka [54]. For constructive accounts, we refer to the previous Chapter 5. We allow $R_0$ to be of positive and mixed characteristics. Hence, the question on the existence of a desingularization is neither known nor clear, in general. A class of such determinantal singularities and how to desingularize them is found in Section 8. We provide an affirmative answer to this in the generic symmetric and generic skew-symmetric setup, both of which we introduce below in more details.

Furthermore, it is an interesting problem to investigate to what extent the determinantal structure resp. the geometry of the entries contribute to the singularity and its resolution.

## 6.2. (Skew-)Symmetric generic determinantal Singularities

In the present section, we focus on the case of symmetric resp. skew-symmetric generic determinantal singularities, where we consider the generic matrix with additional relations $x_{i,j} = x_{j,i}$ resp. $x_{i,j} = -x_{j,i}$, for all $i, j \in \{1, \dots, m\}$.

A more general approach for not necessarily generic entries in the matrix is presented in Chapter 8, where the determinantal structure playes an essential role.

Let us fix some notation in order to formulate our main result in the skew-symmetric case, where $M_m$ is replaced by the generic skew-symmetric matrix. First assume $\mathrm{char}(R_0) \neq 2$. Then $x_{i,j} = -x_{j,i}$ implies for $i = j$ that $x_{i,i} = 0$, for $i \in \{1, \dots, m\}$. Hence we introduce

$$I_{\mathrm{skew}} := \langle x_{i,j} + x_{j,i}, x_{k,k} \mid 1 \leq i < j \leq m, 1 \leq k \leq m \rangle$$

and define

$$S_{\mathrm{skew}} := R/I_{\mathrm{skew}}, \quad Z_{\mathrm{skew}} := \mathrm{Spec}(S_{\mathrm{skew}}), \quad \mathcal{I}^{\mathrm{skew}}_{m,r} := \mathcal{J}_{m,r} + I_{\mathrm{skew}}$$

as well as

$$Y^{\mathrm{skew}}_{m,r} := \mathrm{Spec}(R/\mathcal{I}^{\mathrm{skew}}_{m,r}) \subset Z_{\mathrm{skew}}.$$

In other words, if $\mathrm{char}(R_0) \neq 2$, then $Y^{\mathrm{skew}}_{m,r}$ is defined by the vanishing locus of the

$r$-minors of the matrix

$$A_m := \begin{pmatrix} 0 & x_{1,2} & \cdots & x_{1,m} \\ -x_{1,2} & 0 & & x_{2,m} \\ \vdots & & \ddots & \vdots \\ -x_{1,m} & -x_{2,m} & \cdots & 0 \end{pmatrix}. \tag{6.2.1}$$

By construction, $A_m$ is skew-symmetric, $A_m^T = -A_m$. Recall that $(\mathrm{pf}(A))^2 = \det(A)$ holds for any skewsymmetric square matrix $A$ and note that $\mathrm{pf}(B) = \det(B) = 0$ for any skewsymmetric square matrix $B$ of size $(2r+1) \times (2r+1)$ for $r \in \mathbb{N}_0$, where $\mathrm{pf}(B)$ denotes the pfaffian of the matrix $B$. Furthermore, [62, Theorem 3.2] tells us that $\sqrt{\langle 2\ell\text{-minors of } A_m\rangle} = \sqrt{\langle (2\ell-1)\text{-minors of } A_m\rangle}$, where $\sqrt{I}$ denotes the radical of an ideal $I$. That is the reason why we only need to consider the ideal of the $2\ell$-minors and not the $(2\ell-1)$- ones.

It is necessary to be careful and to consider the reduced strict transforms. For example, $Y^{\mathrm{skew}}_{2,2}$ is defined by the vanishing of $\det\begin{pmatrix} 0 & x_{1,2} \\ -x_{1,2} & 0 \end{pmatrix} = -x_{1,2}^2$. Hence, $Y^{\mathrm{skew}}_{2,2}$ is non reduced, which implies that no blowing up in a regular center improves the singularities, while on the other hand its reduction is regular. Note that we have $(Y^{\mathrm{skew}}_{2,2})_{\mathrm{red}} = Y^{\mathrm{skew}}_{2,1}$ and $\mathrm{pf}\begin{pmatrix} 0 & x_{1,2} \\ -x_{1,2} & 0 \end{pmatrix} = -x_{1,2} = Y^{\mathrm{skew}}_{2,2})_{\mathrm{red}}$.

Furthermore, since $A_m$ is skew-symmetric, we have for $m$ being an odd number such that $m = 2\ell+1$ for some $\ell \in \mathbb{Z}_+$,

$$\det(A_m) = \det(A_m^T) = \det(-A_m) = (-1)\det(A_m).$$

As we assume that $\mathrm{char}(R_0) \neq 2$, this yields that $\det(A_{2\ell+1}) \equiv 0$ is the zero polynomial and hence, $Y^{\mathrm{skew}}_{2\ell+1,2\ell+1} = Z_{\mathrm{skew}}$ is regular.

In conclusion, we do not need to consider $Y^{\mathrm{skew}}_{m,m}$ with $m = 2\ell+1$ and $\ell \in \mathbb{Z}_+$.

**Main Theorem 4.** *Let $m, \ell \in \mathbb{Z}_+$ with $2\ell \leq m$, let $R_0$ be a regular ring with $\mathrm{char}(R_0) \neq 2$, and let $A_m := (x_{i,j})_{i,j}$ be the generic skew-symmetric $m \times m$ matrix with entries in $R_0[x_{i,j} | 1 \leq i \leq j \leq m]$, i.e., $x_{i,j} = -x_{j,i}$ for all $i, j \in \{1, \ldots, m\}$.*

*The following sequence of blowing ups is an embedded resolution of singularities for the generic skew-symmetric determinantal singularity $Y^{\mathrm{skew}}_{m,2\ell} \subset Z_{\mathrm{skew}}$,*

$$Z_{\mathrm{skew}} =: Z_0 \xleftarrow{\pi_1} Z_1 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_{\ell-1}} Z_{\ell-1},$$

*where $\pi_i$ is the blowing up in the strict transform of $(Y^{\mathrm{skew}}_{m,2i})_{\mathrm{red}}$ in $Z_{i-1}$, for $1 \leq j \leq \ell - 1$.*

Note that we only have to blow-up minors of odd size for the desingularization since $\sqrt{\langle 2j\text{-minors of } A_m\rangle} = \sqrt{\langle (2j-1)\text{-minors of } A_m\rangle}$ holds for all $j = 2, 4, \ldots, m$.

The situation changes slightly if we move on to $\mathrm{char}(R_0) = 2$. The relation $x_{i,j} = -x_{i,j}$ becomes $x_{i,j} = x_{j,i}$. In particular, for $i = j$ the relation $x_{i,j} = x_{j,i}$ does not imply $x_{i,i} = 0$. So if $\mathrm{char}(R_0) = 2$, the skew-symmetric generic case naturally leads us to the generic symmetric case.

Let us introduce the symmetric setup. Let $R_0$ be a regular ring of arbitrary characteristic. We define

$$I_{\mathrm{sym}} := \langle x_{i,j} - x_{j,i} \mid 1 \leq i < j \leq m \rangle,$$

$$S_{\mathrm{sym}} := R/I_{\mathrm{sym}}, \quad Z_{\mathrm{sym}} := \mathrm{Spec}(S_{\mathrm{sym}}), \quad \mathcal{I}^{\mathrm{sym}}_{m,r} := \mathcal{J}_{m,r} + I_{\mathrm{sym}}$$

and

$$Y^{\mathrm{sym}}_{m,r} := \mathrm{Spec}(R/\mathcal{I}^{\mathrm{sym}}_{m,r}) \subset Z_{\mathrm{sym}}.$$

Therefore, $Y^{\mathrm{sym}}_{m,r}$ is given as the locus, where all $r$-minors of the following matrix $B_m$ vanish,

$$B_m := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{1,2} & x_{2,2} & & x_{2,m} \\ \vdots & & \ddots & \vdots \\ x_{1,m} & x_{2,m} & \cdots & x_{m,m} \end{pmatrix}. \tag{6.2.2}$$

**Main Theorem 5.** *Let $m, r \in \mathbb{Z}_+$ with $r \leq m$, let $R_0$ be a regular ring, and let $B_m := (x_{i,j})_{i,j}$ be the generic symmetric $m \times m$ matrix with entries in $R_0[x_{i,j} | 1 \leq i \leq j \leq m]$, i.e. $x_{i,j} = x_{j,i}$ for all $i, j \in \{1, \ldots, m\}$.*

*The following sequence of blowing ups is an embedded resolution of singularities for the generic symmetric determinantal singularity $Y^{\mathrm{sym}}_{m,r} \subset Z_{\mathrm{sym}}$,*

$$Z_{\mathrm{sym}} =: Z_1 \xleftarrow{\pi_1} Z_2 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_{r-1}} Z_r$$

*where $\pi_j$ is the blowing up in the strict transform of $Y^{\mathrm{sym}}_{m,j}$ in $Z_j$, for $1 \leq j \leq r-1$.*

The proofs are done by induction on the size $r \geq 1$ of the minors defining the determinantal singularity. In the case $r = 1$, there is nothing to show since $Y^{\mathrm{skew}}_{m,1}$ and $Y^{\mathrm{sym}}_{m,1}$ are regular. For $r > 1$, the key ingredient is to apply in each chart of a

blowing up elimination steps to the strict transform of $A_m$ resp. $B_m$ which preserve the (skew-)symmetric structure. Eventually, the determinantal structure allows to make a reduction from $r$ to $r-1$ resp. $r-2$ and from $m$ to $m-1$ resp. $m-2$.

Let us briefly summarize the content of the chapter: In Section 6.2.1, we discuss examples for $m \in \{2,3,4\}$ to visualize the idea and to provide the basis for our induction. In Section 6.2.2, we prove Theorem 4 and in Section 6.2.3, we show Theorem 5.

## 6.2.1. First Examples

First, we illustrate the ideas and appearing phenomena for some examples.

**Example 6.2.1** ($m = 2$)**.** Fix a regular basis ring $R_0$. Let us consider $Y^{\text{skew}}_{2,r}$ and $Y^{\text{sym}}_{2,r}$ for $r \in \{1,2\}$.

1. It is clear that both are regular for $r = 1$ since we assume $R_0$ to be regular.

2. Suppose that $\text{char}(R_0) \neq 2$. We have already seen in the introduction of this chapter that the reduction of $Y^{\text{skew}}_{2,2}$ is regular and hence no further blowing ups are required.

3. In the symmetric case, $Y^{\text{sym}}_{2,2}$ is a singular hypersurface given by the equation

$$\det \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{1,2} & x_{2,2} \end{pmatrix} = x_{1,1}x_{2,2} - x_{1,2}^2 = 0.$$

   The singular locus is equal to $Y^{\text{sym}}_{2,1}$ and by blowing up the center $D_1 = Y^{\text{sym}}_{2,1} = V(x_{1,1}, x_{1,2,}, x_{2,2})$, we obtain an embedded desingularization.

Therefore, we have verified Theorems 4 and 5 for the special case $m = 2$.

In general, $Y^{\text{skew}}_{m,r}$ and $Y^{\text{sym}}_{m,r}$ are not hypersurfaces if $r < m$. In order to be able to control them after a blowing up, we have to make a particular choice for the system of generators for their respective ideal.

**Remark 6.2.2.**

1. Applying the same arguments as in [55, Corollary (2.21,d), p. 270] it can be shown that a standard basis for $I$ generates $I$.

2. Set $X := \text{Spec}(R/I)$. Suppose $D := \text{Spec}(R/P)$ is regular and that $D \subset \text{Sing}(X)$ is contained in the singular locus of $X$. Let $(f)$ be a standard basis for $I$ with respect to $P \subset R$. Blow-up with center $D$ and consider an affine

chart, as in Remark 2.2.4. By [20, Theorem 9.1] the strict transform $X'$ of $X$ is determined as the vanishing locus of the strict transforms $(f')$ of $(f)$.

In general, the last conclusion is not true if we take any system of generators. For example, consider $R = K[x,y,z]_{\langle x,y,z\rangle}$, $P = \langle x,y,z\rangle \subset R$, and the ideal $I = \langle x^2 - y^3, x^2 - z^5\rangle \subset R$, where $K$ is any field. Clearly, $(g_1, g_2) := (x^2 - y^3, x^2 - z^5)$ is not a standard basis with respect to $P$ since the initial form of $h := y^3 - z^5 = g_2 - g_1$ is not contained in $\langle \mathrm{in}_P(g_1), \mathrm{in}_P(g_2)\rangle$. If we blow-up the origin $D = \mathrm{Spec}(R/P)$ and consider the $Z$-Chart, we get the strict transforms $g_1' = x'^2 - y'^3 z'$, $g_2' = x'^2 - z'^3$, $h' = y'^3 - z'^2$. Note that $g_2' - g_1' = z'(y'^3 - z'^2)$. Let $I'$ be the ideal generated by the strict transforms of all $g \in I$. We see that $h' \in I'$, but $h' \notin \langle g_1', g_2'\rangle$. Hence, $I' \neq \langle g_1', g_2'\rangle$.

In order to discuss the relevant examples of standard bases in our context, we have to introduce the following notation.

**Definition 6.2.3.** Let $r, m \in \mathbb{Z}_+$ with $r \leq m$. Let $R$ be a regular ring and let $M = (m_{i,j})$ be a $m \times m$ matrix with entries in $R$. For the symmetric case, we define $\mathcal{I}_r^{\mathrm{sym}}(M) \subseteq R$ to be the ideal generated by the $r$-minors of a symmetric matrix $M$, i.e.,

$$\mathcal{I}_r^{\mathrm{sym}}(M) = \langle \det(M_{I,J}) \mid I, J \subset \{1, \ldots, m\} : \#I = \#J = r\rangle,$$

where $M_{I,J}$ denotes the $r \times r$ submatrix of $M$ given by $(m_{i,j})_{i\in I, j\in J}$.

For the skewsymmetric case, we define $\mathcal{I}_r^{\mathrm{skew}}(M) \subseteq R$ to be the ideal generated by the radical of the $r$-minors of a skewsymmetric matrix $M$, i.e.,

$$\mathcal{I}_r^{\mathrm{skew}}(M) = \sqrt{\langle \det(M_{I,J}) \mid I, J \subset \{1, \ldots, m\} : \#I = \#J = r\rangle},$$

where $r = 2k$ for $0 \leq k \leq \lfloor \frac{m}{2} \rfloor$.

We fix a regular ring $R_0$ for the rest of the section.

**Example 6.2.4.** Let $r, m \in \mathbb{Z}_+$ with $r \leq m$ and consider the matrices $A_m$ and $B_m$ introduced in (6.2.1) and (6.2.2), respectively.

1. Set $\mathcal{I}_{m,r}^{\mathrm{skew}} := \mathcal{I}_r^{\mathrm{skew}}(A_m)$ and let $f_{I,J}^{\mathrm{skew}} := \sqrt{\det(A_{I,J})}$. We have:
   a) $\mathcal{I}_{m,r}^{\mathrm{skew}} = \langle f_{I,J}^{\mathrm{skew}} \mid I, J \subset \{1, \ldots, r\}, \#I = \#J = r\}$;
   b) each $f_{I,J}^{\mathrm{skew}}$ is homogeneous of degree $r$;
   c) every monomial appearing in $f_{I,J}^{\mathrm{skew}}$ is a product of $r$ variables and every $x_{i,j}$ appears in $f_{I,J}^{\mathrm{skew}}$ for $i \in I$ and $j \in J$.

It follows from [53, Theorem 5.1] that $(f^{\text{skew}}_{I,J} \mid I, J \subset \{1,\dots,r\}, \#I = \#J = r)$ is a Gröbner basis and because of being homogenous this is also a standard basis in the Hironaka sense for $\mathcal{I}^{\text{skew}}_{m,r}$ at the maximal ideal $\langle x_{i,j} \mid 1 \leq i,j \leq m\rangle$.

2. In analogue to (1), for the symmetric case we set $\mathcal{I}^{\text{sym}}_{m,r} := \mathcal{I}^{\text{sym}}_r(B_m)$ and $f^{\text{sym}}_{I,J} := \det(B_{I,J})$.

   a) $\mathcal{I}^{\text{sym}}_{m,r} = \langle f^{\text{sym}}_{I,J} \mid I, J \subset \{1,\dots,r\}, \#I = \#J = r\}$

   b) the $f^{\text{sym}}_{I,J}$ are homogeneous of degree $r$.

   c) Let $m$ be a monomial appearing in $f_j \in f^{\text{sym}}_{I,J}$, then $m$ is a product of $r$ variables. And for all variables $x_{i,j}$ there are at least one and at most two such $f^{\text{sym}}_{I,J}$ where it appears. With [19, Theorem 2.9] we can argue analogous to (1), that $\{f^{\text{sym}}_{I,J} \mid I, J \subset \{1,\dots,r\}, \#I = \#J = r\}$ is a standard base for $\mathcal{I}^{\text{sym}}_r$ at the maximal ideal $\langle x_{i,j} \mid 1 \leq i,j \leq m\rangle$.

In the next example, we illustrate how we preserve the symmetry by appropriate row and column operations.

**Example 6.2.5** ($m = 3$, symmetric)**.** Consider $Y^{\text{sym}}_{3,3}$, which is determined by the vanishing of the determinant of $B_3$,

$$B_3 = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{1,2} & x_{2,2} & x_{2,3} \\ x_{1,3} & x_{2,3} & x_{3,3} \end{pmatrix}.$$

The first center for blowing up proposed by Theorem 5 is

$$Y^{\text{sym}}_{3,1} = \operatorname{Spec}(S/\langle x_{1,1}, x_{1,2}, \dots, x_{3,3}\rangle).$$

First, we consider the $X_{1,1}$-chart, i.e., as described in Remark 2.2.4, we have $x_{1,1} = x'_{1,1}$ and $x_{i,j} = x'_{1,1}x'_{i,j}$ for $(i,j) \neq (1,1)$. The strict transform of $Y_{3,3}$ is defined by the vanishing locus of the determinant of the strict transform of $B_3$,

$$B'_3 := \begin{pmatrix} 1 & x'_{1,2} & x'_{1,3} \\ x'_{1,2} & x'_{2,2} & x'_{2,3} \\ x'_{1,3} & x'_{2,3} & x'_{3,3} \end{pmatrix}.$$

We perform elementary row and column operations to eliminate all entries in the first row and column expect for the 1 at position $(1,1)$. Since these do not change

the determinant, we have

$$\det(B_3') = \det\begin{pmatrix} 1 & 0 & 0 \\ 0 & x'_{2,2} - x'^2_{1,2} & x'_{2,3} - x'_{1,2}x'_{1,3} \\ 0 & x'_{2,3} - x'_{1,2}x'_{1,3} & x'_{3,3} - x'^2_{1,3} \end{pmatrix} = \det\begin{pmatrix} y_{2,2} & y_{2,3} \\ y_{2,3} & y_{3,3} \end{pmatrix},$$

where $y_{2,2} := x'_{2,2} - x'^2_{1,2}$, $y_{2,3} := x'_{2,3} - x'_{1,2}x'_{1,3}$, $y_{3,3} := x'_{3,3} - x'^2_{1,3}$. Recall that the exceptional divisor of the first blowing up is $\operatorname{div}(x'_{1,1})$ and $x'_{1,1}$ is transveral to $y_{2,2}, y_{2,3}, y_{3,3}$.

We have reduced the problem to $r = m = 2$ and the ideal of the upcoming center is $I := \langle y_{2,2}, y_{2,3}, y_{3,3}\rangle$. We leave it to the reader to verify that the singularities are locally resolved after the blowing up.

We claim that the ideal $I$ provides a global center. Indeed, we observe that $y_{i,j}$ is the strict transform of the 2-minor of $B_3$ that contains $x_{1,1}$ and $x_{i,j}$,

$$\det\begin{pmatrix} x_{1,1} & x_{1,j} \\ x_{1,i} & x_{i,j} \end{pmatrix} = x_{1,1}x_{i,j} - x_{1,j}x_{1,i}.$$

The strict transform of every 2-minor of $B_3$ that does not contain $x_{1,1}$ is already contained in $I$: $y_{i,j} = x'_{i,j} - x'_{1,i}x'_{1,j}$ is by definition the strict transform of $x_{1,1}x_{1,j} - x_{i,1}x_{1,j}$. Furthermore, $I = \langle y_{2,2}, y_{2,3}, y_{3,3}\rangle$ is the strict transform of $\langle x_{k,\ell}x_{i,j} - x_{i,\ell}x_{k,j} \mid 1 \leq k, \ell, i, j \leq 3\rangle$. For $i \neq 1 \neq j, b \neq 1$ we have

$$x'_{1,\ell}x'_{i,j} - x'_{i,\ell}x'_{1,j} = x'_{1,\ell}(y'_{i,j} - x'_{1,i}x'_{1,j}) - (y_{i,\ell} + x'_{1,i}x'_{1,\ell})x'_{1,j} = x'_{1,\ell}y_{i,j} - y_{i,\ell}x'_{1,j} \in I.$$

In other words, the center of the second blowing up is the strict transform of the determinantal singularity $Y^{\mathrm{sym}}_{3,2}$ defined by the 2-minors of $B_3$. The situation in the $X_{2,2}$- and the $X_{3,3}$-chart is analogous.

Let us study the $X_{2,3}$-chart. (The cases of the $X_{1,2}$- resp. the $X_{1,3}$-chart are analogous.) In order to have a clear distinction to the $X_{1,1}$-chart, we write $\widetilde{x}_{i,j}$ instead of $x'_{i,j}$ for the variables in the present chart. Hence, the transformation of the variables is $x_{2,3} = \widetilde{x}_{2,3}$ and $x_{i,j} = \widetilde{x}_{2,3}\widetilde{x}_{i,j}$ for $(i,j) \neq (2,3)$. The strict transform of $Y^{\mathrm{sym}}_{3,3}$ is given by the determinant

$$\det\begin{pmatrix} \widetilde{x}_{1,1} & \widetilde{x}_{1,2} & \widetilde{x}_{1,3} \\ \widetilde{x}_{1,2} & \widetilde{x}_{2,2} & 1 \\ \widetilde{x}_{1,3} & 1 & \widetilde{x}_{3,3} \end{pmatrix} = (\widetilde{x}_{2,2}\widetilde{x}_{3,3} - 1)\widetilde{x}_{1,1} + h,$$

where $h$ does neither depend on $\widetilde{x}_{1,1}$ nor on the exceptional variable $\widetilde{x}_{2,3}$.

By Remark 2.2.4, the set of points for which $\widetilde{x}_{2,2}\widetilde{x}_{3,3} - 1 = \frac{X_{2,2}}{X_{2,3}}\frac{X_{3,3}}{X_{2,3}} - 1 = 0$ is entirely contained in the intersection of the $X_{2,2}$- and the $X_{3,3}$-chart. Therefore, we may assume without loss of generality that $\widetilde{x}_{2,2}\widetilde{x}_{3,3} - 1$ is invertible in the given chart. This implies that the strict transform of $Y_{3,3}^{\mathrm{sym}}$ is regular and transversal to the exceptional divisor $\mathrm{div}(\widetilde{x}_{2,3})$. In other words, we have resolved the singularities of $Y_{3,3}^{\mathrm{sym}}$.

Notice that $\widetilde{x}_{2,2}\widetilde{x}_{3,3} - 1$ is the strict transform of the 2-minor of $B_3$ containing $x_{2,3}$ twice. Hence, the strict transform of $Y_{3,2}^{\mathrm{sym}}$ is empty in the $X_{2,3}$-chart, if we assume that we neglect the part that is already contained in $D_+(X_{2,2}) \cap D_+(X_{3,3})$.

In conclusion, we have seen that the strict transform of $Y_{3,2}^{\mathrm{sym}}$, which is the next center for blowing up proposed by Theorem 5, is regular and it is contained in the union of three charts given by the diagonal elements, $(Y_{3,2}^{\mathrm{sym}})' \subset D_+(X_{1,1}) \cup D_+(X_{2,2}) \cup D_+(X_{3,3})$. After blowing up $(Y_{3,2}^{\mathrm{sym}})'$, we have resolved $Y_{3,3}^{\mathrm{sym}}$.

Note that we have especially seen that $Y_{3,2}^{\mathrm{sym}}$ is resolved after blowing up $Y_{3,1}^{\mathrm{sym}}$.

The next example illustrates the method for the reduction in the skew-symmetric setting.

**Example 6.2.6** ($m = 4$, skew-symmetric)**.** Assume that $\mathrm{char}(R) \neq 2$. Recall that $Y_{4,4}^{\mathrm{skew}}$, is given by $\det(A_4) = 0 = \mathrm{pf}(A_4)$, where

$$A_4 = \begin{pmatrix} 0 & x_{1,2} & x_{1,3} & x_{1,4} \\ -x_{1,2} & 0 & x_{2,3} & x_{2,4} \\ -x_{1,3} & -x_{2,3} & 0 & x_{3,4} \\ -x_{1,4} & -x_{2,4} & -x_{3,4} & 0 \end{pmatrix}.$$

Theorem 4 suggests $Y_{4,1}^{\mathrm{skew}} = \mathrm{Spec}(S/\langle x_{1,2}, \ldots, x_{3,4}\rangle)$ as the first center to blow-up.

We consider the $X_{1,2}$-chart of this blowing up. (All other charts are analogous.) In there, we have $x_{1,2} = x'_{1,2}$ and $x_{i,j} = x'_{1,2}x'_{i,j}$ for $(i,j) \neq (1,2)$. In particular, the strict transform $(Y_{4,4}^{\mathrm{skew}})'$ is defined by $\det(A'_4) = 0$, where

$$A'_4 = \begin{pmatrix} 0 & 1 & x'_{1,3} & x'_{1,4} \\ -1 & 0 & x'_{2,3} & x'_{2,4} \\ -x'_{1,3} & -x'_{2,3} & 0 & x'_{3,4} \\ -x'_{1,4} & -x'_{2,4} & -x'_{3,4} & 0 \end{pmatrix}.$$

Observe that $(Y_{4,1}^{\mathrm{skew}})' = \varnothing$ (since the strict transform of the center is always empty) and $(Y_{4,2}^{\mathrm{skew}})' = \varnothing$ (since there is a 2-minor that is equal to one after the blowing up [62, Theorem 3.2]).

We perform elementary row operations on $A_4'$ which do neither change the pfaffian nor the determinant. First, we add $x'_{2,3}$-times the first row to the third row and $x'_{2,4}$-times the first row to the fourth row. After that the second column becomes $(1,0,0,0)^T$ and we can eliminate the entries at position $(1,3)$ and $(1,4)$ via column operations. We get

$$\mathrm{pf}(A_4') = \mathrm{pf}\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & x'_{2,3} & x'_{2,4} \\ -x'_{1,3} & 0 & x'_{1,3}x'_{2,3} & x'_{3,4}+x'_{1,4}x'_{2,3} \\ -x'_{1,4} & 0 & -x'_{3,4}+x'_{1,3}x'_{2,4} & x'_{1,4}x'_{2,4} \end{pmatrix}.$$

Next, we add $x'_{2,3}$-times the first column on the third column and $x'_{2,4}$-times the first column on the fourth column to obtain that the second row becomes $(-1,0,0,0)$. Then, we eliminate the entries at position $(3,1)$ and $(4,1)$ via row operations. This provides

$$\mathrm{pf}(A_4') = \mathrm{pf}\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x'_{3,4}+x'_{1,4}x'_{2,3}-x'_{1,3}x'_{2,4} \\ 0 & 0 & -x'_{3,4}+x'_{1,3}x'_{2,4}-x'_{1,4}x'_{2,3} & 0 \end{pmatrix}.$$

This equals

$$\mathrm{pf}(A_4') = \mathrm{pf}\begin{pmatrix} 0 & y_{3,4} \\ -y_{3,4} & 0 \end{pmatrix} = y_{3,4}, \quad \text{for } \; y_{3,4} := x'_{3,4}+x'_{1,4}x'_{2,3}-x'_{1,3}x'_{2,4}.$$

Therefore, $(Y^{\mathrm{skew}}_{4,4})'_{\mathrm{red}}$ coincides with the divisor $\mathrm{div}(y_{3,4})$ in the given chart. The latter is regular and transversal to the exceptional divisor $\mathrm{div}(x_{1,2})$. Therefore, the singularity is resolved. This is not surprising, since we have

$$\mathrm{pf}(A_4) = (x_{1,2}x_{3,4}+x_{1,4}x_{2,3}-x_{1,3}x_{2,4}).$$

We point out that $y_{3,4}$ is the reduced strict transform of the 3-minor of $A_4$ containing $x_{1,2}$ twice and $x_{3,4}$, i.e., the minor of $(A_4)_{\{1,2,3\},\{1,2,4\}}$:

$$x_{1,2}\big(x_{1,2}x_{3,4}+x_{1,4}x_{2,3}-x_{1,3}x_{2,4}\big) = \det\begin{pmatrix} 0 & x_{1,2} & x_{1,4} \\ -x_{1,2} & 0 & x_{2,4} \\ -x_{1,3} & -x_{2,3} & x_{3,4}. \end{pmatrix}.$$

Furthermore, note that for every non-zero 3-minor the corresponding matrix contains

exactly one element $x_{i,j}$ twice and it is up to sign of the form

$$x_{i,j}\big(x_{1,2}x_{3,4} + x_{1,4}x_{2,3} - x_{1,3}x_{2,4}\big).$$

The other 3-minors are determinants of a generic skewsymmetric square matrix of size 3, so they equals 0. Hence, by Example 6.2.4(1) and Remark 6.2.2(2), the ideal of $(Y^{\text{skew}}_{4,3})'$ is equal to $\langle y_{3,4}\rangle$ – recall that we consider the $X_{1,2}$-chart. Thus, we have $(Y^{\text{skew}}_{4,3})' = (Y^{\text{skew}}_{4,4})'_{\text{red}}$.

For completeness, let us also have a look at the skew-symmetric case for $m = 3$.

**Example 6.2.7** ($m = 3$, skew-symmetric)**.** As explained in the introduction of this chapter, $Y^{\text{skew}}_{3,1}$ and $Y^{\text{skew}}_{3,3}$ are regular. Thus, let us have a look at the 2-minors of

$$A_3 = \begin{pmatrix} 0 & x_{1,2} & x_{1,3} \\ -x_{1,2} & 0 & x_{2,3} \\ -x_{1,3} & -x_{2,3} & 0 \end{pmatrix}.$$

The principal 2-minors provide that $x^2_{1,2}, x^2_{1,3}, x^2_{2,3} \in \mathcal{I}_2(A_3)$ (using the notation of Definition 6.2.3). This implies that

$$(Y^{\text{skew}}_{3,1})_{\text{red}} \cong \text{Spec}(R_0[x_{1,2}, x_{1,3}, x_{2,3}]/\langle x_{1,2}, x_{1,3}, x_{2,3}\rangle) \cong \text{Spec}(R_0)$$

is regular and no blowing ups are required.

## 6.2.2. Proof of Main Theorem 4

*Proof of Theorem 4.* Recall that $R_0$ is a regular ring of characteristic different from 2. In order to lighten the notation, we abuse notation and write

$$S := S_{\text{skew}} \cong R_0[x_{i,j} \mid 1 \leq i < j \leq m], \qquad Z := Z_{\text{skew}} = \text{Spec}(S).$$

Recall that

$$A_m = \begin{pmatrix} 0 & x_{1,2} & \cdots & x_{1,m} \\ -x_{1,2} & 0 & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ -x_{1,m} & -x_{2,m} & \cdots & 0 \end{pmatrix}.$$

Using Definition 6.2.3 and identifying $S$ with the ring on the right hand side of the isomorphism above, the determinantal singularity which we aim to resolve is given as

$$Y_{m,r} := Y_{m,r}^{\text{skew}} = \text{Spec}(S/\mathcal{I}_r(A_m)).$$

First, observe that $Y_{m,1}$ is always regular, as it is isomorphic to $\text{Spec}(R_0)$ and $R_0$ is a regular ring by assumption. Moreover, by considering the principal 2-minors, we see that $(Y_{m,2})_{\text{red}} = (Y_{m,1})_{\text{red}}$.

Let $\ell \geq 2$ and assume that $Z_0 \xleftarrow{\pi_1} Z_1 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_{\ell-2}} Z_{\ell-2}$ is an embedded resolution of $(Y_{m,2(\ell-1)})_{\text{red}}$. Hence, all centers $(Y_{m,2j})_{\text{red}}$ are regular and global defined, for all $j = 0, \ldots, \ell-2$.

It remains to show that the blowing up $\pi_{\ell-1}$ resolves $(Y_{m,2\ell})_{\text{red}}$.

We consider $A_m$ with $m = 2\ell$.

We consider the case of $Y_{m,m}$. As the examples of the previous section indicate, we will prove the statement for $Y_{m,r}$ with $r < m$ along the way.

We only have to blow-up in the $(2k-1)$-minors since the $2k$-minors are zero after blow-up in the $(2k-1)$-minors, since $\sqrt{\langle (2k-1)\text{-minors}\rangle} = \sqrt{\langle (2k-1)\text{-minors}\rangle}$. We have

$$(Y_{m,m})_{\text{red}} = \text{Spec}(S/\operatorname{pf}(A_m)).$$

We perform an induction on the size $m \in \mathbb{Z}_+$ of the matrix $A_m$. For $m = 1$, there is nothing to prove. Furthermore, we treated the cases $m \in \{2, 3, 4\}$ in Examples 6.2.1, 6.2.7, 6.2.6, respectively. Therefore, we go on to the induction step and assume $m \geq 5$ in the following.

The first center proposed by Theorem 4 is $D_1 := Y_{m,1} = (Y_{m,2})_{\text{red}}$, which is regular as we discussed explained before. Let us look at a chart of the blowing up with center $D_1$. Without loss of generality, we consider the $X_{1,2}$-chart. (Note that for the $X_{k,\ell}$-chart, we may perform a suitable interchange of columns and rows followed by a renaming of the variables in order to attain in the same setting as the $X_{1,2}$-chart.)

In the given chart, the strict transform $Y'_{m,m}$ of $Y_{m,m}$ is determined by the strict transform $f'$ of $f := \operatorname{pf}(A_m)$. We have that $f'$ is equal to the following pfaffian

$$f' = \operatorname{pf}\begin{pmatrix} 0 & 1 & x'_{1,3} & \cdots & x'_{1,m} \\ -1 & 0 & x'_{2,3} & \cdots & x'_{2,m} \\ -x'_{1,3} & -x'_{2,3} & 0 & \cdots & x'_{3,m} \\ \vdots & \vdots & & \ddots & \vdots \\ -x'_{1,m} & -x'_{2,m} & -x'_{3,m} & \cdots & 0 \end{pmatrix}.$$

We eliminate all entries in the second column of the matrix except for the 1 at position $(1,2)$ by adding the first row $x'_{2,i}$-times to the $i$-th row, for $i \geq 3$. After that we perform column operations to eliminate all entries in the first row except for the 1 at position $(1,2)$. The latter step has no effect on the other entries of the matrix since we cleaned the second column beforehand. We can obtain that $f'$ is equal to

$$\operatorname{pf}\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & x'_{2,3} & \cdots & x'_{2,m} \\ -x'_{1,3} & 0 & x'_{1,3}x'_{2,3} & \cdots & x'_{3,m} + x'_{1,m}x'_{2,3} \\ \vdots & \vdots & & \ddots & \vdots \\ -x'_{1,m} & 0 & -x'_{3,m} + x'_{1,3}x'_{2,m} & \cdots & x'_{1,m}x'_{2,m}. \end{pmatrix}$$

In order to regain the skew-symmetry, we first add $(-x_{1,i})$-times the second row to the $i$-th row, for $i \geq 3$, which eliminates all entries in the first column except for the $-1$ at position $(2,1)$. Using the latter entry, we then perform column operations to clean up the second row. This leads to

$$f' = \operatorname{pf}\begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & y_{3,4} & \cdots & y_{3,m} \\ 0 & 0 & -y_{3,4} & 0 & \cdots & y_{4,m} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -y_{3,m} & -y_{4,m} & \cdots & 0 \end{pmatrix},$$

where we introduce

$$y_{i,j} := x'_{i,j} - x'_{2,j}x'_{1,i} + x'_{1,j}x'_{2,i}, \quad \text{for } 3 \leq i < j \leq m.$$

Observe that $(y_{i,j})$ are independent variables and they are transversal to the variable $x'_{2,1}$, which describes the exceptional divisor of the blowing up in the present chart.

Finally, we expand the pfaffian with respect to the first two columns and get

$$f' = \operatorname{pf}\begin{pmatrix} 0 & y_{3,4} & \cdots & y_{3,m} \\ -y_{3,4} & 0 & \cdots & y_{4,m} \\ \vdots & \vdots & \ddots & \vdots \\ -y_{3,m} & -y_{4,m} & \cdots & 0 \end{pmatrix}.$$

Note that the latter is the pfaffian of the generic skew-symmetric matrix of size

$(m-2)\times(m-2)$. Therefore, by induction on the size of the matrix, Theorem 4 provides an embedded resolution for the subscheme determined by the vanishing of $f'$.

It remains to show that the desingularization obtained by induction coincides with the one proposed in the statement of Theorem 4. In other words, we have to prove that the strict transform $Y'_{m,r}$ of $Y_{m,r}$ is equal to $Y_{m-2,r-2}$ (up to renaming the generic variables) in the present chart, i.e., if we set

$$A'_m := \begin{pmatrix} 0 & 1 & x'_{1,3} & \cdots & x'_{1,m} \\ -1 & 0 & x'_{2,3} & \cdots & x'_{2,m} \\ -x'_{1,3} & -x'_{2,3} & 0 & \cdots & x'_{3,m} \\ \vdots & \vdots & & \ddots & \vdots \\ -x'_{1,m} & -x'_{2,m} & -x'_{3,m} & \cdots & 0 \end{pmatrix}, \quad \widetilde{A}_{m-2} := \begin{pmatrix} 0 & y_{3,4} & \cdots & y_{3,m} \\ -y_{3,4} & 0 & \cdots & y_{4,m} \\ \vdots & \vdots & \ddots & \vdots \\ -y_{3,m} & -y_{4,m} & \cdots & 0 \end{pmatrix}$$

and use the notation introduced in Definition 6.2.3, then we have to prove:

$$\mathcal{I}_{r-2}(\widetilde{A}_{m-2}) = \mathcal{I}_r(A'_m). \tag{6.2.3}$$

Notice that this also provides that the desingularizations obtained by induction in every chart of the first blowing up glue to a global resolution of singularities as desired.

Moreover, we obtain from this that $Y'_{m,3} = (Y'_{m,4})_{\mathrm{red}}$ is regular and transversal to the exceptional divisor $\mathrm{div}(x'_{1,2})$ since the last two properties are true for $Y_{m-2,1}$.

We already explained that $Y'_{m,m}$ identifies with $Y_{m-2,m-2}$. Moreover, we have $Y'_{m,1} = \varnothing$ (since the strict transform of the center is always empty) and $Y'_{m,2} = \varnothing$ (since we have $(Y_{m,2})_{\mathrm{red}} = Y_{m,1}$).

Let $r \geq 3$. By Example 6.2.4(1) and Remark 6.2.2(2), $Y'_{m,r}$ is generated by the radical of the $r$-minors of the matrix $A'_m$. As we have seen before, after some elementary row and column operation the matrix $A'_m$ becomes

$$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & \widetilde{A}_{m-2} \end{pmatrix}.$$

For every $r$-minor $g_{I,J}$ of $A'_m$ corresponding to $I, J \subset \{1,\ldots,m\}$ with $J \supset \{1,2\} \subset I$, we may perform the same row and column operations and obtain that it is equal to the $(r-2)$-minor of the matrix $\widetilde{A}_{m-2}$, whose index sets correspond to $I \setminus \{1,2\}$

and $J \setminus \{1,2\}$ (up to a shift in the index set). This shows

$$\mathcal{I}_{r-2}(\widetilde{A}_{m-2}) \subseteq \mathcal{I}_r(A'_m).$$

We are left with the task to show that the radical of any $r$-minor $g_{I,J}$ of $A'_m$ with $\{1,2\} \not\subset I$ or $\{1,2\} \not\subset J$ is contained in the ideal $\mathcal{I}_{r-2}(\widetilde{A}_{m-2})$.

This belongs to the basic calculation of minors and determinants by gaussian elimination.

Let $I = \{i_1, \ldots, i_r\}$ and $J = \{j_1, \ldots, j_r\}$, where we can assume that $j_1$ is the first column of $A_{I,J}$. We have

$$f'_{I,J} = \det \begin{pmatrix} x'_{i_1,j_1} & y_{i_1,j_2} & \cdots & y_{i_1,j_r} \\ x'_{i_1,j_2} & y_{i_2,j_2} & \cdots & y_{i_2,j_r} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{i_1,j_r} & y_{i_2,j_r} & \cdots & y_{i_r,j_r} \end{pmatrix}$$

$$= \det \begin{pmatrix} x'_{i_1,j_1} & y_{i_1,j_2} + x'_{i_2,j_2}x'_{i_1,j_1} - x'_{i_1,j_2}x'_{i_1,j_2} & \cdots & y_{i_1,j_r} + x'_{i_2,j_r}x'_{i_1,j_1} - x'_{i_1,j_r}x'_{i_1,j_1} \\ x'_{i_1,j_2} & y_{i_2,j_2} + x'_{i_2,j_2}x'_{i_1,j_2} - x'_{i_1,j_2}x'_{i_2,j_1} & \cdots & y_{i_2,j_r} + x'_{i_2,j_r}x'_{i_1,j_2} - x'_{i_r,j_r}x'_{i_1,j_2} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{i_1,j_r} & y_{i_2,j_r} + x'_{i_2,j_r}x'_{i_1,j_2} - x'_{i_1,j_r}x'_{i_2,j_2} & \cdots & y_{i_r,j_r} + x'_{i_2,j_r}x'_{i_1,j_r} - x'_{i_1,j_r}x'_{i_1,j_r} \end{pmatrix}$$

$$= \det \begin{pmatrix} x'_{i_1,j_1} & x'_{i_1,j_2} & \cdots & x'_{i_1,j_r} \\ x'_{i_1,j_2} & x'_{i_2,j_2} & \cdots & x'_{i_2,j_r} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{i_1,j_r} & x'_{i_2,j_r} & \cdots & x'_{i_r,j_r} \end{pmatrix} \in \mathcal{I}_r(A'_m).$$

□

## 6.2.3. Proof of Main Theorem 5

*Proof of Theorem 5.* Analogous to the previous section, we abuse notation and abbreviate

$$S := S_{\mathrm{sym}}, Z := Z_{\mathrm{sym}}, \mathcal{I}_{m,r} := \mathcal{I}^{\mathrm{sym}}_{m,r}, I := I_{\mathrm{sym}}.$$

We come to the proof of Theorem 5. Recall that

$$B_m := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{1,2} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,m} & x_{2,m} & \cdots & x_{m,m} \end{pmatrix}.$$

Our goal is to resolve the singularity which is given as

$$Y_{m,r} := Y_{m,r}^{\mathrm{sym}} = \mathrm{Spec}(S/\mathcal{I}_r(B_m)).$$

As in the proof of Theorem 4, we perfom an induction on the size $m \in \mathbb{Z}_+$ of the matrix $B_m$. The ideal generated by the 1-minors of $B_m$, $Y_{m,1}$ is always regular, since $Y_{m,1} \cong \mathrm{Spec}(R_0)$ and $R_0$ is assumed to be regular. So we have to consider the case $Y_{m,r}$, for $1 < r \leq m$.

Theorem 5 proposed that the first center is $D_1 := Y_{m,1}$ which is regular.

We first have a look at the $X_{1,2}$-chart, which is a representative of the $X_{i,j}$-charts for $1 \leq i < j \leq m$. For the $X_{i,j}$-charts, we may perform a suitable interchange of columns and rows followed by renaming of the charts and we get the same setting as in the $X_{1,2}$-chart.

In the given chart, the strict transform $Y'_{m,m}$ of $Y_{m,m}$ is determined by the strict transform $f'$ of $f := \det(B_m)$. We know that $f'$ is equal to

$$f' = \det \begin{pmatrix} x'_{1,1} & 1 & \cdots & x'_{1,m} \\ 1 & x'_{2,2} & \cdots & x'_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{1,m} & x'_{2,m} & \cdots & x'_{m,m} \end{pmatrix}.$$

We can eliminate all entries in the first row and second column by adding $(-1) \cdot x'_{2,j}$ times the first row on the $j$-th row, for $j = 2, \ldots, m$, we get

$$f' = \det \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 - x'_{1,1}x'_{2,2} & 0 & x'_{2,3} - x'_{2,2}x'_{1,3} & \cdots & x'_{2,m} - x'_{2,2}x'_{1,m} \\ x'_{1,3} - x'_{1,1}x'_{2,3} & 0 & x'_{3,3} - x'_{2,3}x'_{1,3} & \cdots & x'_{3,m} - x'_{2,3}x'_{1,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x'_{1,m} - x'_{1,1}x'_{2,m} & 0 & x'_{3,m} - x'_{2,m}x'_{1,3} & \cdots & x'_{m,m} - x'_{2,m}x'_{1,m} \end{pmatrix}.$$

In order to regain the symmetry, we multiply the first column with $\varepsilon^{-1}$, where $\varepsilon := 1 - x'_{1,1}x'_{2,2}$, which is invertible (see Example 6.2.5) and multiply the $j$-th row $(3 \leq j \leq m)$ with $\varepsilon$, we get

$$f' = \det \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & x'_{2,3} - x'_{2,2}x'_{1,3} & \cdots & x'_{2,m} - x'_{2,2}x'_{1,m} \\ x'_{1,3} - x'_{1,1}x'_{2,3} & 0 & \varepsilon \cdot (x'_{3,3} - x'_{2,3}x'_{1,3}) & \cdots & \varepsilon(x'_{3,m} - x'_{2,3}x'_{1,m}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x'_{1,m} - x'_{1,1}x'_{2,m} & 0 & \varepsilon(x'_{3,m} - x'_{2,m}x'_{1,3}) & \cdots & \varepsilon(x'_{m,m} - x'_{2,m}x'_{1,m}) \end{pmatrix}.$$

Now we apply $\text{row}_j \mapsto \text{row}_j - (x'_{1,j} - x'_{1,1}x'_{2,j})\text{row}_2$ for $3 \leq j \leq m$ and afterwards we apply $\text{column}_j \mapsto \text{column}_j - (x'_{2,j} - x'_{2,2}x'_{1,j})\text{column}_1$. Altogether we get

$$f' = \det \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & y_{3,3} & \cdots & y_{3,m} \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & y_{3,m} & \cdots & y_{m,m} \end{pmatrix},$$

where

$$y_{i,i} = \varepsilon(x'_{i,i} - x'_{2,i}x'_{1,i}) - (x'_{1,i} - x'_{1,1}x'_{2,i})(x'_{2,i} - x'_{2,2}x'_{1,i}),$$

for $3 \leq i \leq m$. Note that this step is well-defined since $\varepsilon$ is a unit. Moreover, note that $y_{i,j}$ are independent variables and they are transversal to the exceptional variable $x'_{1,2}$. Furthermore, we observe that

$$\begin{aligned} y_{i,j} = &\ \varepsilon(x'_{i,j} - x'_{2,j}x'_{1,i}) - (x'_{i,i} - x'_{1,1}x'_{2,i})(x'_{2,j} - x'_{2,2}x'_{1,j}) \\ = &\ \varepsilon x'_{i,j} - x'_{1,j}x'_{2,i} - x'_{1,i}x'_{2,j} + x'_{1,i}x_{1,j}x'_{2,2} + x'_{1,1}x'_{2,i}x'_{2,4} \\ = &\ \varepsilon(x'_{i,j} - x'_{2,i}x'_{1,j}) - (x'_{j,j} - x'_{1,1}x'_{2,j})(x'_{2,i} - x'_{2,2}x'_{1,i}) \\ = &\ y_{j,i}, \end{aligned}$$

for $3 \leq i, j \leq m$. So the resulting matrix is symmetric.

Like in the previous approaches, we can expand the determinant by the first two rows and we get

$$f' = \det \begin{pmatrix} y_{3,3} & \cdots & y_{3,m} \\ \vdots & \ddots & \vdots \\ y_{3,m} & \cdots & y_{m,m} \end{pmatrix}.$$

Again, we reduced the problem to a determinant of a matrix with size $(m-2) \times (m-2)$.

In this $X_{1,2}$-chart, the vanishing locus of the 2-minors is empty, since $\det \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \in \langle 2\text{-minors} \rangle$ using the assumption that $\varepsilon$ is invertible. In this chart $\langle 2\text{-minors} \rangle' = \varnothing$. Therefore, we do not see the second center of the resolution procedure here in this chart. Hence, the chart remains unchanged under the next blowing up. Since $\langle 3\text{-minors} \rangle' \neq \varnothing$ the third center of the blowing up procedure changes the chart again.

In contrast to the previous chapter we have to look at the $X_{1,1}$-chart as representative of the $X_{i,i}$-charts for $1 \leq i \leq m$, too. Again, by interchanging columns and rows and by renaming variables, we have in the other $X_{i,i}$-charts the same situation.

In considered $X_{1,1}$-chart, the strict transform $f'$ of $f$ is given by the following determinant

$$f' = \det \begin{pmatrix} 1 & x'_{1,2} & \cdots & x'_{1,m} \\ x'_{1,2} & x'_{2,2} & \cdots & x'_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{1,m} & x'_{2,m} & \cdots & x'_{m,m} \end{pmatrix}.$$

We will eliminate the entries in the first column by adding $(-1) \cdot x_{1,j}$-times the first row to the $j$-th row, for $j \geq 2$. We get

$$f' = \det \begin{pmatrix} 1 & x'_{1,2} & \cdots & x'_{1,m} \\ 0 & x'_{2,2} - x'^2_{1,2} & \cdots & x'_{2,m} - x'_{1,m}x'_{1,2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x'_{2,m} - x'_{1,2}x'_{1,m} & \cdots & x'_{m,m} - x'^2_{1,m} \end{pmatrix}.$$

For eliminating the entries of the first row, we do the same column operation as for the row operation above, i.e., $(-1) \cdot x_{1,j}$-times the first column to the $j$-th column, for $j \geq 2$. This yields

$$f' = \det \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & x'_{2,2} - x'^2_{1,2} & \cdots & x'_{2,m} - x'_{1,m}x'_{1,2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x'_{2,m} - x'_{1,2}x'_{1,m} & \cdots & x'_{m,m} - x'^2_{1,m} \end{pmatrix}.$$

Like we have seen in Observation 6.2.2, we now can expand the determinant with respect to the first column or first row, we get

$$f' = \det \begin{pmatrix} y_{2,2} & \cdots & y_{2,m} \\ \vdots & \ddots & \vdots \\ y_{2,m} & \cdots & y_{m,m} \end{pmatrix}, \text{ where } y_{i,j} := x'_{i,j} - x'_{1,i}x_{1,j}, \text{ for } 2 \leq i, j, \leq m.$$

These $y_{i,j}$ are independent variables which are transversal to the exceptional variable $x'_{1,1}$. This shows, that $f'$ is now defined by the generic $(m-1) \times (m-1)$ matrix.

Therefore, by induction on the size of the matrix, Theorem 5 provides an embedded resolution for the subscheme determined by the vanishing locus of $f'$.

It remains to show, that this desingularization coincides with the one proposed in the statement of Theorem 5, i.e., we have to prove that the strict transform $Y'_{m,r}$ of $Y_{m,r}$ resp. $Y''_{m,r}$ of $Y_{m,r}$ (after two blow-ups) coincide with $Y_{m-1,r-1}$ resp. $Y_{m-2,r-2}$.

This resolution is a global one with the same argument as before.

□

An implementation of the author is described in Section A.5. The complexity analysis of the number of charts can be found in the appendix in Section B.2. Experiments on low dimensional examples show that our implementation is faster than the general implementation of Villamayor's algorithm [16] of resolution of singularities in the singular library resolve.lib [37]. This belongs to the fact that our specific implementation needs less time per chart and in total we have to consider less charts (and less final charts). For more information see Section B.4.

So the set of resolution of singularities which are computable in practice is extended by this implementation.

# 7. Algorithmic local monomialization of a single binomial: efficiency considerations

This section presents the article [42] in our context. This article is a joint work with Bernd Schober published in International Journal of Algebra and Computation. Bernd Schober was a part of the DFG-project "Order zeta functions and resolutions of singularities" (principal investigators: Christopher Voll and Anne Frühbis-Krüger), he could motivate the article in the context of explicit computation of specific $p$-adic integrals. In particular, the structure of the integrals considered there allow a reduction to the case of finitely many binomials. An increasing complexity in the $p$-adic integrals (which is reflected in a rapidly increasing number of variables and binomials) requires to find monomialization algorithms which keep the numbers of blowups and final charts that have to be considered small. In contrast to the focus on application the author of this thesis concentrated on the heart of the article, namely the algorithmical and implementational parts of the article resp. chapter.

The goal of this section is to investigate and to compare different methods of transforming a single binomial into a monomial via blowing up appropriate centers. Within this, we develop explicit implementations so that the different approaches can be compared on the basis of numerous examples.

Since the problem is of combinatorial nature, we focus on the situation over a field, while we briefly discuss the case over $\mathbb{Z}_p$ in Remark 7.6.4.

**Definition 7.0.1.** Let $K$ be a field and let $f = \underline{x}^A - \rho\underline{x}^B \in K[\underline{x}] := K[x_1, \ldots, x_n]$ be a binomial, where $\rho \in K^\times$ and $\underline{x}^A = x_1^{A_1} \cdots x_n^{A_n}$ for $A = (A_1, \ldots, A_n) \in \mathbb{Z}_{\geq 0}^n$. We say that $f$ is *locally a monomial* if for every point $q \in \mathbb{A}_K^n = \mathrm{Spec}(K[\underline{x}])$ there exists a regular system of parameters for the local ring $\mathcal{O}_{\mathbb{A}_K^n, q}$ such that $f$ is a monomial

times a unit with respect to these parameters.

For example, $x_1^3x_2^2(1-x_1)^4$ is locally monomial since for every $q \in \mathbb{A}_K^2$ at least $x_1$ or $1-x_1$ is a unit. On the other hand, $x_1x_2(x_1+x_2)$ is not locally a monomial as there is no regular system of parameter for the local ring at the origin such that $x_1x_2(x_1+x_2)$ becomes a monomial times a unit. The tool that we want to apply to make a binomial locally monomial are blowups, e.g., $x_1x_2(x_1+x_2)$ becomes locally monomial after blowing up with center $V(x_1,x_2)$.

The tool that we want to apply to make a binomial locally monomial are blowups as considered in Remark 2.2.6.

**Definition 7.0.2.** We say that a finite sequence of local blow-ups obtained by iterating the previous procedure is a *local monomialization of* $f = \underline{x}^A - \rho\underline{x}^B$ if the total transform of $f$ is locally a monomial in every final chart of the blowup tree.

For example, the latter is the case if the total transform is of the form (up to multiplication by a non-zero constant)

**Example 7.0.3.** Let $g = x_1x_2(x_1+x_2) \in K[x_1,x_2]$. Then $g$ is not locally a monomial as there is no regular system of parameter for the local ring at the origin such that $g$ becomes a monomial times a unit. $g$ becomes locally monomial after blowing up with center $V(x_1,x_2)$.

A finite sequence of local blowups obtained by iterating the previous procedure is a local monomialization if the total transform is of the form (up to multiplication by a non-zero constant)

$$\underline{x}^C(1-\mu\underline{x}^{B'}) \quad \text{or} \quad \underline{x}^C(x_i-\mu\underline{x}^{B'}), \tag{7.0.1}$$

for $C, B' \in \mathbb{Z}^n_{\geq 0}$ and $\mu \in K^\times$, where we require in the second case that $C_i = B'_i = 0$ for the special $i \in \{1,\ldots,n\}$ given. The last hypothesis implies that we may introduce the coordinate change $y_i := x_i - \mu\underline{x}^{B'}$ such that $\underline{x}^C(x_i-\mu\underline{x}^{B'}) = \underline{x}^C y_i$ becomes a monomial. It is clear that $\underline{x}^C(1-\mu\underline{x}^{B'})$ is a monomial times a unit if $1-\mu\underline{x}^{B'}$ is invertible in $\mathcal{O}_{\mathbb{A}^n_K,q}$. On the other hand, if $1-\mu\underline{x}^{B'}$ is not invertible, we have to distinguish two cases. Let $d$ be the greatest common divisor of the entries of $B'$. First, if $d=1$ or if $d>1$ and $\mu$ has no $d'$-th root in $K$ with $d'|d$, then $1-\mu\underline{x}^{B'}$ is irreducible and regular. Hence, it can be extended to a regular system of parameters at $q$. Otherwise, $1-\mu\underline{x}^{B'}$ is not irreducible, but then all but one of the distinct irreducible factors are invertible since they arise from the factorization of $T^{d'}-1 \in K[T]$ (with maximal $d'$ as in the first case and $T$ a substitute for $(\mu\underline{x}^{B'})^{1/d'} \in K[\underline{x}]$). Note that in both cases all $x_i$ with $B'_i \neq 0$ are units in $\mathcal{O}_{\mathbb{A}^n_K,q}$

Observe that (7.0.1) is not equivalent to being locally monomial. For example, $x^2 - y^2 \in \mathbb{C}[x, y]$ and $x^p + y^p \in \mathbb{F}_p[x, y]$ ($p \in \mathbb{Z}$ prime) do not fulfill (7.0.1), but they are monomial after a suitable change of variables $\widetilde{x} := x + y, \widetilde{y} := x - y$ in the first example and $\bar{x} := x + y$ for the second.

We choose the centers in each chart independent of the other charts, i.e., we do not necessarily obtain a sequence of global blowups. This provides more freedom in the choice of the center and is still sufficient for the explicit computations, where the local charts are interpreted as case distinctions. In [72, Question 5.6], this local variant of monomialization via blowups is discussed in the context of resolution of singularities.

The differences in the methods appear in choice of the center for the next blowup. Let us briefly explain the variants. Consider a binomial

$$f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B),$$

where $A, B, C \in \mathbb{Z}^n_{\geq 0}$ are such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. Set

$$g := \underline{x}^A - \rho\underline{x}^B.$$

Suppose that $f$ is not locally monomial. The basic idea for the four variants that we consider are:

**(1)** *Centers contained in the locus of maximal order* (section 7.2 / Construction 7.2.3). Choose $D = V(x_i \mid i \in I)$ such that $D$ is contained in the locus of maximal order of $V(g)$. This is equivalent to imposing

$$\min\big\{\sum_{i\in I} A_i, \sum_{i\in I} B_i\big\} = \min\big\{|A|, |B|\big\}.$$

**(2)** *Centers of codimension two* (section 7.3 / Construction 7.3.4). Choose $i, j \in \{1, \ldots, n\}$ such that $A_i \neq 0$, $B_j \neq 0$ and both are maximal. Then, the center for the blowup is $D = V(x_i, x_j)$.

**(3)** *Centers of minimal codimension contained in the singular locus* (section 7.4 / Construction 7.4.1). If $\min\big\{|A|, |B|\big\} \geq 2$, choose $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i\in I} A_i \geq 2$, $\sum_{i\in I} B_i \geq 2$ and such that $\#I$ is minimal with this property. Then $D := V(x_i \mid i \in I)$. Else, choose $D$ as in (2).

**(4)** *Centers of minimal codimension contained in an exceptional divisor or contained in the singular locus* (section 7.5 / Construction 7.5.1). If there

is a center $D$ of type (2) contained in an exceptional divisor, we choose this. Otherwise, we follow (3).

Note that the centers are not necessarily uniquely determined and one might have to make a choice. In the respective sections, we provide examples for this phenomenon.

While (1) follows the usual approach to resolution of singularities, method (2) solely has the motivation to minimize the numbers of charts after a single blowup in order to make it easier to control the transform of the binomial. In particular, the resulting morphism is not necessarily an isomorphism outside of the singular locus of $g$. In (3), we consider a mixture of (1) and (2); we try to choose the centers as large as possible, but moreover, we require that the centers are contained in the singular locus of $g$ (resp. its variant after a blowup) and if the latter is empty, we follow (2). Finally, in (4), we relax the last condition (3) and allow centers of codimension two, which are not necessarily contained in the singular locus of $g$, if they are contained in an exceptional divisor.

In the respective sections, we discuss the benefits of each approach and show the termination of the local monomialization algorithm resulting from the different choices, see Propositions 7.2.7, 7.3.5, 7.4.2, and Corollary 7.5.2, respectively. Along this, we discuss algorithms for explicit implementations of each variant to monomialize a binomial, which have been realized in the open source computer algebra system Singular [30].

We study the binomial through the appearing exponents $A, B, C$ and their behavior along the combinatorial interpretation of blowups using $\phi_{\pi,i}$ of Remark 2.2.6. More precisely, we deduce from the exponents numerical measures that detect how far the given binomial is from being monomial. Then we show that the respective measure decreases strictly after a single blowup following the corresponding procedure and that a strict decrease may only appear finitely many times.

In the Appendix in Section B.3, we analyze the different variants for the choice of the centers by comparing the numbers of charts for a worst case scenario and for numerous explicit examples. The latter is based on an implementation of the discussed algorithms in Singular. As a measure for the complexity, we consider the number of charts along the monomialization process as well as at the end. A brief summary is that variant (1) has a significant larger number than the other three, while (2) is often the most efficient algorithm. But there exist cases, where (3) and (4) are slightly better than (2). As mentioned above, we have to make a choice among the possible centers. For some of the example, we study the different results if we vary the choices.

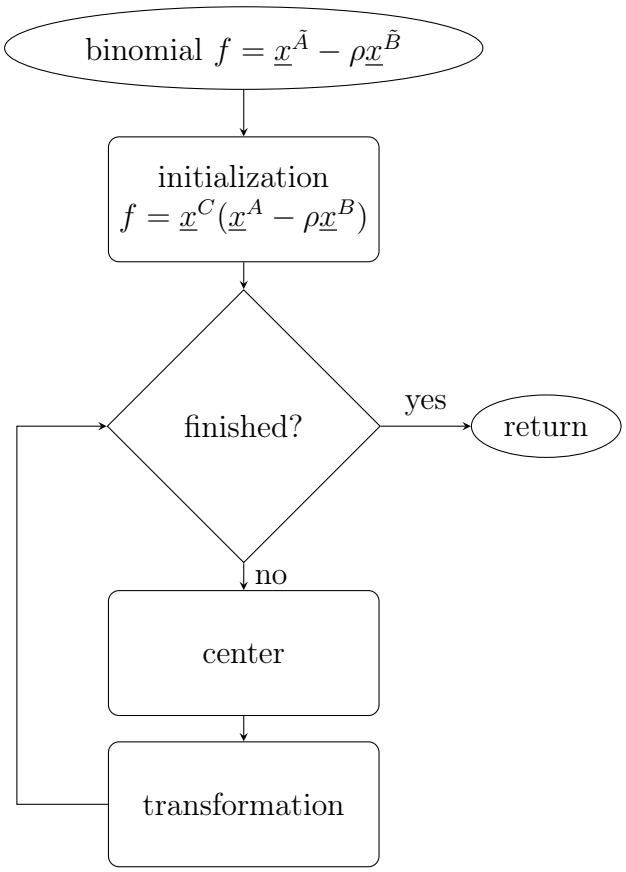



Figure 7.1.: Flow chart of the local monomialization of a single binomial

## 7.1. The basic algorithmic framework

We begin by discussing the basic structure for the implementation of a monomialization procedure. Within this, we also introduce numerical invariants, which we later use to prove the termination of the different monomialization methods. Furthermore, we provide an algorithm testing whether a given binomial fulfills condition (7.0.1), which implies that the binomial is locally monomial. Finally, we give an implementation of the transformation of the exponents along a blowup.

The main method is the same for all of the four strategies. The difference of the monomialization methods appears only in the choice of the center. In Figure 7.1, we provide the flow chart of the main method and in Algorithm 7 the precise implementation.

The implementations are of combinatorial nature. Instead of working with the binomial $\underline{x}^{C}(\underline{x}^{A} - \rho\underline{x}^{B})$, we consider the exponents $A = (A_1, \ldots, A_n)$, $B = (B_1, \ldots, B_n)$ and $C = (C_1, \ldots, C_n)$. Additionally, we introduce a vector of ones and zeros $E = (E_1, \ldots, E_n) \in \{0, 1\}^n$, where we encode, which variables correspond

to exceptional divisors, i.e., $E_i = 1$ if and only if $\mathrm{div}(x_i)$ is an exceptional divisor. This will be necessary for the variant for choosing the center of section 7.5, see Construction 7.5.1.

Before stating and explaining Algorithm 7, let us introduce the following numbers, which play an important role in parts of the monomialization procedures discussed in the present work.

**Definition 7.1.1.** Let $K$ be any field. Let $g = \underline{x}^A - \rho\underline{x}^B \in K[\underline{x}]$, for $\rho \in K^\times$ and $A, B \in \mathbb{Z}^n_{\geq 0}$ such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. We define

$$\begin{aligned}
\alpha(g) &:= \max\{A_i \mid i \in \{1,\ldots,n\}\} \\
\mathfrak{a}(g) &:= \#\{i \in \{1,\ldots,n\} \mid A_i = \alpha(g)\} \\
\beta(g) &:= \max\{B_i \mid i \in \{1,\ldots,n\}\} \\
\mathfrak{b}(g) &:= \#\{i \in \{1,\ldots,n\} \mid B_i = \beta(g)\} \\
\iota(g) &:= (\alpha(g), \mathfrak{a}(g), \beta(g), \mathfrak{b}(g)) \in \mathbb{Z}^4_{\geq 0}.
\end{aligned}$$

Here, we equip $\mathbb{Z}^4_{\geq 0}$ with the lexicographical ordering $\leq_{\ell p}$. Given $f = \underline{x}^C g$ with $C \in \mathbb{Z}^n_{\geq 0}$, we also write $\alpha(f) := \alpha(g), \ldots, \iota(f) := \iota(g)$, if no confusion is possible.

Clearly, $\iota(g)$ depends on the order of the monomials in $g$. In general, we have $\iota(\underline{x}^A - \underline{x}^B) \neq \iota(-\underline{x}^B + \underline{x}^A)$. Since we fix an order of the monomials in an implementation anyways, we will work later with the string $(A, B)$ instead of $g$ and we neglect the matter of making $\iota(g)$ independent of the order of the monomials.

**Remark 7.1.2** (Algorithm 7)**.** The *input* is:

- a binomial $f = \underline{x}^{\tilde{A}} - \rho\underline{x}^{\tilde{B}} \in K[\underline{x}] = K[x_1, \ldots, x_n]$;
- an integer mode $\in \{1, 2, 3, 4\}$, which determines the method for choosing the centers;

The *output* of Algorithm 7 is a list $L$ which consists of all charts of the monomialization process. From this list, one can determine a list of all leaves of the monomialization tree, i.e., of all final charts. The *data in a chart* $L[i]$ is of the following form:

[1] $(A, B, C, E) \in (\mathbb{Z}^n_{\geq 0})^4$ such that the total transform of $f$ in the chart is $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B)$ and $A_iB_i = 0$, for all $i \in \{1, \ldots, n\}$. Furthermore, $E \in \{0, 1\}^n$ is the vector encoding which variables are exceptional.

[2] $\iota(f) = (\alpha, \mathfrak{a}, \beta, \mathfrak{b}) \in \mathbb{Z}^4_{\geq 0}$ is the measure introduced in Definition 7.1.1.

**Algorithm 7** Main method of local monomialization (for a description see Remark 7.1.2)

**INPUT:** $f = \underline{x}^{\tilde{A}} - \rho\underline{x}^{\tilde{B}}$, mode $\in \{1,2,3,4\}$, where $(\underline{x}) = (x_1, \ldots, x_n)$, $\tilde{A}, \tilde{B} \in \mathbb{Z}^n_{\geq 0}$
**OUTPUT:** list $L$, where $L[i]$ is the data of the $i$-th chart

1: list $Lf$
2: $Lf[1]$ = list$(A, B, C, (\underline{0}))$, where $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B)$ such that $A_iB_i = 0$, for all $i$, and $(\underline{0}) \in \mathbb{Z}^n_{\geq 0}$
3: **if** check_finished$(Lf[1])$ **then** ▷ see Algorithm 8
4: $Lf[2] = Lf[3] = Lf[4] = \varnothing$
5: $L[1] = Lf$
6: **return** $L$
7: $Lf[2] = \iota(f)$ = list$(\alpha, \mathfrak{a}, \beta, \mathfrak{b}) \in \mathbb{Z}^4_{\geq 0}$ ▷ Definition 7.1.1
8: $I$_center = compute_center$(Lf,$ mode$)$
9: $Lf[3] = I$_center
10: intmat path$[2][1] = 0, -1$
11: $Lf[4]$ = path
12: $L[1] = Lf$
13: successors = fill_list_for_next_charts$(L[1], 1)$
14: **for** $L_+ \in$ successors **do**
15: $L[size(L) + 1] = L_+$
16: $i = 2$
17: **while** $i \leq size(L)$ **do**
18: $L[i]$ = transformation$(L[i],$ mode$)$ ▷ see Algorithm 9
19: **if** check_finished$(L[i][1])$ == false **then**
20: successors = fill_list_for_next_charts$(L[i], i)$
21: **for** $L_+ \in$ successors **do**
22: $L[size(L) + 1] = L_+$
23: $i = i + 1$
24: **return** L

[3] $I\_\text{center} \subseteq \{1, \ldots, n\}$ is the index set such that $\langle x_i \mid i \in I\_\text{center}\rangle$ is the ideal defining the center for the next blowup.

[4] a pathmatrix $\begin{pmatrix} 0 & \cdots & x \\ -1 & \cdots & y \end{pmatrix}$ such that $x$ is the number of the predecessor chart. The successors of the predecessors are labeled from 1 to #successors. The number $y$ indicates which of these successors the given chart is. This entry is not important for the monomialization process, but for a later evaluation of the final data to keep track of the global picture.

First, Algorithm 7 performs an initialization, by determining the exponents such that $f = \underline{x}^{\tilde{A}} - \rho\underline{x}^{\tilde{B}} = \underline{x}^{C}(\underline{x}^{A} - \rho\underline{x}^{B})$ has the desired form. Since there are no exceptional divisors yet $E = (\underline{0})$. Then, we check whether $f$ verifies condition (7.0.1) using the method *check_finished*, see Algorithm 8. If (7.0.1) holds, then the binomial is locally monomial and we fill the list $Lf$ with trivial data and return $L$.

If (7.0.1) is not fulfilled, we determine the full data of the chart (lines 7–11). In there, *compute_center*$(Lf, mode)$ is the method determining the index set of the center for the next blowup. The input mode $\in \{1, 2, 3, 4\}$ fixes, which of our four methods is used. In the following sections, we discuss the methods for choosing the center in detail. Furthermore, *intmat* initiates an integer matrix called *path*, which encodes the tree structure of the monomialization process.

Then, the method *fill_list_for_next_charts* copies the data from $L[i]$ (in line 13 for $i = 1$) to a list *successors*, which contains as many charts as needed (depending on the center of the upcoming blowup). The only difference in the data of the charts in *successors* is the adapted *path* matrix which contains the tree of the blowup procedure. After this, the charts from list *successors* are added to the end of $L$. Let us explain this step more in details: If we want to determine the successors for $L[i]$ and if the upcoming blowup has $m$ many charts and if $L$ has $k$ entries in total (with $k \geq i$) before adding the successors to it, then the successors become the entries $L[k+1], \ldots, L[k+m]$ and we extend *path* for the successor $L[k+j]$ by the column $(i, j)^T$ at the end since it is the $j$-th chart of the blowup in $L[i]$, where $j \in \{1, \ldots, m\}$.

In the while-loop (lines 17–23), the data of $L[i]$ (except for *path*) is modified such that it becomes the transformed version of its predecessor with respect to the previously determined center. The transformation algorithm is provided in Algorithm 9 and described in Observation 7.1.3. Finally, we verify whether the data in $L[i]$ fulfills (7.0.1). If so, then the procedure continues with the entry $L[i+1]$ if it exists (i.e., with the next chart which needs to be handled) or it stops if $L$ has

$i$ entries. Otherwise, if (7.0.1) does not hold, we blow-up, the successor charts are stored at the end of the list, analogous to before, and we continue with the chart $i+1$. The while-loop will eventually end since we will show in the following sections that the respective monomialization procedures terminate.

*(End of Remark 7.1.2.)*

**Algorithm 8** check_finished

**INPUT:** list $(A, B, C, E)$ of vectors in $\mathbb{Z}^n_{\geq 0}$ such that $A \neq B$ and $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$

**OUTPUT:** true, if the binomial $x^C(x^A - x^B)$ fulfills (7.0.1); false otherwise

```
1: if min{|A|, |B|} == 0 then
2:     return true
3: if |A| = 1 or |B| = 1 then
4:     if ∃ i : C_i = 0 and (A_i = 1 and |A| = 1) or (B_i = 1 and |B| = 1) then
5:         return true
6: return false
```

Note that in Algorithm 8 it does not matter that the coefficient is 1 instead of $\rho$. The result is the same.

**Observation 7.1.3.** Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}]$ be a binomial with $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. Let us consider how the exponents change along the blowup with center $D = V(x_1, \ldots, x_m)$, for some $m \in \{2, \ldots, n\}$. In the $X_1$-chart, we have

$$x_1 = x'_1,\ x_2 = x'_1x'_2,\ \ldots,\ x_m = x'_1x'_m,\ x_{m+1} = x'_{m+1},\ \ldots,\ x_n = x'_n.$$

On the level of exponents, this provides

$$A = (A_1, A_2, \ldots, A_n) \mapsto \widetilde{A}' := (A_1 + A_2 + \ldots + A_m, A_2, \ldots, A_n)$$

and analogous for $B$ and $C$.

We factor the total transform of $f$ as $\underline{x}'^{C'}(\underline{x}'^{A'} - \rho\underline{x}'^{B'})$ such that $A'_iB'_i = 0$ for all $i \in \{1, \ldots, n\}$. If we set

$$\delta := \min\{A_1 + \ldots + A_m, B_1 + \ldots + B_m\},$$

then we get

$$A' = (A_1 + A_2 + \ldots + A_m - \delta, A_2, \ldots, A_n),$$
$$B' = (B_1 + B_2 + \ldots + B_m - \delta, B_2, \ldots, B_n),$$
$$C' = (C_1 + C_2 + \ldots + C_m + \delta, C_2, \ldots, C_n).$$

The other charts are analogous. Furthermore, it is straight forward to adapt this to blowups in centers of the form $V(x_i \mid i \in I)$, where $I \subseteq \{1, \ldots, n\}$ is not necessarily equal to $\{1, \ldots, m\}$.

This leads to Algorithm 9 for determining the transform of a binomial in a given chart of the blowup in $V(x_i \mid i \in I)$.

**Algorithm 9** transformation (for a description see Remark 7.1.4)

**INPUT:** list $M$, mode $\in \{1, 2, 3, 4\}$, where $M$ is of the same form as $L[i]$ in Remark 7.1.2
**OUTPUT:** list $retList$ which is the transformed variant of chart $M$

1: $(A, B, C, E) = M[1]$
2: path $= M[4]$
3: $I = M[3]$ ▷ the index set of the center
4: $i =$ path[2,ncols(path)] ▷ so $M$ is the $X_i$-chart of the blowup ▷ with center $V(x_i \mid i \in I)$
5: $\delta = \min\{\sum_{j\in I} A_j, \sum_{j\in I} B_j\}$
6: $A_i = \sum_{j\in I} A_j - \delta$
7: $B_i = \sum_{j\in I} B_j - \delta$
8: $C_i = \sum_{j\in I} C_j + \delta$
9: $E_i = 1$
10: $retList[1] = \text{list}(A, B, C, E)$
11: **if** check_finished($retList[1]$) **then**
12: $retList[2] = retList[3] = \varnothing$
13: $retList[4] =$ path;
14: **return** $retList$
15: $retList[2] = \iota(\underline{x}^C(\underline{x}^A - \rho\underline{x}^B))$ ▷ Definition 7.1.1
16: $I\_\text{center} = \text{compute\_center}(retList[1], \text{mode})$
17: $retList[3] = I\_\text{center}$
18: $retList[4] =$ path
19: **return** $retList$

**Remark 7.1.4** (Algorithm 9)**.** The *input* is:

- a list $M$, which represents the data of a chart and hence is of the same form as $L[i]$ in Remark 7.1.2
- an integer mode $\in \{1, 2, 3, 4\}$, which determines the method for choosing the centers;

The *output* of Algorithm 9 is the transformed data of the input chart.

First, we initialize the data (lines 1–4). In particular, we specify the index set $I$ corresponding to the center of the blowup and the element $i \in I$ such that $M$ corresponds to the $X_i$-chart of the blowup. After that, we transform the exponents

$(A, B, C, E)$ as described in Observation 7.1.3 (but now for the general case) and mark the variable $x_i$ as exceptional in lines 6–9. Finally, we check whether the transformed binomial fulfills (7.0.1) and determine the remaining data so that the output data is of the same form as $L[i]$ in Remark 7.1.2. (Note that the path matrix is extended in line 20 of Algorithm 7.)

The only part of implementation which differs in the various modes is the computation of the center. We have seen above that every other method of the implementation only uses the *mode* parameter in order to call the *compute_center*-method which is described later.

## 7.2. Centers contained in the locus of maximal order

In this section we discuss the first of the four variants for the choice of center in details. We fix a binomial

$$f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}] = K[x_1, \ldots, x_n],$$

where $\rho \in K^\times$, $A, B, C \in \mathbb{Z}^n_{\geq 0}$ are such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$ and $K$ is a field. We set

$$g := \underline{x}^A - \rho\underline{x}^B.$$

Observe that the condition $A_iB_i = 0$ implies that at least one of them is zero and hence $x_i$ cannot be factored from $g$.

If $g = 1 - \mu\underline{x}^B$ or $f = \underline{x}^C(x_i - \mu\underline{x}^B)$, for $C, B \in \mathbb{Z}^n_{\geq 0}$ and $\mu \in K^\times$, where we require in the second case that $C_i = 0$ for the given $i$, then $f$ is locally monomial and no blowups are required. Hence, throughout this section, we assume that (7.0.1) is not fulfilled, i.e., that the following condition holds:

$$\begin{cases} \bullet \text{ either } \min\{|A|, |B|\} \geq 2, \text{ or} \\ \bullet \ \min\{|A|, |B|\} = 1 \text{ and } C_i \neq 0 \text{ for every } i \text{ such that } A_i = 1 \text{ if} \\ \quad |A| = 1, \text{ or } B_i = 1 \text{ if } |B| = 1. \end{cases} \tag{7.2.1}$$

Again, let us point out that (7.2.1) does not imply that $f$ is not locally monomial. For example, $x^2 - y^2 \in \mathbb{C}[x, y]$ is monomial after introducing $\widetilde{x} := x + y$, $\widetilde{y} := x - y$, but (7.2.1) holds. Nonetheless, it is not hard to test with a computer whether (7.0.1) is true and thus we admit that we might perform some blowups, which are

not needed.

A common approach in resolution of singularities for a hypersurface $V(g)$ is to consider regular centers contained in its locus of maximal order, see Definition 2.4.8.

If $X$ is a variety, which is not a hypersurface, then the order is not an appropriate measure for the complexity of the singularity, see [35, Example 2.7] or Section 2.4.2.

**Lemma 7.2.1.** *Let $g = \underline{x}^A - \rho\underline{x}^B \in K[\underline{x}]$ be a binomial such that $\min\{|A|, |B|\} \geq 1$ and $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. Let $I \subseteq \{1, \ldots, n\}$ be any subset. Set $X := V(g)$ and $D_I := V(x_i \mid i \in I)$. We have:*

1. $\text{max-ord}(X) = \min\{|A|, |B|\}$.

2. $D_I \subseteq \text{Max-ord}(X) \iff \min\left\{\sum_{i\in I} A_i, \sum_{i\in I} B_i\right\} = \min\left\{|A|, |B|\right\}$.

The condition $\min\{|A|, |B|\} \geq 1$ comes from the fact that we assume (7.2.1) to hold. Notice that it is necessary, e.g., for $g = x_1x_2 - 1$ the maximal order is 1 and not zero, which can be seen by computing the order at $\langle x_1 - 1, x_2 - 1\rangle$, cf. Example 7.6.2.

*Proof of Lemma 7.2.1.* Let $\mathfrak{m} := \langle x_1, \ldots, x_n\rangle$ be the maximal ideal corresponding to the origin. We have $\min\{|A|, |B|\} = \text{ord}_{\mathfrak{m}}(g) \leq \text{max-ord}(X)$.

Suppose there is some prime ideal $\mathfrak{p} \subset K[\underline{x}]$ with $\text{ord}_{\mathfrak{p}}(g) > \text{ord}_{\mathfrak{m}}(g)$. This implies, if we base change to an algebraic closure $\overline{K}$ of $K$, then there is a maximal ideal $\mathfrak{n} \subset \overline{K}[\underline{x}]$ such that $\text{ord}_{\mathfrak{n}}(g) > \text{ord}_{\mathfrak{m}}(g) = \min\{|A|, |B|\}$. Since $\overline{K}$ is algebraically closed and $\mathfrak{n}$ is a maximal ideal, there are $c_1, \ldots, c_n \in \overline{K}$ such that $\mathfrak{n} = \langle x_1 - c_1, \ldots, x_n - c_n\rangle$ by Hilbert's Nullstellensatz.

Set $I_1 := \{i \in \{1, \ldots, n\} \mid A_i \neq 0\}$ and $I_2 := \{i \in \{1, \ldots, n\} \mid B_i \neq 0\}$ . Since $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$, we have $I_1 \cap I_2 = \varnothing$ and it makes sense to define $y_i := x_i - c_i$ for $i \in I_1$ and $z_i := x_i - c_i$ for $i \in I_2$. This provides

$$\underline{x}^A - \rho\underline{x}^B = \prod_{i\in I_1}(y_i + c_i)^{A_i} - \rho\prod_{i\in I_2}(z_i + c_i)^{B_i} = \sum_{\alpha\in\mathbb{Z}_{\geq 0}^{|I_1|}} \lambda_\alpha(\underline{c})\underline{y}^\alpha - \sum_{\beta\in\mathbb{Z}_{\geq 0}^{|I_2|}} \mu_\beta(\underline{c})\underline{z}^\beta,$$

where the coefficients $\lambda_\alpha(\underline{c}), \mu_\beta(\underline{c}) \in \overline{K}$ fulfill $\lambda_A(\underline{c}) = 1$, $\mu_B(\underline{c}) = \rho$ and $\lambda_\alpha(\underline{c}) = \mu_\beta(\underline{c}) = 0$ if $|\alpha| \geq |A|$ and $\alpha \neq A$, resp. if $|\beta| \geq |B|$ and $\beta \neq B$ (and we use the obvious notation $\underline{c}, \underline{y}, \underline{z}$). In order to have $\text{ord}_{\mathfrak{n}}(g) > \min\{|A|, |B|\}$, all terms $y^Cz^D$ with $|C| + |D| \leq \min\{|A|, |B|\}$ have to cancel out. This is impossible since the variables appearing in the products are disjoint and $\lambda_A(\underline{c})\mu_B(\underline{c}) \neq 0$. Thus, we arrived to a contradiction and (1) follows.

Let us come to part (2). Set $d := \text{max-ord}(X) = \min\{|A|, |B|\}$. The condition $D_I = V(x_i \mid i \in I) \subseteq \text{Max-ord}(X)$ is equivalent to

$$\underline{x}^A - \rho \underline{x}^B \in \langle x_i \mid i \in I \rangle^d \setminus \langle x_i \mid i \in I \rangle^{d+1}.$$

The latter is equivalent to $\sum_{i \in I} A_i \geq d$ and $\sum_{i \in I} B_i \geq d$, and equality has to hold for one of them. Hence, (2) follows. □

**Example 7.2.2.** Let $g = x_1^3 x_2 - x_3^3 x_4^4$ and set $X := V(g) \subset \mathbb{A}^4_K$. Using the previous lemma, we get that $\text{max-ord}(X) = 4$, $V(x_1, x_2, x_4) \subseteq \text{Max-ord}(X)$, while $V(x_1, x_2, x_3) \not\subseteq \text{Max-ord}(X)$.

Let us blow-up with center $D := V(x_1, x_2, x_3, x_4) \in \text{Max-ord}(X)$, the origin of $\mathbb{A}^4_K$. In the $X_3$-chart, we have $(x_1, x_2, x_3, x_4) = (x_1' x_3', x_2' x_3', x_3', x_3' x_4')$ and the total transform of $g$ is $x_3'^4(x_1'^3 x_2' - x_3'^3 x_4'^4)$. We obtain essentially the same binomial and no improvement is detected. The reason for this is that the center has been chosen too small.

We leave it as an exercise to the reader to verify that the maximal order decreases at every chart after blowing up with center $V(x_1, x_2, x_4)$.

Let us now describe the method for choosing the center for a binomial using the locus of maximal order.

**Construction 7.2.3.** Let $f = \underline{x}^C(\underline{x}^A - \rho \underline{x}^B) \in K[\underline{x}]$ with $\rho \in K^\times$ and $A, B, C \in \mathbb{Z}^n_{\geq 0}$ such that $A_i B_i = 0$ for all $i \in \{1, \ldots, n\}$. Set $g := \underline{x}^A - \rho \underline{x}^B$. Assume that hypothesis (7.2.1) holds. We choose $I \subseteq \{1, \ldots, n\}$ such that

$$\min\big\{\sum_{i \in I} A_i, \sum_{i \in I} B_i\big\} = \min\big\{|A|, |B|\big\}$$

and we require additionally that

$$\forall\, j \in I \,:\, \min\big\{\sum_{i \in I \setminus \{j\}} A_i, \sum_{i \in I \setminus \{j\}} B_i\big\} < \min\big\{|A|, |B|\big\}. \tag{7.2.2}$$

Then, the center for the next blowup is $D_I := V(x_i \mid i \in I)$.

By Lemma 7.2.1, the center $D_I$ is contained in the maximal order locus of $V(g)$. On the other hand, (7.2.2) guarantees that $D_I$ is not too small so that an improvement can be detected.

**Example 7.2.4.** Let $f = \underline{x}^A - \underline{x}^B = x_1^3 x_2^2 - x_3^5 x_4 \in K[x_1, x_2, x_3, x_4]$. Since $\min\{|A|, |B|\} = |A| = 5$, we have $\{1, 2\} \subseteq I$ for every $I \subseteq \{1, \ldots, 4\}$ fulfilling

the conditions of Construction 7.2.3. Furthermore, $I' := \{1,2,3,4\}$ does not fulfill (7.2.2) for $j = 4$. Therefore, the unique center determined by Construction 7.2.3 is $V(x_1, x_2, x_3)$.

Clearly, the subset $I \subseteq \{1, \ldots, n\}$ is not unique in general and we may have to make a choice, as the following example shows. As explained in the introduction, we do not require that our procedure provides a global monomialization of $V(f)$. Therefore we may allow to make choices as long as we can prove the termination of the resulting procedure (Proposition 7.2.7).

**Example 7.2.5.** Let $f = x_1x_2^2 - x_3^3x_4^2x_5 \in K[x_1, \ldots, x_5]$. Since no $x_i$ can be factored in $f$, we have $g = f$. The maximal order of $g$ is three and $D_1 := V(x_1, x_2, x_3)$ and $D_2 := V(x_1, x_2, x_4, x_5)$ are the possible choices for the center following Construction 7.2.3.

1. Blow-up with center $D_1$. In the $X_3$-chart, we have

$$(x_1, x_2, x_3, x_4, x_5) = (x_1'x_3', x_2'x_3', x_3', x_4', x_5').$$

   Hence, the total transform of $f$ is $f = x_3'^3(x_1'x_2'^2 - x_4'^2x_5') = x_3'^3g'$, where we define $g' := x_1'x_2'^2 - x_4'^2x_5'$. (Note that $g'$ fulfills the property that no $x_i'$ divides $g'$.) We have max-ord$(g) = 3 =$ max-ord$(g')$ and $|B'| = 3 = |A| < |B| = 6$.

2. Blow-up with center $D_2$. In the $X_4$-chart, we have

$$(x_1, x_2, x_3, x_4, x_5) = (\widetilde{x}_1\widetilde{x}_4, \widetilde{x}_2\widetilde{x}_4, \widetilde{x}_3, \widetilde{x}_4, \widetilde{x}_4\widetilde{x}_5).$$

   (For a better distinction to (1), we use $\widetilde{*}$ instead of $*'$ for the coordinates here.) The total transform of $f$ is $f = \widetilde{x}_4^3(\widetilde{x}_1\widetilde{x}_2^2 - \widetilde{x}_3^3\widetilde{x}_5)$. Thus, we set $\widetilde{g} := \widetilde{x}_1\widetilde{x}_2^2 - \widetilde{x}_3^3\widetilde{x}_5$. We get max-ord$(\widetilde{g}) = 3 =$ max-ord$(g)$ and $|\widetilde{B}| = 4 < 6 = |B|$. (The situation in the $X_5$-chart is analogous.)

On the other hand, in both cases, one can show that the maximal order is strictly smaller than three if we consider the $X_1$- or the $X_2$-chart.

For the general case, we have to introduce a measure which detects the improvement.

**Definition 7.2.6.** Let $g = \underline{x}^A - \rho\underline{x}^B \in K[\underline{x}]$ with $\rho \in K^\times$ and $A, B \in \mathbb{Z}_{\geq 0}^n$ such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. We define

$$\mathrm{inv}(g) := \big(\ \min\{|A|, |B|\},\ \max\{|A|, |B|\}\ \big) \in \mathbb{Z}_{\geq 0}^2.$$

Here, we equip $\mathbb{Z}^2_{\geq 0}$ with the lexicographical ordering $\geq_{\ell p}$.

In the above example, we have $\operatorname{inv}(g) = (3,6)$, $\operatorname{inv}(g') = (3,3) <_{\ell p} \operatorname{inv}(g)$, and $\operatorname{inv}(\widetilde{g}) = (3,4) <_{\ell p} \operatorname{inv}(g)$.

In fact, $\operatorname{inv}(g) = (\text{max-ord}(g), \text{max-ord}(g) \cdot \delta(g))$, where $\delta(g)$ is a known secondary invariant to measure the complexity of a given singularity, e.g., see [57, p. 120, where it is called $\gamma$], [25, Theorem 3.18] or [21, Corollary 5.1].

**Proposition 7.2.7.** *Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}] = K[x_1, \ldots, x_n]$ with $\rho \in K^\times$ and $A, B, C \in \mathbb{Z}^n_{\geq 0}$ such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. Set $g := \underline{x}^A - \rho\underline{x}^B$. Let $\pi\colon B\ell_{D_I}(\mathbb{A}^n_K) \to \mathbb{A}^n_K$ be the blowup in a center $D_I$, which fulfills the properties as in Construction 7.2.3. For every standard chart $U_{x_j} := D_+(X_j) \cong \mathbb{A}^n_K$, $j \in I$, we have*

$$\operatorname{inv}(g') <_{\ell p} \operatorname{inv}(g),$$

*where $f = \underline{x}'^{C'}(\underline{x}'^{A'} - \rho\underline{x}'^{B'}) \in K[\underline{x}']$ with $A'_iB'_i = 0$ for all $i \in \{1, \ldots, n\}$, $g' := \underline{x}'^{A'} - \rho\underline{x}'^{B'}$ is the* strict transform *of $g$, and $(\underline{x}') = (x'_1, \ldots, x'_n)$ are the coordinates in $U_{x_j}$.*

*In particular, the local monomialization process obtained by choosing the centers as in Construction 7.2.3 terminates.*

*Proof.* If hypothesis (7.2.1) does not hold for $f$, then $f$ is already locally monomial and there is nothing to show. Hence, suppose that (7.2.1) holds. Without loss of generality, we have $|A| \leq |B|$. After relabeling the variables $(x_1, \ldots, x_n)$ we may assume that

$$\{i \in \{1, \ldots, n\} \mid A_i \neq 0\} = \{1, \ldots, m\},$$

for some $m < n$. Thus, we have $\{1, \ldots, m\} \subseteq I$, i.e., $D_I \subseteq V(x_1, \ldots, x_m)$.

Let us consider the $X_i$-chart of the blowup with center $D_I$. We distinguish two cases, $i \leq m$ and $i > m$.

Assume that $i \leq m$, i.e., $A_i \neq 0$. Using the notation of the proposition, we have

$$g' = \underline{x}'^A (x'_i)^{-A_i} - \rho\underline{x}'^B (x'_i)^{|B_I|-|A|} \in K[\underline{x}'],$$

where $|B_I| := \sum_{j \in I} B_j$. Notice that $|B_I| - |A| \geq 0$. Our hypothesis $|A| \leq |B|$ implies that $\operatorname{inv}(g') = (|A| - A_i, |B| + |B_I| - |A|)$ in this case. Since $A_i \neq 0$, we have $|A| > |A| - A_i$ and hence $\operatorname{inv}(g') <_{\ell p} \operatorname{inv}(g)$.

Now, suppose that $i \in \{m+1, \ldots n\}$, i.e., $B_i \neq 0$. We get

$$g' = \underline{x}'^{A'} - \rho\underline{x}'^{B'} = \underline{x}'^A - \rho\underline{x}'^B x_i'^{|B_I|-|A|-B_i} \in K[\underline{x}'].$$

First, we observe that $\min\{|A'|,|B'|\} \leq |A'| = |A|$. If the inequality is strict, we obtain $\mathrm{inv}(g') <_{\ell p} \mathrm{inv}(g)$ as desired. Hence, let us assume that $\min\{|A'|,|B'|\} = |A'| = |A|$. The claim follows if we can show $|B'| < |B|$. By (7.2.2) and the hypothesis $|A| \leq |B|$, we have that $|B_I| - B_i < |A|$ and therefore we get $|B'| = |B| + |B_I| - |A| - B_i < |B|$ and in particular $\mathrm{inv}(g') <_{\ell p} \mathrm{inv}(g)$.

Since the improvement of inv(.) is strict and since inv(.) takes values in $\mathbb{Z}^2_{\geq 0}$, the local monomialization procedure using centers of the kind in Construction 7.2.3 ends after finitely many steps. □

Given a binomial in $K[\underline{x}] = K[x_1,\ldots,x_n]$, we provide in Algorithm 10 a method to determine a subset $I \subseteq \{1,\ldots,n\}$ fulfilling the conditions of Construction 7.2.3. Therefore, $V(x_i \,|\, i \in I)$ will be our center contained in the maximal order locus of the binomial.

**Remark 7.2.8** (Algorithm 10)**.** The *input* is:

- a list $M$, which represents the data of a chart and hence is of the same form as $L[i]$ in Remark 7.1.2
- the integer mode $=$ 1, which tells us to choose the center as in Construction 7.2.3;

The *output* of Algorithm 10 is the index set $I \subseteq \{1,\ldots,n\}$ determining the next center for the monomialization procedure.

First, we initialize the data (lines 1–3). We introduce the exponents of the binomial $\underline{x}^C(\underline{x}^A - \rho\underline{x}^B)$ in the given chart. Further, we introduce two auxiliary sets $I$ and $J$, where $I$ will become the output set.

Then we perform a case distinction depending on whether $|A| < |B|$ (lines 4–14), or $|A| > |B|$ (lines 15–25), or $|A| = |B|$ (lines 26–28).

Suppose $|A| < |B|$. Then, the maximal order of $\underline{x}^A - \rho\underline{x}^B$ is $|A|$ and every $i$ with $A_i > 0$ contributes to the index set of the center (line 5). After that, we sum up the elements of $B$ until the resulting sum is $\geq |A|$ (lines 6–9). Within this, we collect in $J$ the indices $j$ with $B_j > 0$ appearing in the sum. At this moment, the index set $\widetilde{I} := I \cup J$ fulfills the first condition of Construction 7.2.3, $\min\big\{\sum_{i\in\widetilde{I}} A_i, \sum_{i\in\widetilde{I}} B_i\big\} = \min\big\{|A|,|B|\big\}$. But the second condition (7.2.2) does not necessarily hold, i.e., the number of elements in $\widetilde{I}$ might be too large. Hence, we remove step-by-step elements from $J$, without destroying the first condition, until (7.2.2) holds (lines 10–12).

The case $|A| > |B|$ is analogous, we only have to interchange the role of $A$ and $B$. Finally, if $|A| = |B|$, all variables appearing in the binomial with non-zero exponent have to be contained in the ideal of the center.

**Algorithm 10** compute center (in the locus of maximal order)

**INPUT:** list $M$, mode $= 1$, where $M$ is of the same form as $L[i]$ in Remark 7.1.2
**OUTPUT:** $I \subseteq \{1, \ldots, n\}$ such that $V(x_i \,|\, i \in I)$ is the next center in the monomialization process

1: $(A, B, C, E) = M[1]$
2: $I = \varnothing$, $J = \varnothing$
3: $\mathcal{A} = \{i \mid A_i > 0\}$, $\mathcal{B} = \{i \mid B_i > 0\}$
4: **if** $|A| < |B|$ **then**
5: $\quad I = \mathcal{A}$
6: $\quad$ **for** $b \in \mathcal{B}$ **do**
7: $\quad\quad J = J \cup \{b\}$
8: $\quad\quad$ **if** $\sum_{j \in J} B_j \geq |A|$ **then**
9: $\quad\quad\quad$ **break**
10: $\quad$ **for** $i \in J$ **do**
11: $\quad\quad$ **if** $\sum_{j \in J \setminus \{i\}} B_j \geq |A|$ **then**
12: $\quad\quad\quad J = J \setminus \{i\}$
13: $\quad I = I \cup J$
14: $\quad$ **return** $I$
15: **else if** $|B| < |A|$ **then**
16: $\quad I = \mathcal{B}$
17: $\quad$ **for** $a \in \mathcal{A}$ **do**
18: $\quad\quad J = J \cup \{a\}$
19: $\quad\quad$ **if** $\sum_{j \in J} A_j \geq |B|$ **then**
20: $\quad\quad\quad$ **break**
21: $\quad$ **for** $i \in J$ **do**
22: $\quad\quad$ **if** $\sum_{j \in J \setminus \{i\}} A_j \geq |B|$ **then**
23: $\quad\quad\quad J = J \setminus \{i\}$
24: $\quad I = I \cup J$
25: $\quad$ **return** $I$
26: **else if** $|A| = |B|$ **then**
27: $\quad I = \mathcal{A} \cup \mathcal{B}$
28: $\quad$ **return** $I$

Clearly, the choice of center depends on the ordering of the variables. In Example 7.2.5, we obtain the center $D_1$ if we choose the ordering $(x_1, x_2, x_3, x_4, x_5)$, while we get $D_2$ for the ordering $(x_1, x_2, x_4, x_5, x_3)$ (using the notation of the example).

## 7.3. Centers of codimension two

Let us come to the second method for choosing the center. Recall that $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B)$ and $g = \underline{x}^A - \rho\underline{x}^B$ such that no $x_i$ can be factored from $g$. In contrast to the previous method, we may neglect the connection to the singularities of $V(g)$ and choose centers of minimal codimension. This has the benefit that we reduce the number of charts which we have to control after a single blowup. Hence, the idea is to take $i, j \in \{1, \ldots, n\}$ such that $A_i \neq 0$ and $B_j \neq 0$, which provides the center $D = V(x_i, x_j)$. An additional requirement, which we shall need in order to detect an improvement after the blowup, is that the exponents $A_i$ and $B_j$ are maximal among the possible choices.

**Example 7.3.1.** (cf. Example 7.2.4) Let $X = V(f)$ be the hypersurface described by the binomial $f = x_1^3x_2^2 - x_3^5x_4 \in K[x_1, x_2, x_3, x_4]$. The maximal exponents appearing in the monomials are $A_1 = 3$ and $B_3 = 5$. Therefore, we will choose $D = V(x_1, x_3)$ as the center for the next blowup.
In comparison to Example 7.2.4 we see that we can reduce the number of successor charts by choosing a center of codimension 2.

Note that the maximal order is not an appropriate measure to detect the improvement along a blowup of the given type.

**Example 7.3.2.** Let $f = g = x_1^a x_2^a x_3^a - x_4^b \in K[x_1, x_2, x_3, x_4]$ with $a, b \in \mathbb{Z}_{\geq 2}$ such that $a < b < 2a$. (For example, take $a = 3$ and $b = 5$). Observe that max-ord$(g) = b$. We choose the center $V(x_1, x_4)$. In the $X_1$-chart, we have

$$(x_1, x_2, x_3, x_4) = (x_1', x_2, x_3, x_1'x_4')$$

and the total transform of $f$ is $f' = x_1'^a(x_2^a x_3^a - x_1'^{b-a}x_4'^b) = x_1'^a g'$, where we define

$$g' := \underline{x}'^{A'} - \underline{x}'^{B'} := x_2^a x_3^a - x_1'^{b-a}x_4'^b.$$

Since $a < b$, we have $|B'| = 2b - a > b$, while $b < 2a$ implies $|A'| = 2a > b$. Therefore, we have max-ord$(g') >$ max-ord$(g)$, which is not a surprise since the center $V(x_1, x_4)$ is not contained in Max-ord$(g)$.

Note that the same happens if we blow-up one of the other reasonable centers of codimension two, which are $V(x_2, x_4)$ and $V(x_3, x_4)$.

We use $\iota(g) = (\alpha(g), \mathfrak{a}(g), \beta(g), \mathfrak{b}(g)) \in \mathbb{Z}^4_{\geq 0}$ of Definition 7.1.1 to deduce the termination for the present method of monomialization. In Example 7.3.2, we have $\iota(g') = (a, 2, b, 1) <_{\ell p} (a, 3, b, 1) = \iota(g)$. In general, not any codimension two center provides an improvement of $\iota(.)$.

**Example 7.3.3.** Consider the binomial $f = g = x_1^a - x_2^b x_3^c \in K[x_1, x_2, x_3]$ with $a, b, c \in \mathbb{Z}_{\geq 2}$ and $b \leq c$. We have

$$\iota(g) = \begin{cases} (a, 1, c, 1), & \text{if } b < c, \\ (a, 1, c, 2), & \text{if } b = c. \end{cases}$$

Suppose that $b < c$. Let us blow-up with center $V(x_1, x_2)$, which does not fulfill the additional hypothesis that the corresponding exponents in $g$ are maximal. For simplicity, we define $m := \min\{a, b\}$. In the $X_2$-chart, the total transform of $f$ is $f = x_2'^m(x_1'^a x_2'^{a-m} - x_2'^{b-m} x_3^c)$, which provides $g' = x_1'^a x_2'^{a-m} - x_2'^{b-m} x_3^c$ and $\iota(g') = (a, 1, c, 1) = \iota(g)$.

In order to guarantee a decrease of $\iota(.)$, we choose the center as follows:

**Construction 7.3.4.** Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}]$ with $\rho \in K^\times$ and $A, B, C \in \mathbb{Z}^n_{\geq 0}$ such that $A_i B_i = 0$ for all $i \in \{1, \ldots, n\}$. Set $g := \underline{x}^A - \rho\underline{x}^B$. Assume that hypothesis (7.2.1) holds. If $\iota(g) = (\alpha, \mathfrak{a}, \beta, \mathfrak{b}) \in \mathbb{Z}^4_{\geq 0}$, then we choose $j_1, j_2 \in \{1, \ldots, n\}$ such that $A_{j_1} = \alpha$ and $B_{j_2} = \beta$. The center for the next blowup is then $D_I = V(x_{j_1}, x_{j_2})$, for $I = \{j_1, j_2\}$.

Already in Example 7.3.2, we have seen that the center described in the above construction is not unique. But, analogous to Proposition 7.2.7, we can prove the following result.

**Proposition 7.3.5.** *Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}] = K[x_1, \ldots, x_n]$ with $\rho \in K^\times$ and $A, B, C \in \mathbb{Z}^n_{\geq 0}$ such that $A_i B_i = 0$ for all $i \in \{1, \ldots, n\}$. Set $g := \underline{x}^A - \rho\underline{x}^B$. Let $\pi\colon B\ell_{D_I}(\mathbb{A}^n_K) \to \mathbb{A}^n_K$ be the blowup in a center $D_I$, which fulfills the properties as in Construction 7.3.4. For every standard chart $U_{x_j} := D_+(X_j) \cong \mathbb{A}^n_K$, $j \in I$, we have*

$$\iota(g') <_{\ell p} \iota(g),$$

*where $f = \underline{x}'^{C'}(\underline{x}'^{A'} - \rho\underline{x}'^{B'}) \in K[\underline{x}']$ with $A'_i B'_i = 0$ for all $i \in \{1, \ldots, n\}$, $g' := \underline{x}'^{A'} - \rho\underline{x}'^{B'}$, and $(\underline{x}') = (x'_1, \ldots, x'_n)$ are the coordinates in $U_{x_j}$.*

*In particular, the local monomialization process obtained by choosing the centers as in Construction 7.3.4 terminates.*

*Proof.* We may assume $j_1 = 1$ and $j_2 = 2$ after relabeling the variables $(x_1, \ldots, x_n)$. Hence, the center is $V(x_1, x_2)$. Since the exponents $A_+ := (A_3, \ldots, A_n)$ and $B_+ := (B_3, \ldots, B_n)$ are not changed by the blowup, we use the abbreviation

$$g = x_1^{\alpha} \underline{x}_+^{A_+} - \rho x_2^{\beta} \underline{x}_+^{B_+},$$

where $\iota(g) = (\alpha, \mathfrak{a}, \beta, \mathfrak{b})$ and $\underline{x}_+ := (x_3, \ldots, x_n)$. Without loss of generality, we assume $\alpha \leq \beta$.

In the $X_1$-chart, the total transform of $f$ provides $g' = \underline{x}'^{A'} - \rho \underline{x}'^{B'} := \underline{x}'^{A_+}_+ - \rho x_1'^{\beta-\alpha} x_2'^{\beta} \underline{x}'^{B_+}_+$. Therefore, $\alpha(g') < \alpha = \alpha(g)$, if $\mathfrak{a} = \mathfrak{a}(g) = 1$, or $(\alpha(g'), \mathfrak{a}(g')) = (\alpha, \mathfrak{a} - 1)$ otherwise. So, we get $\iota(g') <_{\ell p} \iota(g)$.

Let us consider the $X_2$-chart of the blowup. In there, we obtain $g' = \underline{x}'^{A'} - \rho \underline{x}'^{B'} := x_1'^{\alpha} \underline{x}'^{A_+}_+ - \rho x_2'^{\beta-\alpha} \underline{x}'^{B_+}_+$. In order to show that $\iota(.)$ improves, we first notice that $(\alpha(g'), \mathfrak{a}(g')) = (\alpha(g), \mathfrak{a}(g))$ since $A' = A$. Analogous to the $X_1$-chart, we have $\beta(g') < \beta = \beta(g)$, if $\mathfrak{b} = \mathfrak{b}(g) = 1$, or $(\beta(g'), \mathfrak{b}(g')) = (\beta, \mathfrak{b} - 1)$ otherwise. Hence, we have in all cases that $\iota(g') <_{\ell p} \iota(g)$.

Since the improvement of $\iota(.)$ is strict and since $\iota(.)$ takes values in $\mathbb{Z}^4_{\geq 0}$, the local monomialization procedure using centers of the kind in Construction 7.3.4 ends after finitely many steps. □

The blowup $\pi\colon B\ell_{D_I}(\mathbb{A}^n_K) \to \mathbb{A}^n_K$ with center $D_I$ chosen the following Construction 7.3.4 is not necessarily an isomorphism outside the singular locus of $V(g)$.

**Example 7.3.6.** Let $f = g = x_1 x_2 - x_3^5 \in K[x_1, x_2, x_3]$. Construction 7.3.4 provides the possible centers $V(x_1, x_3)$ and $V(x_2, x_3)$. Both are not contained in the singular locus of $V(g)$, which is $\mathrm{Sing}(V(g)) = V(x_1, x_2, x_3)$. Therefore, the potential centers are strictly larger than the singular locus and the corresponding blowup morphisms are not an isomorphisms outside of the singular locus.

In Algorithm 11, we provide an implementation of Construction 7.3.4 to choose a center of codimension 2.

**Remark 7.3.7** (Algorithm 11)**.** The input and the output are of the same form as in Algorithm 10 (see Remark 7.2.8) with the only difference that the input mode is 2. We initialize the data, by fixing the names of the exponents of the binomial $f = \underline{x}^C(\underline{x}^A - \rho \underline{x}^B)$ of this chart and determining the maximal exponents $\alpha$ resp. $\beta$

**Algorithm 11** compute center (with codimension 2)

**INPUT:** list $M$, mode $= 2$, where $M$ is of the same form as $L[i]$ in Remark 7.1.2
**OUTPUT:** $I \subseteq \{1,\dots,n\}$ such that $V(x_i \,|\, i \in I)$ is the next center in the monomialization process
1: $(A,B,C,E) = M[1]$
2: $\alpha = \max\{A_i\}$, $\beta = \max\{B_i\}$
3: $I = \{\, \min\{i \mid A_i = \alpha\}, \min\{i \mid B_i = \beta\}\,\}$
4: **return** $I$

appearing on each side. Then, we choose $i$ and $j$ minimal in $\{1,\dots,n\}$ such that $A_i = \alpha$ and $B_j = \beta$ achieve the maximal values.

Note that we cannot have $\alpha = 0$ or $\beta = 0$ in this algorithm since we tested whether (7.0.1) holds before applying *compute_ center* in Algorithm 7 (line 3) resp. in Algorithm 9 (line 11).

## 7.4. Centers of minimal codimension contained in the singular locus

The third variant is a mixture of the first two. Namely, we want to choose centers as large as possible (as in Construction 7.3.4), but we require additionally that along the monomialization process the centers are contained in the singular locus of the factor, which we obtain after factoring the monomial part (similar to Construction 7.2.3). If this singular locus is empty, we follow the method of section 7.3 (Construction 7.3.4). Within this, we have to distinguish several cases.

**Construction 7.4.1.** Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}]$ with $\rho \in K^\times$ and $A,B,C \in \mathbb{Z}^n_{\geq 0}$ such that $A_iB_i = 0$ for all $i \in \{1,\dots,n\}$. Set $g := \underline{x}^A - \rho\underline{x}^B$. Assume that hypothesis (7.2.1) holds. Let $\iota(g) = (\alpha,\mathfrak{a},\beta,\mathfrak{b}) \in \mathbb{Z}^4_{\geq 0}$.

(i) If $\min\{\alpha,\beta\} \geq 2$ or $\min\{|A|,|B|\} = 1$, choose $D_I = V(x_{j_1},x_{j_2})$, for $I = \{j_1,j_2\}$ as in Construction 7.3.4, for the center of the blowup.

(ii) If $\alpha = 1$, $\beta \geq 2$ and $\min\{|A|,|B|\} \geq 2$, choose $j_1,j_2,j_3 \in \{1,\dots,n\}$ with $A_{j_1} = A_{j_2} = 1$ and $B_{j_3} = \beta$. The center of the next blowup is $D_I = V(x_{j_1},x_{j_2},x_{j_3})$, for $I = \{j_1,j_2,j_3\}$.

(iii) If $\alpha \geq 2$, $\beta = 1$ and $\min\{|A|,|B|\} \geq 2$, choose $j_1,j_2,j_3 \in \{1,\dots,n\}$ with $A_{j_1} = \alpha$ and $B_{j_2} = B_{j_3} = 1$. The center of the next blowup is $D_I = V(x_{j_1},x_{j_2},x_{j_3})$, for $I = \{j_1,j_2,j_3\}$.

(iv) If $\alpha = \beta = 1$ and $\min\{|A|, |B|\} \geq 2$, choose $j_1, j_2, j_3, j_4 \in \{1, \ldots, n\}$ with $A_{j_1} = A_{j_2} = B_{j_3} = B_{j_4} = 1$. The center of the next blowup is $D_I = V(x_{j_1}, x_{j_2}, x_{j_3}, x_{j_4})$, for $I = \{j_1, j_2, j_3, j_4\}$.

We say $(f, g)$ *is in case (∗)* if condition $(*)$ is fulfilled, where $* \in \{\mathrm{i}, \mathrm{ii}, \mathrm{iii}, \mathrm{iv}\}$.

Observe that for $f = x_1(x_1 - x_2x_3)$ we are in case (i), while $f = x_1(x_1 - x_2)$ is monomial. Furthermore, $f = x_1x_2x_3 - x_4x_5x_6$ is case (iv) and there are several choices for the center.

**Proposition 7.4.2.** *Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[x_1, \ldots, x_n]$ with $\rho \in K^\times$ and $A, B, C \in \mathbb{Z}^n_{\geq 0}$ such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. Let $g = \underline{x}^A - \rho\underline{x}^B$ and $\iota(g) = (\alpha, \mathfrak{a}, \beta, \mathfrak{b}) \in \mathbb{Z}^4_{\geq 0}$. Let $\pi\colon B\ell_{D_I}(\mathbb{A}^n_K) \to \mathbb{A}^n_K$ be the blowup in a center $D_I$, which fulfills the properties as in Construction 7.4.1. For every standard chart $U_{x_j} := D_+(X_j) \cong \mathbb{A}^n_K$, $j \in I$, we have*

$$\iota(g') <_{\ell p} \iota(g),$$

*where $f = \underline{x}'^{C'}(\underline{x}'^{A'} - \rho\underline{x}'^{B'}) \in K[\underline{x}']$ with $A'_iB'_i = 0$ for all $i \in \{1, \ldots, n\}$, $g' := \underline{x}'^{A'} - \rho\underline{x}'^{B'}$, and $(\underline{x}') = (x'_1, \ldots, x'_n)$ are the coordinates in $U_{x_j}$.*

*In particular, the local monomialization process obtained by choosing the centers as in Construction 7.4.1 terminates.*

*Proof.* We show the result by going through all cases of Construction 7.4.1. First, if $(f, g)$ *is in case (i)*, then we have $\iota(g') <_{\ell p} \iota(g)$ by Proposition 7.3.5.

Next, we assume that $(f, g)$ *is in case (ii)*, i.e., $\alpha = 1, \beta \geq 2$ and $\min\{|A|, |B|\} \geq 2$. We relabel the variables, so that we have $B_{m+1} = \beta$ and $\{i \in \{1, \ldots, n\} \mid A_i = 1\} = \{1, \ldots, m\}$ for some $2 \leq m < n$. In particular,

$$g = x_1x_2 \cdots x_m - \rho\underline{x}^B.$$

Hence, without loss of generality, the center is $D_I := V(x_1, x_2, x_{m+1})$.

The $X_1$- and the $X_2$-chart are analogous, so we consider only one of them. In the $X_1$-chart, we get (using the notation of the statement of the proposition)

$$g' = x'_2 \cdots x'_m - \rho\, x'^{\beta-2}_1\, \underline{x}'^B.$$

We see that $(\alpha(g'), \mathfrak{a}(g')) = (1, m-1) <_{\ell p} (1, m) = (\alpha(g), \mathfrak{a}(g))$. This implies the desired decrease $\iota(g') <_{\ell p} \iota(g)$. Observe that $(\beta(g'), \mathfrak{b}(g')) = (\beta(g), \mathfrak{b}(g))$ did not change.

On the other hand, using the notation $\underline{x}^B = x_{m+1}^{\beta}\underline{x}_+^{B_+}$, we obtain in the $X_{m+1}$-chart

$$g' = x_1' \cdots x_m' - \rho\, x_{m+1}'^{\beta-2}\, \underline{x}_+'^{B_+}.$$

We have $(\alpha(g'), \mathfrak{a}(g')) = (\alpha(g), \mathfrak{a}(g))$ and $\beta(g') \leq \beta(g)$. Either the inequality is strict or we have equality and $\mathfrak{b}(g') = \mathfrak{b}(g) - 1$ since the power of $x_{m+1}'$ decreased strictly. In both cases, we get $\iota(g') <_{\ell p} \iota(g)$.

The case that $(f, g)$ *is in case (iii)* is analogous to the previous one. One only has to interchange the role of $A$ and $B$ and take into account that $(\alpha(g'), \mathfrak{a}(g')) = (\alpha(g), \mathfrak{a}(g))$ in the $X_1$-(resp. $X_2$-)chart.

Finally, suppose that $(f, g)$ *is in case (iv)*. After relabeling the variables we get

$$g = x_1 \cdots x_m - \rho x_{m+1} \cdots x_{m+\ell},$$

for some $2 \leq m < n$ and $2 \leq \ell < n$ with $m + \ell \leq n$. Without loss of generality, the center is $D_I = V(x_1, x_2, x_{m+1}, x_{m+2})$. So, we have to consider four charts. In the $X_1$-chart we get

$$g' = \underline{x}'^{A'} - \rho\underline{x}'^{B'} = x_2' \cdots x_m' - \rho x_{m+1}' \cdots x_{m+\ell}'.$$

This implies that $(\alpha(g'), \mathfrak{a}(g')) = (\alpha(g), \mathfrak{a}(g) - 1)$ and (since $B' = B$) we also have $(\beta(g'), \mathfrak{b}(g')) = (\beta(g), \mathfrak{b}(g))$. In particular, we get $\iota(g') <_{\ell p} \iota(g)$. The other three charts are analogous.

Since the improvement of $\iota(.)$ is strict in every case and since $\iota(.)$ takes values in $\mathbb{Z}^4_{\geq 0}$, the local monomialization procedure using centers of the kind in Construction 7.4.1 ends after finitely many steps. □

In Algorithm 12 we discuss an implementation for the choice of the center following Construction 7.4.1.

**Remark 7.4.3** (Algorithm 12)**.** The input and the output are of the same form as in Algorithm 10 (see Remark 7.2.8) with the only difference that the input mode is 3. First, we initialize the data and determine the minimal indices $i_1$ and $i_2$, for which the maximal entry of $A$ resp. $B$ is achieved, where $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B)$ is the monomial in the given chart.

If $V(x_{i_1}, x_{i_2})$ is contained in the singular locus of $V(\underline{x}^A - \rho\underline{x}^B)$ or if $f$ is of the form $f = \underline{x}^C(x_j - \underline{x}^D)$, for $j \in \{i_1, i_2\}$ and $D \in \{A, B\}$ the corresponding element, then the algorithm returns $\{i_1, i_2\}$ as the index set for the upcoming center (line 5–6). This is case (i) of Construction 7.4.1.

**Algorithm 12** compute center of minimal codimension contained in the singular locus

**INPUT:** list $M$, mode $= 3$, where $M$ is of the same form as $L[i]$ in Remark 7.1.2
**OUTPUT:** $I \subseteq \{1, \ldots, n\}$ such that $V(x_i \,|\, i \in I)$ is the next center in the monomialization process

1: $(A, B, C, E) = M[1]$
2: $\alpha = \max\{A_i\}$, $\beta = \max\{B_i\}$
3: $i_1 = \min\{i \mid A_i = \alpha\}$, $i_2 = \min\{i \mid B_i = \beta\}$
4: $I = \{i_1, i_2\}$
5: **if** $\min\{\alpha, \beta\} \geq 2$ or $\min\{|A|, |B|\} == 1$ **then** ▷ case(i)
6: **return** $I$
7: **else if** $\min\{|A|, |B|\} \geq 2$ **then**
8: **if** $\alpha = 1$ and $\beta \geq 2$ **then** ▷ case(ii)
9: $I = I \cup \min\{i \mid A_i = 1 \text{ and } i > i_1\}$
10: **else if** $\alpha \geq 2$ and $\beta = 1$ **then** ▷ case(iii)
11: $I = I \cup \min\{i \mid B_i = 1 \text{ and } i > i_2\}$
12: **else if** $\alpha = 1$ and $\beta = 1$ **then** ▷ case(iv)
13: $I = I \cup \{ \min\{i \mid A_i = 1 \text{ and } i > i_1\}, \min\{i \mid B_i = 1 \text{ and } i > i_2\} \}$
14: **return** $I$

Otherwise, we have $\min\{\alpha, \beta\} = 1$ and $\min\{|A|, |B|\} \geq 2$. Thus we make a case distinction depending on the value of $\alpha$ and $\beta$ (starting line 7), where we have to add an index $i > i_1$ for which $A_i = 1$ if $\alpha = 1$ and analogous if $\beta = 1$. The latter guarantees the center is contained in the singular locus of $\underline{x}^A - \rho\underline{x}^B$. This covers the missing cases (ii)–(iv) of Construction 7.4.1.

As in Algorithm 11, the case $\min\{\alpha, \beta\} = 0$ cannot appear since we tested whether the data of the chart is of the form (7.0.1) (line 3 Algorithm 7 and line 11 Algorithm 9).

If we consider the binomial of Example 7.3.1 and apply Algorithm 12, we obtain the same center as using Algorithm 11. But it may happen that the two algorithms lead to different centers.

**Example 7.4.4.** Let $f = x_1x_2 + x_3^2 \in K[x_1, x_2, x_3]$. Algorithm 12 provides the center $V(x_1, x_2, x_3)$, while Algorithm 11 gives the center $V(x_1, x_3)$.

## 7.5. Centers of minimal codimension contained in an exceptional divisor or contained in the singular locus

Now, we present the fourth variant, where we slightly relax the restriction on the centers that we imposed in the previous section. We obtain this by taking the preceding resolution process into account.

Let $\pi\colon B\ell_D(\mathbb{A}^n_K) \to \mathbb{A}^n_K$ be the blowup with center $D = V(x_i \mid i \in I)$ for $I \subseteq \{1,\dots,n\}$. In the $X_i$-chart $U_i = D_+(X_i) \cong \mathbb{A}^n_K$ $(i \in I)$, we have the coordinates $(\underline{x}') = (\underline{x}'_1,\dots,\underline{x}'_n)$ and $\langle x_j \mid j \in I\rangle \cdot K[\underline{x}'] = \langle x'_i\rangle \subset K[\underline{x}']$. Hence, the preimage of the center along the blowup $\pi$ coincides with the divisor $E := \operatorname{div}(x_i)$ in $U_i$. Since a blowup is an isomorphism outside of its center, we have the freedom to choose any center $D' \subset U_i$ contained in $E$ without losing the condition that the composition of $\pi$ and the blowup in $D'$ is an isomorphism outside of $D$. In particular, we may choose centers of codimension two as in Construction 7.3.4.

Clearly, the previous observation extends to any finite sequence of (local) blowups of the above type. This motivates the following method for choosing the center.

**Construction 7.5.1.** Let $f = \underline{x}^C(\underline{x}^A - \rho\underline{x}^B) \in K[\underline{x}]$ with $\rho \in K^\times$ and $A, B, C \in \mathbb{Z}^n_{\geq 0}$ such that $A_iB_i = 0$ for all $i \in \{1,\dots,n\}$. Let $(y_1,\dots,y_m)$, $m \leq n$, be a subsystem of distinguished variables of $(x_1,\dots,x_n)$ such that the exceptional divisor of the local monomialization procedure is given by $\operatorname{div}(y_1\cdots y_m)$. If there is a center $D_I = V(x_{j_1}, x_{j_2})$ as in Construction 7.3.4, which is also contained in the exceptional locus, $D_I \subset \operatorname{div}(y_1\cdots y_m)$, then choose $D_I$ as the center for the next blowup. Otherwise, we follow Construction 7.4.1.

Note that $D_I \subset \operatorname{div}(y_1\cdots y_m)$ is equivalent to the condition that $x_{j_1} = y_{k_1}$ or $x_{j_2} = y_{k_2}$, for some $k_1, k_2 \in \{1,\dots,m\}$.

As a consequence of Propositions 7.3.5 and 7.4.2, we get the termination of the local monomialization procedure using centers given by Construction 7.5.1 (since $\iota(.)$ decreases strictly after the blowup).

**Corollary 7.5.2.** *The local monomialization process obtained by choosing the centers as in Construction 7.5.1 terminates for every binomial $f \in K[\underline{x}]$.*

In Algorithm 13 we present an implementation of Construction 7.5.1. This is analogous to Algorithm 12 with the only difference in line 5, where we have the additional condition $E_{i_1} + E_{i_2} > 0$. The latter holds whenever $x_{i_1}$ or $x_{i_2}$ correspond to an exceptional divisor of a previous blowup.

**Algorithm 13** compute center with minimal codimension contained in an exceptional divisor or contained in the singular locus

**INPUT:** list $M$, mode $= 4$, where $M$ is of the same form as $L[i]$ in Remark 7.1.2
**OUTPUT:** $I \subseteq \{1, \ldots, n\}$ such that $V(x_i \,|\, i \in I)$ is the next center in the monomialization process

```
(A, B, C, E) = M[1]
α = max{A_i}, β = max{B_i}
i_1 = min{i | A_i = α}, i_2 = min{i | B_i = β}
I = {i_1, i_2}
if min{α, β} ≥ 2 or min{|A|, |B|} == 1 or E_{i_1} + E_{i_2} > 0 then
    return I                                          ▷ exceptional or case(i)
else
    I = compute_center(M, 3)
    return I                                          ▷ case(ii),(iii),(iv)
```

## 7.6. A glimpse into the case of more than one binomial and non-invertible coefficients

Let us briefly outline a method to extend our procedures to finitely many binomials and to a single binomial with coefficients in $\mathbb{Z}_p$.

**Construction 7.6.1.** Let $f_1, \ldots, f_m \in K[\underline{x}] = K[x_1, \ldots, n]$ be finitely many binomials, where $K$ is a field. In order to monomialize them, we may successively apply one of the discussed procedures to $f_1$, then the total transform of $f_2$ and so on. Since each of these methods terminates, we reach the case that each $f_i$ is locally monomial.

At the end of the last construction, the total transform of the binomials $f_1, \ldots, f_m$ are not necessarily simultaneously locally monomial, or in other words, the product $f_1 \cdots f_m$ is not necessarily locally monomial, as the Example 7.6.2 illustrates.

**Example 7.6.2.** Let $K$ be a field and consider

$$f_1 := x_1 - 1, \quad f_2 := x_2 - 1, \quad f_3 := x_1 x_2 - 1.$$

We claim that the product $f_1 f_2 f_3$ is not locally monomial. To see this, we introduce $y_1 := x_1 - 1$ and $y_2 := x_2 - 1$. Then $f_1 f_2 f_3 = y_1 y_2 (y_1 y_2 + y_1 + y_2)$.

Eventually, the task to make $f_1, \ldots, f_m$ simultaneously locally monomial can be reduced to the problem of (locally) monomializing an element of the form

$$\prod_{i=1}^{m} (\underline{x}^{A(i)} - \lambda_i), \qquad \lambda_i \in K^\times \text{ and } A(i) \in \mathbb{Z}^n_{\geq 0} \text{ for } 1 \leq i \leq m.$$

This problem is connected to the desingularization of arrangements of smooth subvarieties, which is treated over algebraically closed fields in [60] or [64], for example. Nonetheless, for finitely many prime characteristics $p = \mathrm{char}(K) > 0$ (depending on the exponents $A(i)$) the situation becomes more involved and further investigations are required. Since the present article focuses on the case of a single binomial, we do not go into the details here.

When successively applying a monomialization method to $(f_1, f_2, \ldots, f_m)$, the order in which we handle the elements has an impact on the final numbers of charts.

**Example 7.6.3.** Let $K$ be any field. Consider the binomials

$$f_1 = v^2 - y^4 z, \quad f_2 = x^2 y - z^3 \quad \in K[x, y, z, v].$$

If we use our local monomialization method with codimension two centers for $(f_1, f_2)$ (i.e., first for $f_1$ and then for the total transform of $f_2$), then the procedure needs computation in 43 charts and 19 of these charts are final charts.

On the other hand, if we take the order $(f_2, f_1)$, then the same procedure needs only 31 charts and 12 of them are final charts.

We now turn our attention to the situation over $\mathbb{Z}_p$ instead of over a field.

**Remark 7.6.4.** We observe that our proofs for the termination of the local monomialization procedures rely on a study of the exponents. This suggests that the methods may also be used for a first step towards a monomialization if we are not necessarily restricted to the situation over field $K$. Consider a binomial in $\mathbb{Z}_p[\underline{x}] = \mathbb{Z}_p[x_1, \ldots, x_n]$, say

$$f = p^e \underline{x}^C ( \underline{x}^A - \lambda p^d \underline{x}^B ), \qquad \lambda \in \mathbb{Z}_p^\times,$$

for $d, e \in \mathbb{Z}_{\geq 0}$ and $A, B, C \in \mathbb{Z}_{\geq 0}^n$ with $A_i B_i = 0$ for all $i \in \{1, \ldots, n\}$. We fix one of the procedures, which we discussed, and apply it to $f$, considered as a binomial with coefficients in the field $\mathbb{Q}_p = \mathrm{Quot}(\mathbb{Z}_p)$. As we have seen, this terminates after finitely many blowups. Since the coefficients are in $\mathbb{Z}_p$, the resulting total transform of $f$ is not necessarily locally monomial. For example, it may appear that the total transform of $f$ is of the form

$$p^e \underline{x}'^{C'} ( \underline{x}'^{A'} - \lambda p^d ),$$

for some $A' \in \mathbb{Z}_{\geq 0}^n$ with $|A'| \geq 2$. Here, $d, e \in \mathbb{Z}_{\geq 0}$ are the same integers as at the beginning since we did not touch coefficients. If $d \neq 0$, then the monomialization

process is not finished, yet.

There are at least two directions that one could follow:

1. We blow-up centers of the form $\langle x'_i, p\rangle$, where $i$ is chosen appropriately. This has the drawback that the ambient ring after the blowup is not necessarily isomorphic to a polynomial ring over $\mathbb{Z}_p$. More precisely, in the $X'_i$-chart, we get $\mathbb{Z}_p[\underline{x}', v]/\langle p - x'_i v\rangle$.

2. An alternative method is to make a case distinction depending on the residues of $x'_i$ modulo $p$. If $x'_i \equiv 0 \mod p$, we can write it as $x'_i = p\, y_i$ for some new variables taking values in $\mathbb{Z}_p$. By choosing $i$ appropriately, we can make $d$ decrease. On the other hand, if $x'_i \not\equiv 0 \mod p$ for all $i$, then $\underline{x}'^{A'} - \lambda p^d$ is a unit. Thus $f$ is monomial.

The analysis of the complexity and a comparison of the variants (Section B.3) can be found in the Appendix.

# 8. Algorithmic aspects of the determinantal case

In this chapter, we present a new algorithm for resolution of singularities defined by minors of matrices with entries for which a principalization construction exists, e.g, binomial entries. We focus on centers which are globally chosen but not required to be contained in the singular locus of the variety, but which take advantage of the determinantal binomial structure. Furthermore, we use some black boxes for later improvements.

Throughout the chapter, we fix $K$ to be an algebraic closed field and we use multiindex notation: We write $\underline{x}^A = x_1^{A_1} \cdots x_n^{A_n}$, for $A = (A_1, \ldots, A_n) \in \mathbb{Z}^n_{\geq 0}$. Furthermore let $f = \underline{x}^A - \lambda \underline{x}^B \in K[\underline{x}] := K[x_1, \ldots, x_n]$ be a binomial, where $\lambda \in K^\times$.

First methods of resolution of so called determinantial singularities can be found in [75]. In [71], Schober described the resolution of determinantial singularities of generic matrices. In this section, we want to generalize and implement our generalization of the strategy of [71], which we discussed in Section 6.1.

Furthermore, we use for our approach the results in resolution on binomial ideals, namely, the algorithm of Blanco and Encinas.

The following example shows that it is not obvious which singularities are determinantal since some terms are vanishing when we calculate the determinant.

**Example 8.0.1.** Let $f = x^2z^2 + y^3wv + v^2yz + x^2wv \in \mathbb{F}_2[x, y, z, v, w]$. Then $f$ is a determinantal singularity generated by a matrix with binomial entries, since

$$\begin{aligned} f = \det \begin{pmatrix} x^2 - y^3 & yz \\ v^2 - y^2z & z^2 - wv \end{pmatrix} &= x^2z^2 + y^3wv + x^2wv + y^3z^2 + v^2yz + y^3z^2 \\ &= x^2z^2 + y^3wv + x^2wv + v^2yz. \end{aligned}$$

## 8.1. Determinantally monomial case

We want to generalize the generic approach and let our matrices be matrices with monomial entries. So we consider a matrix $M = (m_{i,j})_{1\leq i\leq r, 1\leq j\leq k}$, with $m_{i,j} = \underline{x}^{A_{i,j}}$ being some arbitrary monomials in $K[\underline{x}]$.

Our goal is to mimic and adapt the algorithm described above in Section 6.1. The ideal of 1-minors is a monomial ideal and the ideal generated by the 2-minors is a binomial ideal. In the worst case, the generators of the ideal generated by the 3-minors have six terms, so the number of terms does not increase linearly. And in arbitrary characteristic, it is an open question if there is a finite procedure to resolve a singularity of dimension $> 3$.

Without loss of generality and for better readability, we restrict $M$ to be a quadratic matrix. In general this is no restriction.

We treat binomial resolution as a black box in this section. We fix the algorithm of [9] and [10] but it is also possible to use one of the algorithms we have mentioned in Construction 6.1.1.

We use the notation of idealistic exponents (see [71] or Section 2.3.3 for more details).

With the necessary theoretical background (Section 2.3.3), we can principalize the monomial ideal $\langle \underline{x}^{A_{i,j}} \mid 1 \leq i,j \leq m\rangle$ generated by the entries of the matrix $M = (\underline{x}^{A_{i,j}})_{1\leq i,j\leq m}$.
The subgoal is to blow-up in the ideal generated by the 1-minors of $M$ as before in the generic setting. But this ideal could be an arbitrary monomial ideal which is potentielly very singular. That is why we need a canonical global procedure to get this reduction of dimension, like in the generic case.

We can factor out the greatest common divisor of the set of all entries of the matrix. We set

$$d := \min\{|A_{i,j}| \mid i,j\} = \operatorname{ord}_{\underline{x}}\langle \underline{x}^{A_{i,j}}\rangle.$$

Without loss of generality, we can assume that the ideal generated by the entries in row $j$ has order $d$, for all $1 \leq j \leq m$, since according to Lemma 2.3.34, we can multiply with an $\underline{x}^{d_{i,j}}$, such that $d$ equals the least common multiple.

Then we consider the idealistic exponent $\bigcap_{i,j}(\underline{x}^{A_{i,j}}, d)$, i.e., we consider every entry in our matrix but we only want to factorize out a term of order $d$. For better

readability, we write

$$\bigcap_{\ell=0}^{N}(\underline{x}^{A_{i,j}},d)$$

for $\ell = 1,\dots,i\cdot j$ and count the variables line by line, i.e., $x_{i,j} = x_{(i-1)\cdot m+j}$.
We define new variables $y_{2\ell}$ and $y_{2\ell+1}$ for $\ell = 0,\dots,N$ and use the canonical embedding to $K[\underline{x},\underline{y}]$. We consider the idealistic exponent

$$(y_{2\ell}^{d-1}y_{2\ell+1}+\underline{x}^{A_\ell},d) \sim (y_{2\ell},1)\cap(y_{2\ell+1},1)\cap(\underline{x}^{A_\ell},d).$$

We see by considering the partial derivative $\frac{\partial}{\partial y_{2\ell+1}}$ that these idealistic exponents are equivalent. This yields us $(y_{2\ell}^{d-1}y_{2\ell+1},d)\cap(y_{2\ell}^{d-1},d-1)$ which is equivalent to $(y_{2\ell},1)$.

The goal is to consider each of these idealistic exponents similarly, so we have to do the same calculations with $\bigcup_{\ell=0}^{N}(y_{2\ell}^{d-1}y_{2\ell+1}+\underline{x}^{A_\ell},d)$. Then $\langle y_{2\ell}^{d-1}y_{2\ell+1}+\underline{x}^{A_\ell} \mid \ell = 0,\dots,N\rangle$ yields a standard basis. We can resolve this ideal by any algorithm which resolves binomial ideals by resolving the idealistic exponent. Since we only want to resolve $\bigcup_{\ell=0}^{N}(\underline{x}^{A_\ell},d)$ and $d$ is the maximal order, we obtain after a finitely number of blow-ups a new value $d_{\text{new}} < d$. The value of $d$ can only decrease finitely many times. And this yields a resolution of the ideal $\langle y_{2\ell}^{d-1}y_{2\ell+1}+\underline{x}^{A_\ell} \mid \ell = 0,\dots,N\rangle$.

This algorithm has centers of the form $V(\underline{y},\underline{x}_I)$. We get a resolution of the ideal $\langle \underline{x}^{A_\ell} \mid \ell = 0,\dots,N\rangle$ by using only the $X_I$-charts, namely by blowing up in centers $V(\underline{x}_I)$.

**Remark 8.1.1.** We need two types of variables $y_{2\ell}$ and $y_{2\ell+1}$ for the case that $d$ divides the characteristic $\text{char}(K)$ of the underlying field $K$. If we only add $y_\ell^d$ instead of $y_{2\ell}^{d-1}y_{2\ell+1}$ then applying the partial derivative $\frac{\partial}{\partial y_\ell}$ yields $(dy_\ell^{d-1},d-1)$ which vanishes in characteristic $\text{char}(K)$.

**Remark 8.1.2.** In characteristic 0, the $d$ cannot divide the characteristic like in Remark 8.1.1. So we only need to add $y_\ell^d$ for each monomial. We use this fact in our implementation described in Section A.5.2 in order to reduce complexity.

Altogether we have the following principalization of a monomial ideal, with the black box of a binomial resolution procedure.

**Construction 8.1.3.** Let $\mathcal{A}$ be an algorithm, which resolves binomial ideals globally and canonically, e.g., the algorithm of [9]. Let $M$ be a monomial matrix in $K[\underline{x}]$ and let $J$ be the ideal generated by the entries of $M$.

1. Set $d = \text{max-ord}(J)$.

2. If $d = 1$, then the procedure is finished.

3. Set $\mathcal{J} := \langle y_{2\ell}^{d-1} y_{2\ell+1} + \underline{x}^{A_\ell} \mid 1 \leq \ell \leq N \rangle$.

4. Apply $\mathcal{A}$ to $\mathcal{J}$ and substitute each center of the blowup $\mathcal{C}$ for $\mathcal{J}$ by deleting the $V(\underline{y})$-components (name it $C$) and calculate the blowup with center $C$ for $J$.

5. Go to (1).

We have a binomial ideal in each affine chart which can be embedded in some $\mathbb{A}^n \times \mathbb{P}^{M_1} \times \cdots \times \mathbb{P}^{M_\ell}$, where $M_i$, $i = 1, \ldots, \ell$, is the number of new variables that first appear in the $i$-th blow-up. The algorithm described in [9] and [10] chooses a global center where we have to blow-up our binomial ideal. Thus our algorithm inherits the property of being global and it ensures the glueing of our charts. Therefore, this resolution step is a global one.

If we have a 1 entry in our matrix, we can use the gaussian step. Without loss of generality, we can assume that we have locally $m_{1,1} = 1$ than, we have the following situation:

$$M' = \begin{pmatrix} m_{2,2} - m_{2,1}m_{1,2} & \ldots & m_{2,r} - m_{2,1}m_{1,r} \\ \vdots & \ddots & \vdots \\ m_{k,2} - m_{k,1}m_{1,2} & \ldots & m_{k,r} - m_{k,1}m_{1,r} \end{pmatrix},$$

where $m_{i,j}$ are monomials. Since we do not have several variables $x_{i,j}$ but some arbitrary monomials here, it is not guaranteed that our entries have enough normal crossings, i.e., that we find a linear transformation such that every entry of our matrix is a monomial.
That is why we cannot handle the determinantally monomial case without the determinantally binomial case, which we discuss in the next section.

## 8.2. Main algorithm for resolution of determinantal singularities of at most binomial type

We now turn to the main task of this chapter. We wish to resolve a determinantal singularity of at most binomial type ,i.e., for which there exists a generating matrix with at most binomial entries. We want to take advantage of the approach in the resolution of binomial ideals and the exploitation of the matrix structure as described above.

From now on, we will distinguish the following cases:

1. $K$ has characteristic 0 or

2. $K$ has positive characteristic.

In characteristic 0, we have at least two possibilities to desingularize the ideal of maximal minors by an algorithm that provides a Hironaka-style resolution, so the most obvious way to resolve this singularity is

(a) calculate the ideal generated by the minors,

(b) resolve it by a Hironaka-style resolution.

The complexity of a Hironaka-style resolution is bounded by the $(\ell + 3)$-rd level of the Grzegorczyk hierarchy, where $\ell$ denotes the dimension of the scheme which we want to resolve. See [5] for more information about the complexity analysis and [47] or Section B for more information on the Grzegorczyk hierarchy.
So our approach can be useful for practical reasons also in characteristic 0 since it could provide an algorithm with a smaller computational complexity and it could extend the set of resolution of singularities that can be calculated by a computer.

The second approach we describe below. In characteristic $p > 0$, the question, if there exist such a procedure for dimension $> 3$, is still wide open. The considered determinantal ideals are generated by matrices with binomial entries, but the ideal itself is far from being a binomial ideal. A (constructive) resolution of determinantal singularities in positive characteristics is, however, not known due to their large dimensionality.

First, we state the important parts of the algorithm as black boxes. This has the advantage that we can replace our explicit modular formulation of Algorithms 1, 2 and 3 with another (possibly improved) algorithm in the future. Furthermore, it is obvious that we are not restricted to binomial entries here, but to entries for which a constructive principalization procedure exists. The different parts do not depend on the explicit algorithms but we give examples here.
We divide our computation (illustrated in Figure 8.1) into the following four parts.

1. Principalization of a list of, e.g., binomial entries, for example, see section 8.2.1.

2. Establishing normal crossings and a covering to obtain monomial entries in each chart, see for example Section 8.2.2.

3. Determinantally monomial case, see Section 8.1.

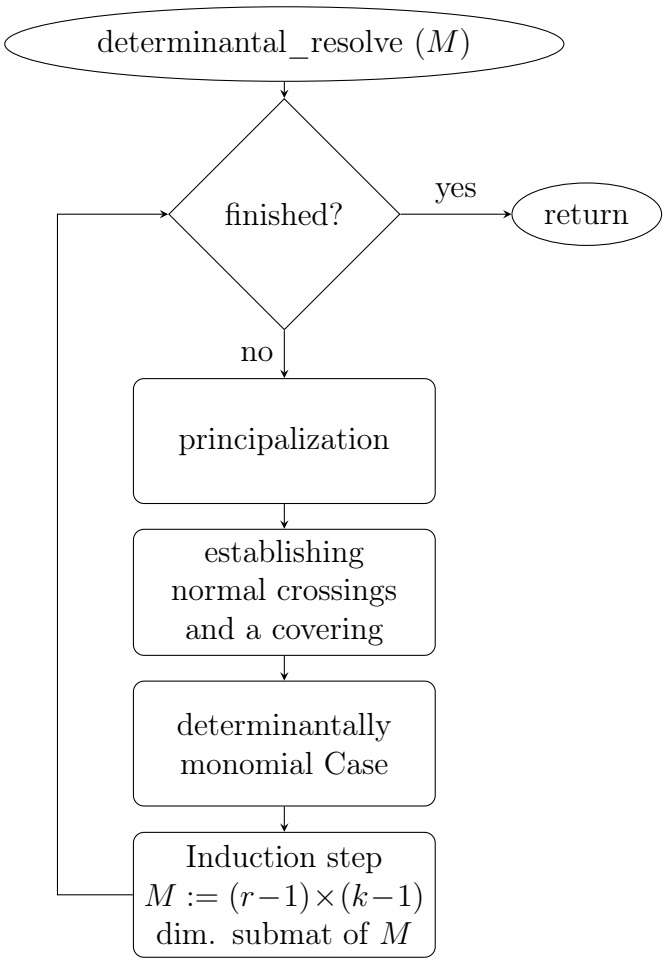



Figure 8.1.: Flow chart of the main algorithm for determinantal resolution

4. Induction step to a matrix with $(r-1) \times (k-1)$ binomial entries, see Remark 6.1.1 and Remark 6.1.2

**Remark 8.2.1.** For $M = (f_{i,j})$ we can relax the condition of $f_{i,j}$ being binomials. We only have to assume that a constructive principalization of the $f_{i,j}$ is possible. If we know such a global principalization algorithm $A$ for the $f_{i,j}$ then we can replace the algorithm in section 8.2.1 by $A$. The main algorithm leads us to a weak determinantal resolution of the singularity with these small changes.

**Construction 8.2.2.** Mainly we get the construction described in Figure 8.1.
As said before, the main algorithm consists of some black boxes we have to encode: First, we want to principalize the entries of our matrix $M$ and finally, we need to make them normal crossing to do some variable transformation to have only monomial entries in the strict transform $M'$ of $M$.

### 8.2.1. Principalization of the entries in positive characteristic

The goal is to obtain entries that are principalized.

We will not focus on choosing centers in the singular locus of our determinantal singularity but rather on the binomial structure in our entries.

**Remark 8.2.3.** An effective simultaneous resolution of binomials with toric Artin stacks, for which only a single weighted blow-up is needed, we refer to [3]. The computational effort is shifted to the destackification of the Artin stack, here. It is possible to use this approach instead of the variant which we present here.

We focus on the transformation of the individual entries as polynomials.

To reach principalized entries, which does not need to have normal crossings (for more information see Section 8.2.2), in a covering of our variety, we use the following construction.

**Construction 8.2.4.**

1. We fix the entries of our matrix and store the binomial entries which are not principalized in an ordered list $(f_1, \ldots, f_m)$.
2. We concentrate on the first binomial $f_1$ and monomialize the ideal $\langle f_1 \rangle$ using a resolution procedure for a binomial ideal (e.g., see [8]).
3. After this sequence of blow-ups we have a principalized weak transform of $f_1$. So we can delete this item from our list.
4. We repeat this process until our list is empty.

**Remark 8.2.5.**

- The second step of Construction 8.2.4 uses the fact that every monomial generating set of a monomial ideal is a standard basis.
- Construction 8.2.4 terminates after finitely many steps since our list contains finitely many binomials and the algorithm for each binomial terminates after finitely many steps.
- The choices of the centers of the blowing ups are global since the order is fixed, and for every resolution of each hypersurface we are using the same algorithm with canonical choice of center. For binomial entries, see [9] and [10].
- The ordering in the list influences the number of charts and the choice of the canonical centers.

- To obtain a global algorithm, we need to store and edit the tuple of tuples of 1-, 2-, ... resp. $\min\{r,k\}$-minors of the initial matrix. During the whole process we have to edit each such tuple. Thus we fix an ordering, then choose an ordered tuple of 1-minors, then the tuple of 2-minors etc. until we reach $\min\{r,k\}$. This yields a fixed order on the minors and in the lists during the whole process. For this reason, the centers during this process are canonical. The Construction 8.2.4 then has to deal with the unique list which has principalized entries. Note, that if the $i$-minors are principalized, the $(i+1)$-minors are locally binomial and the $j$-minors for $j < i$ contain a unit.

Note, that we cannot guarantee that the computed centers are in the singular locus of the considered determinantal singularity. Therefore, this step is the only step of the algorithm which ensures that our resolution is a weak resolution of singularities.

**Example 8.2.6.** Let

$$M = \begin{pmatrix} x^2 - y^3 & z \\ v^2 - w^3 & xy \end{pmatrix} \in (K[x,y,z,v,w])^{2\times 2}.$$

Then our algorithm for principalization of the entries may choose $V(x,y)$ as the next center but this is not contained in the singular locus of the determinant of $M$.

**Remark 8.2.7.** This part of the main algorithm is the only part where we can not guarantee that our centers are contained in the singular locus of the determinantal variety. Former ideas were to

- principalize the ideal of the 1-minors of the matrix,
- principalize the ideals generated by single rows of the matrix and
- principalize the ideals generated by single columns of the matrix.

We can guarantee in these approaches that the centers are contained in the singular locus of the determinantal variety, but let $I$ be a binomial ideal then the equality $I = \langle 1 \rangle$ does not imply that every entry in the matrix become principalized.
Let

$$M = \begin{pmatrix} 1 & z^3 - x^2 & 0 \\ v^2 - w^3 & 1 & x \\ v^2 - w^3 & z^3 - x^2 & 1 \end{pmatrix} \in (K[x,y,z,v,w])^{3\times 3}.$$

Then every of the considered ideals equal $\langle 1 \rangle$ but a subsequent use of the gaussian step would destroy the binomial structure of the algorithm.

**Remark 8.2.8.** Some analysis of the complexity, namely the required number of charts for some examples and worst-case number of charts, for local monomialization of a single binomial is considered in [42, Section 8].

Our goal is to obtain binomials of the form $(1 - \lambda \underline{x}^A)$. More precisely, each entry of the matrix is of the form $\underline{x}^C(1 - \lambda \underline{x}^A)$, for some $\lambda \in K$. We want to avoid entries of the form $\underline{x}^C(y - \lambda \underline{x}^B)$ and blow-up some more time if it is necessary.

## 8.2.2. Establishing normal crossings and a covering

The fact that every entry of our matrix $M$ is principalized is not sufficient. We need normal crossings instead. The following example illustrates the differences.

**Example 8.2.9** ([42, Example 7.2], Example 7.6.2)**.** Let

$$M = \begin{pmatrix} 1 & xy-1 \\ x-1 & y-1 \end{pmatrix} \in (K[x,y])^{2\times 2}.$$

Each entry $f_1 = 1, f_2 = x-1, f_3 = xy-1, f_4 = y-1$ is principalized, but they have not normal crossings, i.e., the product $f_1 \cdots f_4$ is not principalized. If we introduce new variables $x' := x-1$ and $y' := y-1$, we get $f_1 f_2 f_3 f_4 = x'y'(x'y' + x' + y')$ which is not principalized.

The idea of this 'establishing normal crossing and a covering'-step is illustrated in Figure 8.2. The principalized entries of the matrix coincide with several hypersurfaces which intersects in a point. In a two-dimensional space, three different lines could not meet transversal in a point. So we have to blow-up to seperate the lines. We get two points where two lines have normal crossings. We can cover this chart by the complement of the green line (where we consider the left point) and the complement of the red line (where we consider the right point).

This problem is connected to the desingularization of arrangements of smooth subvarieties, which is discussed in [60] and Section 5.5. We want to use the Theorem of Hu (Theorem 5.5.2).

If the intersection lattice of the entries of the matrix is a simple arrangement, the process of forcing normal crossings is a combinatorial problem, which is solved by Hu [60].

**Remark 8.2.10.** In our setting, $X^0 := X \setminus Y$, where $Y := V(\prod_{\ell=1}^m f_\ell)$ and $f_\ell = 1 - \rho_\ell x^{A_\ell}$ are the principalized entries. Furthermore, let $I = \mathfrak{P}(\{1,\dots,m\})$ and for all $i \in I$ we define $D_i = \bigcap_{\ell \in i} V(f_\ell)$.

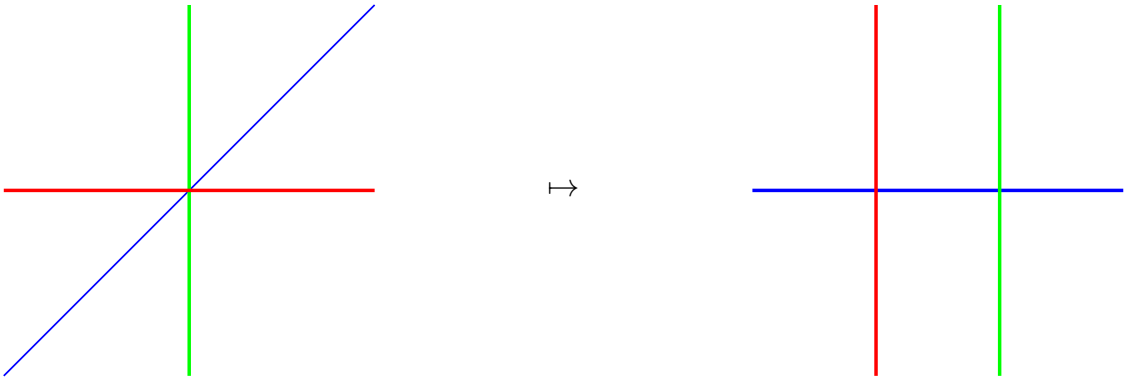


Figure 8.2.: Illustration of the 'establishing normal crossing and a covering' step

The rank of $D_i$ is equal to $m - |i|$.

Altogether we get:

**Lemma 8.2.11.** *If the intersection lattice of the entries of the matrix is a simple arrangement, the simultaneous local monomialization process is a combinatorial problem, which is solved by the sequence of blowups by [60] (Theorem 1.1) for all but finitely many characteristics.*
*Let* $f_\ell = 1 - \lambda_\ell \underline{x}^{A_\ell}$ *be the locally monomial entries. Furthermore, let* $I = \mathfrak{P}(\{1, \ldots, m\})$ *and for all* $i \in I$ *we define* $D_i = \bigcap_{\ell \in i} V(f_\ell)$.
*Then* $\{D_i\}_{i \in I}$ *is a simple arrangement, if the following holds:*

1. $D_i$ *is smooth.*
2. $D_i$ *and* $D_j$ *meet cleanly.*
3. $D_i \cap D_j = \emptyset$ *or a disjoint union of* $D_\ell$.

Not every such intersection lattice needs to be a simple arrangement. The following example should illustrate this.

**Example 8.2.12.** Consider $f = (x-1)(y-1)(x^p y-1) \in K[x, y]$ with $p = char(K) > 0$.
Let $w := x - 1$ and $z := y - 1$. Then $f = wz(z + w^p + w^p z)$.
Let $D_{\{1\}} = V(w)$, $D_{\{2\}} = V(z)$ and $D_{\{3\}} = V(z + w^p + w^p z)$.
Then $D_{\{2\}}$ and $D_{\{3\}}$ do not meet cleanly, since the scheme-theoretic intersetion $\langle z, w^p \rangle$ is non-reduced and it is not smooth.

**Remark 8.2.13.** Example 8.2.12 illustrates the reason for stating that our algorithm terminates and is correct for all but finitely many positive characteristics: In this special example, we have to exclude the characteristic $p > 0$.

**Lemma 8.2.14.** *The assignments in Remark 8.2.10 meet the assumptions in Theorem 5.5.2.*

*Proof.* 1. It is clear that $D_i = \bigcap_{\ell\in i} V(f_\ell) = V(f_\ell|\ell \in i) = V(1-\lambda_\ell x^{A_\ell}|\ell \in i)$, for each $i \in \{1,\dots,m\}$ is smooth.

(3) We have

$$D_i \cap D_j = \bigcap_{\ell_1\in i} V(f_{\ell_1}) \cap \bigcap_{\ell_2\in i} V(f_{\ell_2}) = \bigcap_{\ell\in i\cup j} V(f_\ell).$$

Since $i,j \subseteq \{1,\dots,n\}$, we know that $i\cup j \in \mathfrak{P}(\{1,\dots,n\})$. And therefore we can find an $i'$ with $i' = i\cup j$ and $D_{i'} = D_i \cap D_j$.

(2) $D_i$ and $D_j$ meet cleanly, since $D_i \cap D_j$ is smooth by (1) and (3) for $i \neq j \in \mathfrak{P}(\{1,\dots,n\})$ and for their tangent spaces

$$\begin{aligned} T(D_i \cap D_j) &= T(\bigcap_{\ell_1\in i} V(f_{\ell_1}) \cap \bigcap_{\ell_2\in i} V(f_{\ell_2})) \\ &= T(V(1-\lambda_\ell x^\ell \mid \ell \in i\cap j)) \\ &= T(V(1-\lambda_\ell x^\ell \mid \ell \in i)) \cap T(V(1-\lambda_\ell x^\ell \mid \ell \in j)) \\ &= T(\bigcap_{\ell\in i} V(f_\ell)) \cap T(\bigcap_{\ell\in j} V(f_\ell)) \\ &= T(D_i) \cap T(D_j) \end{aligned}$$

holds in all but finitely many characteristics. □

Obviously, there are only finitely many $D_i$ so the following algorithm ends after finitely many steps.

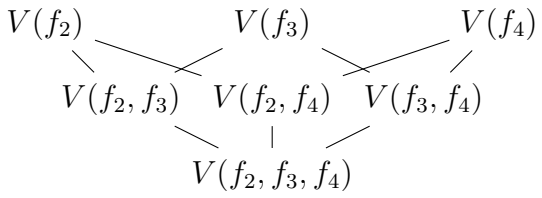
**Remark 8.2.15.** Since we are dealing with algebraic varieties in Theorem 5.5.2 our algorithm for forcing normal crossings is a global one.

**Example 8.2.16** (Continuation of Example 8.2.9)**.** Let $f_1 = 1, f_2 = x-1, f_3 =$

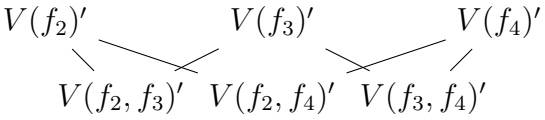
$xy-1$ and $f_4 = y-1$. Then $Y := V(f_2 \cdot f_3 \cdot f_4)$ and $I = \mathfrak{P}(\{2,3,4\})$ and we define

$$\begin{aligned} D_\emptyset &= \emptyset \\ D_{\{2\}} &= V(f_2) \\ D_{\{3\}} &= V(f_3) \\ D_{\{4\}} &= V(f_4) \\ D_{\{2,3\}} &= V(f_2) \cap V(f_3) \\ D_{\{2,4\}} &= V(f_2) \cap V(f_4) \\ D_{\{3,4\}} &= V(f_3) \cap V(f_4) \\ D_{\{2,3,4\}} &= V(f_2) \cap V(f_3) \cap V(f_4). \end{aligned}$$

We only need to consider the $D_i \neq \emptyset$, for $i \in \mathfrak{P}(\{2,3,4\})$. If we order the $D_i$ for $i \in \mathfrak{P}(\{2,3,4\})$ by their rank then we obtain the following diagram:




Hu's algorithm's choice for the center is the lowest level in the intersection lattice which is not empty, thus our center of the first blowing up is $V(f_1, f_2, f_3)$. After this first blowing up, we have the following situation:




Here $V(\cdot)'$ denotes the strict transform of $V(\cdot)$. For the next blowing up, the center of our choice is

$$C = V(f_2, f_3)' \cup V(f_2, f_4)' \cup V(f_3, f_4)'.$$

Finally, we have to consider

$$V(f_2)'' \qquad V(f_3)'' \qquad V(f_4)'',$$

where $V(\cdot)''$ denotes the strict transform of $V(\cdot)'$. The center of the last blowing up is

$$\tilde{C} := V(f_2)'' \cup V(f_3)'' \cup V(f_4)''.$$

Then $f_2''', f_3''', f_4'''$ have normal crossings.

Note, that in practice we do not need to blow-up everywhere in the intersection lattice. More information can be found in Section A.4.

### 8.2.3. Gaussian step

Now we are in the situation of Remark 6.1.2. Without loss of generality, since swapping rows and columns does not change the vanishing locus of the determinant, we can assume the following local situation

$$M' = \begin{pmatrix} 1 & g_{1,2} & \dots & g_{1,k} \\ g_{2,1} & g_{2,2} & \dots & g_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{r,1} & g_{r,2} & \dots & g_{r,k} \end{pmatrix},$$

where $\{g_{i,j} \mid 1 \leq i \leq r, 1 \leq j \leq k\}$ are principalized and we had applied the algorithm of Hu. We only have to make a change of coordinates and obtain some monomials $m_{i,j}$:

$$M' = \begin{pmatrix} 1 & m_{1,2} & \dots & m_{1,k} \\ m_{2,1} & m_{2,2} & \dots & m_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ m_{r,1} & m_{r,2} & \dots & m_{r,k} \end{pmatrix}$$

We can perform some elementary gaussian steps to eliminate the entries in row 1 and column 1 as before. This yields

$$M'_1 = \begin{pmatrix} m_{2,2} - m_{2,1}m_{1,2} & \dots & m_{2,r} - m_{2,1}m_{1,r} \\ \vdots & \ddots & \vdots \\ m_{k,2} - m_{k,1}m_{1,2} & \dots & m_{k,r} - m_{k,1}m_{1,r} \end{pmatrix}.$$

We have reduced our problem of resolving the ideal of the $\max\{r,k\}$-minors of $M$ to the problem of resolving the ideal of the $(\max\{r,k\}-1)$-minors of $M'_1$.

**Remark 8.2.17.** The gaussian step does not change the determinant and it is the reason why we have to deal with a tuple of tuples in the principalization step since it gives us the next tuple of the $(i+1)$-minors.
Therefore in total our presented algorithm is a global one.

Altogether, we have seen:

**Main Theorem 1.** *In arbitrary characteristic the algorithm in Construction 8.2.2*

*provides a resolution of arbitrary determinantal singularities of at most binomial type.*

Furthermore, we have seen that only our black boxes uses the specific structure of binomial entries. In general we can substitute them by other algorithm. This yield:

**Main Theorem 3.** *If there is a constructive procedure which principalizes and establishes normal crossings to the entries of a matrix that generates the determinantal singularity, then the algorithm in Construction 8.2.2 provides a resolution.*

Since in characteristic zero there is always a constructive Hironaka-style resolution, we obtain the following result:

**Main Theorem 2.** *In characteristic* 0*, the algorithm in Construction 8.2.18 provides a resolution of arbitrary determinantal singularities.*

If we perform the gaussian steps before monomializing the entries we get entries with a high number of terms, which are, in particular, not binomial anymore. In this case, we cannot use specialized algorithms, e.g., for resolving binomial ideals. This is a problem since it is well known that resolution of singularities in positive and mixed characterstic is still an open problem.

Another problem could be the complexity of the algorithms with respect to the high runtime.

### 8.2.4. Flexibility of the algorithmic setup

In this subsection, we discuss possible variants of our main algorithm.

As already mentioned, we can replace in particuar the first two steps of our algorithm (principalization and establishing normal crossings). This may provide a perspective of other cases beyond at most binomial entries. The usage of Hu's algorithm and the gaussian step do not depend on the initially binomial structure. Thus, if we know that there exists a constructive procedure of principalization of the entries of the matrix, we can replace step 1. For example, in characteristic zero we know by Hironaka's famous paper [54] that there exists a sequence of blowing ups such that the transform of every hypersurface is resolved. Furthermore, we know that resolution of singularities in characteristic 0 implies principalization of ideals. So in characteristic 0, we can use the algorithm of Hironaka as a subroutine which plays the role of the principalization and yield a resolution, here.

In characteristic $p > 0$, our presented algorithm yields a principalization of a determinant of a binomial matrix, so it is a principalization procedure as required in Section 8.2.1. Hence, we could use this algorithm recursively to resolve nested determinants, for instance of the form

$$\begin{pmatrix} \det\begin{pmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \end{pmatrix} & \det\begin{pmatrix} g_{1,1} & \det\begin{pmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{pmatrix} \\ g_{2,1} & g_{2,2} \end{pmatrix} \\ x_{2,1} & x_{2,2} \end{pmatrix},$$

where $f_{i,j}, g_{i,j}, h_{i,j}$ are binomials for $i, j = 1, 2$ and $x_{2,1}$ and $x_{2,2}$ are variables. Furthermore, we can replace the generic approach by the (skew-)symmetric generic approach of Section 6.2.

In the future, we can take a look at another new method that is described in the following construction:

**Construction 8.2.18.** Let $M = (m_{i,j})$ be an arbitrary $(k \times n)$ -matrix over a field $K$ of characteristic zero. The goal is to reduce the ideal of the $r$-minors of $M$ with $r \leq \min\{k, n\}$. We can take advantage of the determinantal structure and apply a Hironaka-style resolution in characteristic zero as follows:

1. Set $I := \langle m_{i,j} \mid 1 \leq i \leq k, 1 \leq j \leq n \rangle$.
2. Resolve $I$ by a Hironaka-style resolution.
3. If $r > 1$. Blow-up in the regular center which coincides with the ideal generated by the entrys of the weak transform of $M$.
   Else: stop.
4. In every chart we get a 1 entry in the weak transform of $M$. Apply Gauß.
5. Set $r := r - 1$. If $r \geq 1$, go to 1.

This should lead to a strong resolution of arbitrary determinantal singularities in characteristic zero.

In characteristic $p > 0$, we have discussed in Remark 8.2.7 that we can always principalize the ideals generated by the rows and colums which lead to a matrix (maybe we have to generate coverings here) with a one entry in each column and each row. All of the considered centers are in the singular locus and one only has to show that the singularity is already resolved after this resolution of the ideals

generated by the columns and rows. In the case that we have a $(m \times m)$-matrix of which we can assume without loss of generality that the one-entrys are on the diagonal and than we can apply Gauß to see this.

If these proof ideas lead to correct proofs, we can substitute the prinzipalization step of our matrix (which is the only weak step in our resolution) and we obtain a strong resolution in all of the presented cases.

# 9. Summary and Future Work

We have seen several well-known strategies for the resolution of singularities in Section 3 and Section 5. We can summarize all of these strategies as 'blow-up in centers contained in the most singular locus and repeat,' and the calculation of these centers can be quite different. Furthermore, we have seen in Sections 6 and 8 that determinantal singularities possess a quite good matrix structure which helps us to resolve the resolution in the generic case.
We have adapted this approach to the generic (skew-)symmetric case and saw that this case is more efficient concerning the exact number of charts and the running time.
Afterwards, we have put everything together to generalize the resolution strategy of generic determinantal singularities to a resolution of

1. arbitrary determinantal singularities in characteristic 0,

2. determinantal singularities of at most binomial type in positive characteristic,

3. determinantal singularities for which there exists a matrix with entries for which there exists a constructive principalization procedure which force normal crossings, in positive characteristic and

4. determinantal singularities for which a matrix exists with entries of recursive determinants of the form of 3 in positive characteristics.

We have seen that our new algorithm provides a strong resolution for (skew)-symmetric generic determinantal singularities but only a weak resolution for arbitrary determinantal singularities in characteristic $p \geq 0$ of the types in the enumeration above. This is because our principalization procedure uses centers that do not have to be contained in the singular locus of the determinantal variety. This gap has to be closed to achieve a strong resolution.

Moreover, our new algorithm is very modular and flexible, so we can use the algorithm as an algorithm of black boxes and use every improved algorithm in the

future instead of the presented ones. Under the stated assumptions, the correctness and termination will not be touched.

The author will implement the algorithms in the new computer algebra system Oscar because of the more developed data structures. Although the computer algebra systems Oscar and Singular differ, the pseudo code in the Appendix should be helpful since the implementation of the main algorithm is the first modular implementation of resolution of singularities, such that we can exchange the choice of the center by replacing single function calls. Moreover, this can be a starting point for a general implementation of resolution of singularities, of which the input consists of the singularity we want to resolve and the underlying algorithm.

From the complexity-theoretic point of view, it remains to do a worst-case complexity analysis of the algorithm of Blanco to be able to compare the newly presented algorithm with the one of Blanco or a general Hironaka-style resolution. In the ((skew-)symmetric) generic case, we have seen that a more specialized algorithm can yield an exact number of chart analysis and a much better complexity than a general resolution algorithm. For more information, see B.4.

The bottleneck in practice remains the high worst-case complexity and the high number of charts, so it might be helpful to parallelize the implementation such that each chart of a generation can be calculated simultaneously. Since the implementation is written in Singular, the tool of choice could be GPI-Space [68]. We have seen in [14] that this parallelization with GPI-Space often produces results in mainly lower runtime.

Furthermore, we have seen different approaches of local monomializations of single binomials. We can generalize it for binomial ideals for a better use in the calculation of p-adic integrals.

The following example illustrates the situation:

**Example 9.0.1.** The goal is to enumerate unitary subrings of $\mathbb{Z}^n$. We can do this with zeta functions

$$\zeta_{\mathbb{Z}^n}^{1,<}(s) = \sum_{1 \in H \leq \mathbb{Z}^n} |\mathbb{Z}^n : H|^{-s} = \prod_{p \text{ prime}} \zeta_{\mathbb{Z}^n,p}^{1,<}(s).$$

So the goal is reachable when determining the Euler factors $\zeta_{\mathbb{Z}^n,p}^{1,<}(s)$ for $p$ being a prime.

By applying a result of du Sautoy and Grunewald [31], the equation

$$\zeta^{1,<}_{\mathbb{Z}^n,p}(s) = (1-p^{-1})^{-n} \int_S |x_{1,1}|^{s-n+1} \cdots |x_{n-1,n-1}|^{s-1} d\mu$$

holds. This is a $p$-adic integral over the region $S$ which is given by inequalities:

$$\begin{aligned}
v_p(x_{1,1}) &\leq v_p(x_{2,1}(x_{2,1}-x_{2,2})) \\
&\vdots \\
v_p(x_{1,1}x_{2,2}) &\leq v_p(x_{2,1}x_{3,2}(x_{3,2}-x_{3,3}) - x_{2,2}x_{3,1}(x_{3,1}-x_{3,3})) \\
&\vdots \\
v_p(x_{1,1}x_{2,2}x_{3,3}) &\leq v_p(x_{2,1}x_{3,2}x_{4,3}(x_{4,3}-x_{4,4}) - \cdots \text{further terms}
\end{aligned}$$

$v_p$ denotes the $p$-adic valuation and the $x_{i,j}$ could have all $p$-adic values.
Let $n = 3$. Then

$$\int_{v_p(x_{1,1}) \leq v_p(x_{2,1}(x_{2,1}-x_{2,2}))} |x_{1,1}|^{s-2} \cdot |x_{2,2}|^{s-1} d\mu.$$

Since $|x_{1,1}|^{s-2} \cdot |x_{2,2}|^{s-1}$ is monomial, we only need to consider the region $S$. We have to distinguish different cases:

1. $v_p(x_{1,1}) = 0$. In that case, the inequality does always hold.
2. $v_p(x_{1,1}) > 0$ and $v_p(x_{2,2}) = 0$, the situation is always monomial.
3. $v_p(x_{i,j}) > 0$, for all $i, j$, the situation is more complicated and we need monomialization.

Figure 9.1 illustrates the situation of case 3 as a real picture. The yellow plane illustrates $V(x_{1,1})$, the green and blue one are the $V(x_{2,1})$ and $V(x_{2,2})$ and the red diagonal is $V(x_{2,1} - x_{2,2})$. We see commonalities to the process of establishing normal crossings, we discussed in Section 2.3.1 and Section 8.2.2. So we have to blow-up in $V(x_{2,1}, x_{2,2})$.

Figure 9.2 illustrates the situation after this blow-up. We obtain an exceptional plane which is the grey one in Figure 9.2 and the other planes have normal crossings. This is more or less the same procedure and an analogue picture we have seen in Figure 8.2 in the two dimensional case.
In these two charts, we have:

$X_{2,1}$-chart: $E = V(x_{2,1})$ and all three cases are monomial.

Figure 9.1.: Real picture illustration of the most complicated case for the $p$-adic integration

Figure 9.2.: Real picture illustration of situation after the blow-up

$X_{2,2}$-chart: $E = V(x_{2,2})$ and the remaining case is monomial.

So the integration becomes more easier after this monomialization procedure.

Furthermore, we can proof that the ideal in Example 9.0.1 is a determinantal one, we also can use our main algorithm or we could compare both approaches in this setting.

# A. Implementational Aspects

In this chapter, we will discuss the implementational aspects of this thesis.

The constructive parts of this thesis are implemented in the computer algebra system Singular in version 4.1.1. [30]. This relies on the author deciding which computer algebra system to use at the beginning of the corona pandemic in 2020 and the lower risk to implement in the well-known Singular instead of a newly developing computer algebra system Oscar [29]. In principle, it is sense full to immigrate or reimplement these parts in Oscar, too.
Further reasons for version 4.1.1. belongs to the fact that the implementation of the locus of maximal order was already implemented in the master's thesis of the author [41] in that version.

In Section A.1, we give pseudo code representations of the calculation of the maximal order, the locus of maximal order and the respective refined versions we discussed in Section 2.4.2. We follow the argumentation of [35] and discuss optimizations, too. Note, that the implementation is explicitly written for the variant of the CJS algorithm of [35], so we prefer the horizontal components in the arithmetic case like discussed in Section 5.6.

Afterwards, we give in Section A.2 a pseudo code of an implementation in Singular of the CJS algorithm. This implementation is a test environment for a better study of the algorithm. Furthermore, we present the main ideas of the data structures and how to handle the recursion, which we had seen in Section 5.6. Some comparisons can be found in Section B.4.

In Section A.4, we give a pseudo code of the algorithm of Hu, which we discussed in Section 5.5. In there, we give different implementations. The first is a general variant of Hu's algorithm, and the second is explicitly written for our setting of resolution of determinantal singularities.

At the end of this chapter, we discuss in Section A.5 how to implement the generic determinantal case and our new algorithm for resolving determinantal singularities.

# A.1. Implementational Aspects of the locus of maximal order and the locus of maximal log-refined order

In [41], the author describes how to implement the algorithms of maximal order in the arithmetic case, which are given in [35]. We recall the main ideas of both work together with new optimization in this topic.

As the first version of the implementation, the discussed version is also implemented in SINGULAR version 4.1.1. [30] which was published in 2018.

In this section, we assume that $X$ is a reduced excellent Noetherian scheme (not necessarily of dimension two), embedded in some excellent regular scheme $Z$.

## A.1.1. MaxOrd Calculations

In this subsection, we consider the computation of the locus of maximal order, described in [35] with additional optimizations by the author.

**Construction A.1.1** ([35, Construction 3.7])**.** Let $0 \neq J \subset B = A[\underline{x}]$ be an ideal, where $A$ is a field or a Dedekind ring. Let $X := V(J)$ and let $d := \max\{\mathrm{ord}_x(X) \mid x \in X\} \geq 1$. If $d = 1$. Then the locus of maximal order of $X$ is $X$ itself. Assume $d > 1$.

- If $B$ contains a field $k$, then Max-ord$(X)$ = Max-ord$(J)$ equals

$$\Delta^{d-1}(J) := V(\mathcal{D}f \mid f \in J, \mathcal{D} \in \mathrm{Diff}_{\mathbb{Z}}^{\leq d-1}(B)).$$

- If $B$ does not contain such a field, i.e, $A$ has mixed characteristic. Then let $F := \mathrm{Frac}(A)$ be the field of fractions of $A$ and we set $B_F := B \otimes_A F = F[\underline{x}]$. If the maximal order of $V(J \cdot B_F)$ equals $d$, then $\Delta^{d-1}(J \cdot B_F)$ is the set of horizontal irreduzible components of Max-ord$(X)$.

For the vertical components, let $p \in A$ be a prime element. We pass from $B = A[\underline{x}]$ to $B_p := B \otimes_A \mathrm{gr}_p(A) = (\mathrm{gr}_p(A))[\underline{x}]$. An element $\sum f_{\boldsymbol{a}} \underline{x}^{\boldsymbol{a}} \in B$ is mapped to $\sum F_{\boldsymbol{a}} \underline{x}^{\boldsymbol{a}}$. Note that

$$F_{\boldsymbol{a}} = F_{\boldsymbol{a}}(P) = \mathrm{in}_p(f_{\boldsymbol{a}}) \in \mathrm{gr}_p(A) \cong k_p[P],$$

with $P := \mathrm{in}_p(p) \mod p^2$ und $k_p := A/p$ the residual field. Note that $B' \cong k_p[p, \underline{x}]$ is a polynomial ring over the field $k_p$.

If max-ord$(V(J \cdot B_p)) = d$, then $\Delta^{d-1}(J \cdot B_p)$ is the set of vertical components of the locus of maximal order of $X = V(J)$ with $I \cap A = \langle p \rangle_A$. Applying this procedure for every prime $p \in A$, we obtain Max-ord$(X)$.

First, we give the pseudocode of Algorithm 1 in [35], which differs in specially optimized lines.

**Remark A.1.2** (Algorithm 14)**.** Our implementation of Algorithm 14 gets the following input:

- An ideal $I_Z \subseteq \mathbb{Q}[x_1, \ldots, x_n]$, such that $Z = V(I_Z)$ is equidimensional and regular
- An ideal $I_X \subsetneq \mathbb{Q}[\underline{x}]$, such that $I_Z \subset I_X$.

**Algorithm 14** Max-ord(ideal $I_Z$, ideal $I_X$) (char $K = 0$)

```
1:  I_temp = I_X;
2:  if (I_Z == ⟨0⟩) then
3:      I_max = ⟨1⟩;
4:      max-ord = 0;
5:      while (I_temp ≠ ⟨1⟩) do
6:          I_max = I_temp;
7:          I_temp = I_temp + ⟨∂f_i/∂x_j | 1 ≤ j ≤ n, 1 ≤ i ≤ r⟩;        ▷ ⟨f_1, ... f_r⟩ = I_temp
8:          max-ord = max-ord +1;
9:          return(max-ord, I_temp);
10: else
11:     L_1 = GenerateL1(codim(Z), J(I_Z), I_X, I_Z);
12:     max-ord = 1, thisord = 0, I_max = ⟨0⟩;
13:     for M ∈ L_1 do
14:         I_temp = I_X;
15:         while I_temp + I_Z ≠ ⟨1⟩ do;
16:             I_old = I_temp;
17:             I_temp = I_temp + ⟨∂f_i/∂y_j | 1 ≤ j ≤ s, 1 ≤ i ≤ r⟩;    ▷ See Remark A.1.5.
18:             I_temp = sat(I_temp, det(M));
19:             thisord = thisord +1;
20:         if (thisord ≥ max-ord) then
21:             if (thisord == max-ord) then
22:                 I_max = I_max ∩ I_old;
23:             else
24:                 max-ord = thisord;
25:                 I_max = I_old;
26:         thisord = 0;
27: return(max-ord, I_temp);
```

$J(I_Z)$ in line 11 denotes the jacobian matrix of $I_Z$. *(End of Remark A.1.2.)*

The covering named $L_1$ is calculated (line 11) by adding elements to the covering until the covering fulfills the following conditions.

- $L$ is the set of $\operatorname{codim}(Z)\times \operatorname{codim}(Z)$ submatrices of the Jacobian matrix of $I_Z$.
- $L_1 \subset L$ such that $X \subset \bigcup_{M\in L_1} D(\det(M))$, where $D(\det(M))$ is the principal Zariski-open set determined by $\det(M)$.

If there is a 1 entry in the jacobian matrix, we only need to consider the minors with this 1 entry. If there is no 1 entry, we add all the submatrices of $\mathcal{J}(I_Z)$ until $L_1$ fulfills all conditions. The whole pseudo code of the GenerateL1-function is described in Algorithm 15.

**Remark A.1.3** (Algorithm 15)**.** The input consists of

- an integer $\operatorname{codim}(Z)$, describing the the codimension of $Z$,
- the considered Jacobian matrix $J_Z$ of $I_Z$,
- the ideal $I_X$,
- the ideal $I_Z$ and
- optional: a list of prime factors (only needed for InterestingPrimes in the ConditionOfL1 subcall).

The output is the covering $L_1$ with the required properties.

**Algorithm 15** GenerateL1(int codim($Z$), matrix $J_Z$, ideal $I_X$, ideal $I_Z$)

containsOne = false;
**for** ($i = 1$;$i \leq$ #rows; $i$++) **do**
    **for** ($j = 1$; $j \leq$ #cols; $j$++) **do**
        **if** ($J_Z[i,j] == 1$) **then**
            containsOne = true;
            $h_i = x_j - f_i$;
            **for** ($k = 1$; $k \leq$ #rows; $k$++) **do**
                **if** ($k \neq i$) **then**
                    $f_k(x_j \mapsto h_i)$; ▷ Substitute $x_j$ by $h_i$
            break;
**if** containsOne == true **then**
    return(codim(Z) square submatrices of $J_Z$ which contains the 1-entry);
L1 = $\emptyset$;
**while** (ConditionOfL1(L1) == false) **do**
    L1.add(arbitrary invertible codim($Z$) $\times$ codim($Z$) submatrix $M$ of $J_Z$, not contained in L1);
return(L1);

*(End of Remark A.1.3.)*

**Remark A.1.4** (Algorithm 16)**.** The ConditionOfL1 algorithm differs, if we call it from the InterestingPrimes or any MaxOrd-Calculation. If we are in a call of an *InterestingPrimes*-call, we have an optional parameter stored in the list #. In this case, we first calculate the product of all prime factors. Then we define

$$I_{\text{tmp}} = \bigcup_{M \in L_1} \det(M) \cup V(p_1 \cdots p_\alpha).$$

The pseudocode of this function is described in Algorithm 16. The input consists of

- the potential covering $L_1$, which is to check if it fulfills the conditions,
- the considered Jacobian matrix $J_Z$ of $I_Z$ and
- optional: a list of prime factors (only needed for InterestingPrimes).

The output is a boolean. It is true if $L_1$ fulfills the required properties and false otherwise.

**Algorithm 16** ConditionOfL1(list L1, matrix $J_Z$, optional: list #)

1: $I_{\text{tmp}} = \bigcap_{M \in L_1} \det(M)$;
2: **if** (primelist $= \emptyset$) **then**
3: **if** (deg(std($J_X + I_{\text{tmp}}$))[1] == 0) **then** ▷ $X \cap \text{tmp} = \emptyset$
4: return(true);
5: **else**
6: return(false);
7: primeproduct = $\Pi_{p \in \text{primelist}} p$;
8: $I_{\text{tmp}} = I_{\text{tmp}} \cap$ primeproduct;
9: **if** ($I_{\text{tmp}} == \langle 1 \rangle$) **then**
10: return(true);
11: **else**
12: return(false);

*(End of Remark A.1.4.)*

**Remark A.1.5** ([15, Remark 3.3])**.** The minor $\det(M)$ is only invertible on a principal open set, the differentiation can be calculated as follows:
We start the computation by determining the matrix of cofactors $A$, a square matrix satisfying

$$A \cdot M = q \cdot E_{\text{codim}(Z)},$$

where $q = \det(M)$ and $E_{\text{codim}(Z)}$ denotes the unit matrix of size $\text{codim}(Z)$. On $D(q)$, $\frac{1}{q} \cdot A$ is exactly the inverse of $M$. The system of parameters $\{y_1, \ldots, y_s\}$ of our choice is induced by the set $\{x_i \mid i \text{ is not a column in } M\}$.
Let $g_1, \ldots, g_t \in \mathbb{Q}[\underline{x}]$ be a set of generators of $I_Z$. We want to choose a set of generators $\overline{f_1}, \ldots \overline{f_r} \in \mathbb{Q}[\underline{x}]/I_Z$ for the ideal $I_Z \cdot \mathbb{Q}[\underline{x}]/I_Z$ and we choose representatives $f_1, \ldots, f_r \in \mathbb{Q}[\underline{x}]$ for these. For better readability, we assume that $M$ involves exactly the last columns of the Jacobian matrix so that the indices of $y_i$ and the corresponding $x_i$ coincide. By applying the chain rule, we get the following derivatives

$$q \cdot \frac{\partial f_i}{\partial y_j} = q \cdot \frac{\partial f_i}{\partial x_j} - \sum_{\substack{k \text{ column of } M, \\ \ell \text{ row of } M}} \frac{\partial g_\ell}{\partial x_j} A_{\ell k} \frac{\partial f_i}{\partial x_k} \mod I_Z.$$

The discard the extra factor $q$, we need to saturate the resulting ideal with $\langle q \rangle$. For more information see [14, Construction 3.15].

For the arithmetic case, we want to systematically determine the primes relevant to the locus of maximal order by trying to mimic the MaxOrd algorithm for characteristic 0. On $\mathbb{Z}$, we are missing the derivative with respect to a prime. We

call primes arising from the following Algorithm 17 interesting as the bad primes will eventually appear among those in the course of resolution. However, not all of these arising primes have to be bad.

**Remark A.1.6** (Algorithm 17)**.** The Algorithm 17 shows how to compute these interesting primes. [35, Lemma 4.6] shows that the returned set contains all bad primes. The input of InterestingPrimes consists of

- an ideal $I_Z = \langle g_1, \ldots, g_t \rangle$, such that $Z = V(I_Z)$ is equidimensional and regular and
- an ideal $I_X = \langle g_1, \ldots, g_t, f_1, \ldots, f_r \rangle \subsetneq \mathbb{Z}[\underline{x}]$.

The output is a set of interesting primes which contains all bad primes for $X$.

**Algorithm 17** InterestingPrimes(ideal $I_Z$, ideal $I_X$)

1: $I_{\text{tmp}} = \langle f_1, \ldots, f_r \rangle$;
2: $I_{\text{int}} = \langle 0 \rangle$;
3: **if** $(I_Z \cap \mathbb{Z} \neq \langle 0 \rangle)$ **then**
4: return(primefactors(generator of the principal ideal $I_Z \cap \mathbb{Z}$));
5: **else if** $(I_Z == \langle 0 \rangle)$ **then**
6: **while** $(I_{\text{int}} == \langle 0 \rangle)$ **do**
7: $I_{\text{tmp}} = I_{\text{tmp}} + \langle \frac{\partial F_i}{\partial x_j} \mid 1 \leq j \leq n, 1 \leq i \leq q \rangle$; ▷ $\langle F_1, \ldots, F_q \rangle = I_{\text{tmp}}$
8: $I_{\text{int}} = I_{\text{tmp}} \cap \mathbb{Z}$;
9: return(primefactors(generator of the principal ideal $I_{\text{int}}$));
10: resultlist = $\{p_1, \ldots, p_\alpha\}$;
11: L1 = GenerateL1(codim($Z$), $J(Z)$,$I_X$,$I_Z$,resultlist);
12: **for** $M \in$ L1 **do**
13: $I_{\text{int}} = \langle 0 \rangle$;
14: $I_{\text{tmp}} = \langle f_1, \ldots, f_r \rangle$;
15: **while** $(I_{\text{int}} == \langle 0 \rangle)$ **do**
16: $I_{\text{tmp}} = I_{\text{tmp}} + \langle \frac{\partial F_i}{\partial y_j} \mid 1 \leq j \leq s, 1 \leq i \leq q \rangle$; ▷ $\langle F_1, \ldots, F_q \rangle = I_{\text{tmp}}$
17: $I_{\text{int}} = (I_{\text{tmp}} + I_Z) \cap \mathbb{Z}$;
18: **if** $(I_{\text{int}} \neq \langle 1 \rangle)$ **then**
19: resultlist = resultlist $\cup$ primefactors(generator of the principal ideal $I_{\text{int}}$;
20: return(resultlist);

In line 10 $p_\ell$ are the primefactors appearing in coefficients of $g_1, \ldots, g_t$ for some $1 \leq \ell \leq \alpha$ and $\alpha \in \mathbb{Z}_{\leq 0}$.

In line 16, $(y_1, \ldots, y_s)$ denotes the system of parameters in $D(\det(M)) \cap D(p_1 \cdots p_\alpha)$ which is induced by the subsystem of $(\underline{x})$ of variables which do not correspond to a column of $M$.

*(End of Remark A.1.6.)*

We can finally present the pseudo code of the computation of the locus of maximal order in the arithmetic case. We call the algorithm MaxOrdArith (see Algorithm 18). This implementation is explicitly written for the CJS implementation in Section A.2 (see also Section 5.6 for the theoretical details). Therefore the output is not necessarily the whole locus of maximal order in the arithmetic case but the horizontal components of this locus if there are some and the vertical components otherwise.

**Remark A.1.7** (Algorithm 18)**.** The input of Algorithm 18 contains

- an ideal $I_Z \subseteq \mathbb{Z}[x_1, \ldots, x_n]$ such that $Z = V(I_Z)$ is equidimensional and regular and
- an ideal $I_X \subsetneq \mathbb{Z}[\underline{x}]$ such that $I_Z \subset I_X$.

The output is a pair $(\text{max-ord}(X), L)$, where $L$ is a list such that $L[i] = (p_i, I_i)$ with

- either $p_i = 0$ for all $i$ and $\text{Max-ord}(X)_{\text{hor}} = \bigcup_i V(I_i) \neq \emptyset$
- or $\text{Max-ord}(X) = \text{Max-ord}(X)_{\text{vertcal}} = \bigcup_i V(I_i)$ such that $I_i$ has been detected locally at the bad prime $p_i$.

Recall, that we ignore vertical components of the locus of maximal order, if there are horizontal components.

**Algorithm 18** MaxOrdArith(ideal $I_Z$, ideal $I_X$)

MaxOrd0 = MaxOrd($I_Z \otimes \mathbb{Q}, I_X \otimes \mathbb{Q}$), maxord = MaxOrd0[1];
$I_{\max}$ = MaxOrd0[2] $\cap \mathbb{Z}[\underline{x}]$, RetList[1] = $(0, I_{\max})$;
PrimeList = InterestingPrimes($I_Z, I_X$);
**for** $p \in$ PrimeList **do**
    $J_X$ = ReplaceCoeffs($I_X, p$), $J_Z$ = ReplaceCoeffs($I_Z, p$);
    **if** ($I_Z == \langle 0 \rangle$) **then**
        DiffList = HasseDeriv($\langle 0 \rangle, J_X, (\underline{x}, P), 0$);
        $m$ = size(DiffList);
        **for** ($i = 1; i \leq m; i++$) **do**
            DiffList[$i$] = ideal(substitute(DiffList[$i$],$P, p$));
        **while** (DiffList[$m$] == $\langle 1 \rangle$) **do**
            $m = m - 1$;
        **if** ($m \geq$ maxord) **then**
            $I_{\max}$ = DiffList[m];
            **if** ($m >$ maxord) **then**
                RetList = $\emptyset$, maxord = $m$, RetList[1] = $(p, I_{\max})$;
            **else if** RetList[1][1] $\neq 0$ **then**; ▷ no horizontal components
                RetList[last] = $(p, I_{\max})$;
    **else**
        L1 = GenerateL1(codim($Z$), $\mathcal{J}(J_Z)$,$J_X$,$J_Z$);
        locord = 1;
        **for** $M \in$ L1 **do**
            DiffList = HasseDeriv($J_Z, J_X, \underline{y}, M$); ▷ $\underline{y}$ like in Remark A.1.5
            $m$ = size(DiffList);
            **for** ($i = 1; i \leq m; i++$) **do**
                Difflist[$i$] = ideal(substitute(DiffList[$i$],$P, p$));
            **while** (DiffList[$m$] == $\langle 1 \rangle$) **do**
                $m = m - 1$;
            **if** ($m >$ locord) **then**
                $I_{\max}$ = DiffList[$m$];
                locord = $m$;
            **else if** ($m ==$ locord) **then**
                $I_{\max} = I_{\max} \cap$DiffList[$m$];
        **if** (locord $\geq$ maxord) **then**
            **if** (locord $>$ maxord) **then**
                RetList = $\emptyset$, maxord = $m$, RetList[1] = $(p, I_{\max})$;
            **else if** (RetList[1][1] $\neq 0$) **then** ▷ no horizontal components
                RetList[last] = $(p, I_{\max})$;
return(maxord,RetList);

*(End of Remark A.1.7.)*

The submethod ReplaceCoeffs replaces each appearing coefficient $c \in \mathbb{Z}$ by $\frac{c}{p^\ell} P^\ell$,

where $\ell$ is maximal, and $P$ is a new variable. The resulting ideals $J_X$ and $J_Z$ are in $\mathbb{Z}[\underline{x}, P]$.

Since the $p$-th usual derivatives vanish in characteristic $p$, it is standard to use Hasse-Schmidt derivatives in characteristic $p$.

**Definition A.1.8** ([43, Sections 2.5 and 2.6])**.** Let $S = K[\underline{x}] = K[x_1, \ldots, x_n]$ be a polynomial ring over a field $K$. Let $F = F(\underline{x}) \in S$. We introduce new variables $(\underline{t}) = (t_1, \ldots, t_n)$ and consider

$$F(\underline{x} + \underline{t}) = F(x_1 + t_1, \ldots, x_n + t_n) = \sum_{\underline{a} \in \mathbb{Z}^m_{\geq 0}} F_{\underline{a}}(\underline{x}) \underline{t}^{\underline{a}}.$$

The Hasse-Schmidt derivative of $F$ by $\underline{x}^{\underline{a}}$ is defined by the coefficient of $\underline{t}^{\underline{a}}$ in the previous expansion.

**Remark A.1.9.** We can relate the Hasse-Schmidt derivatives to usual derivatives via the following symbolic computation

$$\frac{\partial}{\partial \underline{x}^{\underline{a}}} = \frac{1}{a_1! \cdots a_n!} \left(\frac{\partial}{\partial x_1}\right)^{a_1} \cdots \left(\frac{\partial}{\partial x_n}\right)^{a_n}.$$

That is why Hasse Derivatives often were used in positive characteristics instead of the usual ones.

Algorithm 19 shows how to compute Hasse derivations in our setting.

**Remark A.1.10** (Algorithm 19)**.** The input contains

- an ideal $I_Z = \langle g_1, \ldots, g_t \rangle \subseteq \mathbb{Z}[\underline{x}]$ such that $Z = V(I_Z)$ is equidimensional and regular,
- an ideal $I_X = \langle g_1, \ldots, g_t, f_1, \ldots, f_r \rangle \subsetneq \mathbb{Z}[\underline{x}]$,
- a system of parameters $(\underline{y})$ on $D(\det(M))$ and
- a matrix $M$ which is a square submatrix of the Jacobian matrix of $I_Z$.

The output is a list where the $i$-th entry are the Hasse derivatives up to the $i$-th derivatives of $I_X$.

**Algorithm 19** HasseDeriv(ideal $I_Z$, ideal $I_X$, list y , matrix M)

1: **if** $(I_Z == \langle 0\rangle)$ **then**
2: **for** $(j = 1; j \leq r; j++)$ **do**
3: $F_j(\underline{y}, \underline{t}) = f_j(x_1 + t_1, \ldots, x_n + t_n)$; ▷ new variables $t_i$
4: $i = 1$;
5: tempid $= f_1, \ldots, f_r$;
6: **while** $((i == 1)$ or (tempid $\neq$ RetList$[i-1]))$ **do**
7: RetList$[i]$ = tempid;
8: **for** $(\underline{a} \in \{\underline{b} \in \mathbb{Z} \mid |\underline{b}| == i\})$ **do**
9: tempid = tempid, { coefficients of $\underline{t}^{\underline{a}}$ in $F_1, \ldots, F_r$};
10: $i = i + 1$;
11: **else if** $(I_Z \neq \langle 0\rangle)$ **then**
12: $I_{\text{tmp}} = I_X$;
13: **for** $(i = 1; i \leq r; i++)$ **do**
14: L$[i]$ = $(f_i, (\underbrace{0, \ldots, 0}_{\#(\underline{y})}))$;
15: old $= 0$, cur $= r$;
16: **while** $(I_{\text{tmp}} \cap \mathbb{Z} == \langle 0\rangle)$ **do**
17: **for** (old $\leq i \leq$ cur) **do**
18: **for** $(y_j \in (\underline{y}))$ **do**
19: $(f_{\text{tmp}}, \text{note}) =$ L$[i]$;
20: note$[j]$ = note$[j]$+1;
21: $f_{\text{tmp}} = \frac{1}{\text{note}[j]} \cdot \frac{f_{\text{tmp}}}{\partial y_j}$;
22: L.append($f_{\text{tmp}}$,note);
23: $I_{\text{tmp}} = I_{\text{tmp}} + \langle f_{\text{tmp}}\rangle$;
24: $I_{\text{tmp}} = \text{sat}(I_{\text{tmp}}, \det(M))$;
25: RetList.append($I_{\text{tmp}}$);
26: old = cur, cur = size(L);
27: return(RetList);

Note, that RetList[-1] doesn't throw an error in line 6, since the boolean function OR evaluates $i == 1$ first. If this is true, the second term of the OR-function will not evaluated anymore. *(End of Remark A.1.10.)*

## A.1.2. Max-$\nu$ Calculations

With Construction A.1.11 we can reduce the calculation of the maximal refined-order locus to the locus of maximal order for which we apply Construction A.1.1.

**Construction A.1.11** ([35, Construction 3.9])**.** Let $0 \neq J \subset B = A[\underline{x}]$ be an ideal, where $A$ is a field or any principal Dedekindring. Set $X = V(J)$. Assume, like in

Definition 2.4.8

$$\text{max-}\nu(X) = (\alpha, \delta) \in \mathbb{N}.$$

We provide an inductive construction of Max-$\nu(X)$ depending on $a := a(Z) := N - \alpha = \dim(Z) - \alpha \geq 0$.
If $a(Z) = 0$, then Max-$\nu(X)$ coincides with Max-ord$(X)$ and we apply Construction A.1.1.
So we can assume $a(Z) \geq 1$. Then $J$ has order 1 at every point in $X$. So we can descent in the dimension of the ambient space locally at every point. Hence, for every point $q$ in Spec$(B)$ there exists a differential operator $\partial = \partial(q)$, such that $(\partial J) \cdot B_{I_q} = B_{I_q}$, with $\partial J = \langle \partial h \mid h \in J \rangle$ and $B_{I_q}$ denotes the localization of $B$ at the ideal of $q$.
Let $(f_1, ...f_r)$ be a set of generators of $J$. This implies that there is a refined finite open covering $X \subset \bigcup_\ell U_\ell$, such that in each $U_\ell$ we have

$$X_\ell := X \cap U_\ell \subset V(f_{j(\ell)}) \cap U_\ell =: Z_\ell.$$

Furthermore, $V(f_{j(\ell)})$ is regular in $U_\ell$ for $j(\ell) \in \{1, ..., r\}$ .

- If $A$ is a perfect field, we can apply the Jacobian criterion (Theorem 3.1.4) for each generator of $J$ separately, and we consider the derivatives with respect to the variables $(\underline{x})$.

- If $A$ is a non-perfect field, we apply Zariski's Jacobian criterion (see [82, Theorem 11]). This involves a $p$-basis for $A$ and uses that there are only infinitely many non-zero coefficients of each generator $f_1, ...f_r$.

- If $A$ is a principal ideal Dedekindring, which is not a field (e.g., $A = \mathbb{Z}$), there are only finitely many prime elements $p_1, ..., p_\rho$ in $A$, appearing in the coefficients of $f_i$ for each $i$ and for all $j \in 1, ..., \rho$. Then we substitute each $a \in A$, appearing in the generators by $\frac{a}{p_j^\ell} P^\ell$, with $\ell$ is the maximal choice and $P$ is a new variable. Then we can apply the Jacobian criterion and use the derivative by $P$.

Note that $\dim(Z_\ell) = \dim(Z) - 1$ implies $a(Z_\ell) = a(Z) - 1$, since $\nu_{\text{ref}}$ and max-$\nu(X)$ do not depend on the embedding.
By Induction, we can calculate Max-$\nu(X_\ell)$ and max-$\nu(X_\ell)$. if $\max_\nu(X_\ell) = \text{max-}\nu(X)$, then Max-$\nu(X_\ell) =$ Max-$\nu(X)$. We obtain Max-$\nu(X)$ by gluing the relevant affine charts.

**Remark A.1.12** ([35, Remark 4.1]). 1. If the maximal order of $X$ equals one, the scheme can locally be embedded into a regular hypersurface, and it thus suffices to find an open covering such as using a single such hypersurface on an open set is possible. The equation of this hypersurface can then be added to the generators of the ideal of the ambient space $Z$.
Iterating this process, we descent in ambient space as long as we have not reached to minimal appearing $a_x$ at any point $x \in X$. At these points, the first entry of the invariant $N - a_x$ attains a maximal value.

2. Again, computing the locus of maximal order of $X$ with the new ambient space resulting from (1.), we reach the locus of maximal refined order.

3. By considering the exceptional divisors, we obtain the locus of log-refined order.

4. We label the irreducible components of Max-$\nu^O(X)$ using the history of the resolution process and detect the locus of components that have the smallest label. Since $X$ is reduced and of dimension two, the latter components have at most dimension one. Hence, we either blow-up the smallest label locus or prepare it to become weakly permissible by blowing up closed points.

This yield the following construction.

**Construction A.1.13.** Let $X$ and $Z$ be as above. We can calculate Max-$\nu(X, Z)$ as follows:

1. Calculate max-ord$(X, Z)$ and Max-ord$(X, Z)$.

2. a) Case max-ord$(X, Z) > 1$ :
The equalities Max-ord$(X, Z)$ = Max-$\nu(X, Z)$ and max-ord$(X, Z)$ = max-$\nu(X, Z)$ hold.
So we already have calculated Max-$\nu(X, Z)$ by calculating Max-ord$(X, Z)$.

   b) Case max-ord$(X, Z) = 1$:
Let $f_1, \ldots, f_r$ be a standard basis of $I_X$ such that $X = V(I_X)$. Take a regular generator $f_i$ of $X$ and add the equation of $f_i$ to the generators of $Z$, so we have $\tilde{Z} = V(\langle f_i \rangle + I_Z)$.
If $I_{\tilde{Z}} \neq I_X$, go to (1) where $\tilde{Z}$ plays the role of $Z$.
Otherwise, we are finished, too.

By counting the number of runs through this process $a$ (number of loops which run through the case (2)(b)), we get

$$\text{max-}\nu(X) := (\underbrace{1,\ldots,1}_{a}, \text{max-ord}(X,\tilde{Z})).$$

**Remark A.1.14** (Algorithm 20)**.** Like described in [35], we need for $V(f_1,\ldots,f_r)$ for $i > 1$ a refinement of the ordering. Therefore we show how to compute this refinement of the ordering and the locus of maximal refined order (see Construction A.1.13). The input is

- an ideal $I_X$ describing $X$ and
- an ideal $I_Z$ describing the ambient space.

The output is a list with the same form as Max-ord$(I_X, I_Z)$, i.e., invariant and locus of maximal order and the ambient space.

**Algorithm 20** Max-$\nu$( ideal $I_X$, ideal $I_Z$)

```
1: counter = 0;
2: descented_I_Z = ⟨0⟩;
3: while (max-ord(I_X,descented_I_Z) == 1) do
4:     Choose one generator f_i of a standard basis of I_X which is regular.
5:     descented_I_Z :=descented_I_Z + ⟨f_i⟩;
6:     counter = counter +1;
7: invariant = list(1,...,1,max-ord(I_X,descented_I_Z));   (1,...,1: counter)
8: return list(list(invariant, Max-ord(I_X,descented_I_Z,)), descented_I_Z);
```

If max-ord$(I_X) > 1$, then Max-$\nu(I_X)$ = Max-ord$(I_X)$. So we only need to consider the interesting case max-ord$(I_X) = 1$.
So we assume max-ord$(I_X) = 1$. Then we compute the set of generators of a standard basis of $I_X = V(f_1,\ldots,f_r)$. Furthermore. we compute the subset of generators for which max-ord$(f_i) = 1$ holds. This set is the set of generators of our new ambient space descented_$I_Z$. Then Max-ord$(I_X, I_Z)$ = Max-$\nu(\langle 0\rangle, I_X)$ and we can use the method which computes Max-ord$(I_X, I_Z)$.

*(End of Remark A.1.14.)*

Section 2.4.2 defines the refined order as a pair $(N-a_x, d_x)$. For this construction, it is easy to show that it is upper semi-continuous. For practical reasons, it might be more sense-full to encode the number $N - a_x$ unary as a $(N - a_x)$-tuple of 1s.

This belongs to the fact that it is more or less a truncated Hironaka invariant, so in practice, it is easier to use the same data types in order to be able to reuse some methods of the Hironaka setting. Therefore, we have implemented the same 'value' in a different encoding, so it is not a mistake.

### A.1.3. Max-$\nu^O$ Calculations

As we have seen when discussing Hironaka-style resolution above, the version in [35] of the CJS algorithm distinguishes old and new exceptional divisors, too. So we consider the locus of maximal refined order with respect to the old boundary components instead of the locus of maximal order as the measure how singular a point is. If the Input is generated by a single polynomial, i.e., if we consider a hypersurface $X = V(f)$, it suffices to use Max-ord$^O(\cdot)$.

**Construction A.1.15** ([35], Remark 4.1)**.** Let $X$ be an excellent noetherian scheme, embedded in a regular ambient space $Z$. We can calculate Max-$\nu^O(X, Z)$ as follows:

1. Calculate max-$\nu(X, Z)$ and Max-$\nu(X, Z)$.
2. Define $O(X) := \{B_i \in \mathcal{B} \mid B_i \text{ is an old component and } X \cap B_i = \emptyset\}$.
3. If $|O| = 0$, we have Max-$\nu(X, Z)$ = Max-$\nu^O(X, Z)$ and
$$\text{max-}\nu^O(X, Z) = (\text{max-}\nu(X, Z), 0).$$
4. If $|O| > 0$, Max-$\nu^O(X, Z)$ = Max-$\nu(X, Z) \cap O(X)$ and
$$\text{max-}\nu^O(X, Z) = (\text{max-}\nu(X, Z), |O(X)|).$$

**Remark A.1.16** (Algorithm 21)**.** In Algorithm 21, we intersect the locus of maximal order resp. maximal refined order, with only necessary old boundary components. So we do not intersect with all components, but all the old components, which possibly are not transversal to the strict transform.

**Algorithm 21** Max-$\nu^O$( ideal $I_X$, ideal $I_Z$)

1: OldComps = $\{var(i) \mid boundary[2][i] < 0$, and $var(i) \cup I_X \neq \langle 1 \rangle\}$;
2: invariant = (max-$\nu(I_X, I_Z)$, |OldComps|);
3: descented_$I_Z$ = Max-$\nu(I_X, I_Z)[2]$ ;
4: return(list(invariant,Max-$\nu(I_X, I_Z)\cap$ OldComps),descented_$I_Z$);

*(End of Remark A.1.16.)*

The described singular code is explicitly written for the CJS-Algorithm (see 5.6), so we have to discuss this Algorithm before we are able to construct the old components explicitely.

## A.2. Singular Implementation of the CJS-Algorithm

In this section, we discuss a dimension-free implementation of the CJS-Algorithm. We present two theoretically possibilities to handle the recursion in Section A.2.1. Then we present the pseudo code of the main method in Section A.2.2. Finally, we discuss the used submethods in Section A.2.3.

Recall that first, the CJS-Algorithm calculates a locus where the singularity is maximal with respect to our measure max-$\nu^O(\cdot)$. We follow the approach of [35] and choose the log-refinement of the order function, which we discussed in Definition 2.4.13.
So we calculate $Y := \text{Max-}\nu^O(I_X)$. Then there is a decomposition in irreducible components $Y = Y_1 \cup \ldots \cup Y_n$. The irreducible components are stored in a list together with a labeling. At the beginning, every label equals 0.

After that, our implementation checks whether the procedure has already been finished. This is the case, if $\text{max-}\nu^O(X) = (1, \ldots, 1, 0)$.
If we are already finished, our algorithm returns the initial chart.

If not, then the CJS algorithm calculates recursively a canonical center with another CJS-call with the corresponding input (ambient space $I_Z$, ideal $Y$, boundary $\mathcal{B}$).

This recursion *tower* of CJS-calls stabilize as soon as we get an $\mathcal{B}$-permissible center $\text{Max-}\nu^O(Y_{(k)}) = Y_{(k)}$ for a recursion-level $k$. And then we have to blow-up in $Y_{(k)}$.

When the transform of the canonical center of the upperst CJS-call is $\mathcal{B}$-permissible, we have already gotten an $X''$ which is the transform with respect to the sequence of blow-ups. Now we can calculate the blowing Up of the ambient space $Z''$, which contains $X''$ in this $\mathcal{B}$-permissible center.

The diagram in Figure A.1 illustrates the situation.

If the invariant has not improved, our algorithm calculates $\text{Max-}\nu^O(X_1)$. The next step is to calculate a decomposition of this locus in irreducible components. These labels are determined in each step of the sequence of blow-ups which are

$$
\begin{array}{llll}
\mathcal{B}_1 & \mathcal{B}'' & \mathcal{B}' & \mathcal{B} \\
Z_1 \to & Z'' \to \ldots \to & Z' \to & Z \\
\cup & \cup & \cup & \cup \\
X_1 \to & X'' \to \ldots \to & X' \to & X \\
 & \cup & \cup & \cup \\
 & Y'' \to \ldots \to & Y' \to & Y = \text{Max-}\nu^O(X) \\
 & & \vdots & \vdots \\
 & & \cup & \cup \\
 & & A' \to & A = \text{Max-}\nu^O(\cdot) \\
 & & & \cup \\
 & & & B = \text{Max-}\nu^O(A) (= \text{Max-}\nu^O(B)).
\end{array}
$$

Figure A.1.: CJS recursion as a tower

done in order to get a $\mathcal{B}$-permissible center. These irreducible components, which do not lie over the center, are isomorphic to their image since the blow-up is an isomorphism outside the center. Therefore these components inherit their labels. These components lying over the center are subdivided into two types. The first type is the components dominating the center. Components of this type inherit their label. Components not dominating the center get the number of the blow-up as a label. Theoretically a component could posses different labels in different CJS-calls. But in practice it suffices that $j < i$ or $j = i$ hold for labels $i$ and $j$ of different components, if it should hold. The exact values of the labels does not matter.

Afterwards, we calculate the union of all irreducible components of Max-$\nu^O(X_1)$ with the same label, i.e., we get

$$\text{Max-}\nu^O(X_1) =: Y_1 = Y_1^{(0)} \cup Y_1^{(1)},$$

where $Y_1^{(0)}$ denotes the union of the label 0 components and $Y_1^{(1)}$ denotes the union of the label 1 components of Max-$\nu^O(X_1)$.

The next pass through the loop (see Figure 5.6) we replace $Y$ by $Y_1^{(k)}$, where $k = \min_{i \in \{0,1\}}\{i \mid Y_1^{(i)} \neq \emptyset\}$.

If the invariant has improved, we can delete all labels and start at the first level of recursion, like at the beginning of the CJS-Algorithm with $X_1$ instead of $X$, $Y_1 :=$ Max-$\nu^O(X_1)$ instead of $Y$ and the divisor of the exceptional divisor instead of an empty boundary.

## A.2.1. How to handle the recursion

There are at least two possible data structures to handle the recursion of the main algorithm (see Figure 8.1).

The first method is described above in Figure A.1 with the additional data structures as follows:

We store the towers of centers, boundaries, and invariants in several lists. In the case of the tower of the center, this list's first element is a list containing the irreducible components of Max-$\nu^O(X) =: Y$ together with their labels. These labels have the value 0 at the beginning of the algorithm. This list of centers' second element is the irreducible components of Max-$\nu^O(Y) =: Z$ and their labels. The last element of the list of centers is the unique irreducible component of $B$, namely $B$ itself, with the label 0. The lowest level of the tower theoretically needs no label, but for practical reasons, it is easier to handle every level of the *tower* equally.
Then the ambient space $Z$ is blow-up with center $B$.

After that, we have to label the irreducible components of the locus of maximal refined order of $X$ and the labels of the irreducible components obtained in a sublist of the list of centers. The next step is to evaluate the situation, and we have to check if $A'$ is $\mathcal{B}$-permissible or not. If it is the case, we can blow-up the ambient space with center $A'$ and so on. If not, we need to generate a new *tower* below $A'$ as described above. The tower of boundaries is a list of boundaries with the labeling old or new for each component, where the boundary of an arbitrary level of recursion (at the beginning) consists of the new components of the boundary of the level of recursion above. The tower of invariants is a list with entries max-$\nu^O(X)$, max-$\nu^O(Y)$ resp. max-$\nu^O(A)$ and so on concerning the boundary on this recursion level.

There is a second way to handle this construction. We can consider it as a sequential recursion. This is the natural way to implement recursion. The sequential recursion is described in [35] in Construction 2.16. and in Figure A.2. In this Construction, we first calculate the locus of maximal log-refined order of the input ideal $I_X$ and denote the ideal which describes this locus as $I_Y$. Then we get a $\mathcal{B}$-permissible center by calculating CJS($I_Y$). So this implementation structure is a pure recursion. One part of the output has to be a list of charts of the sequence of blow-up. In each of these charts, the parent chart has to be stored in the path matrix. In the following setting, when we have a recursion in CJS($Y$) and another recursion in CJS($Y'$) it is complicated to append the successor charts to $X$ or $X'$.

Another problem is the labeling of the components of Max-$\nu^O(X')$ because there is no single blow-up but a composing of multiple blow-ups. A further problem is

$$
\begin{array}{llll}
\mathcal{B}' & & & \mathcal{B} \\
Z' & & \to & Z \\
\cup & & & \cup \\
X' & & \to & X \\
\cup & & & \cup \\
Bl_{A''}(Y'') = \emptyset \to & Bl_{B_1}(Y') =: Y'' \to & Bl_B(Y) =: Y' \to & Y = \text{Max-}\nu^O(X) \\
 & \cup & \cup & \cup \\
 & Bl_{B_1}(A') =: A'' \to & Bl_B(A) =: A' \to & A = \text{Max-}\nu^O(Y) \\
 & & \cup & \cup \\
 & & B_1 & B = \text{Max-}\nu^O(A) \text{ regular}
\end{array}
$$

Figure A.2.: CJS recursion as natural recursion

the inheritance of the labels in the different stages.
That is why we reject this approach and choose the first possibility of implementation.

## A.2.2. CJS main method

Now we can formulate the CJS algorithm in a pseudo code way which is the one that underlies the implementation.

**Remark A.2.1** (Algorithm 22)**.** The *input* consists of

- an ideal $I_X = \langle f_1, \ldots, f_r \rangle$, where $f_i \in K[\underline{x}] = K[x_1, \ldots, x_n]$, for $1 \leq i \leq r$ and $K$ being a field of arbitrary charakteristic,
- an integer $mode \in \{1, 2\}$, which determines the method for choosing the transform, i.e., $mode = 1$ means choosing the strict transform and $mode = 2$ means choosing the weak transform,
- a list *boundary* that consists of a sublist of polynomials and a sublist of same size with entries in $\{-2, -1, 1, 2\}$ (see below), which describes the boundary. If it is not given, it would be state by default to *boundary*= *list*() the empty list. This describes the first level of the *tower of boundaries* and
- an ideal describing the ambient space $I_Z \subset I_X$ (default $I_Z = \langle 0 \rangle$).

The *output* of Algorithm 22 is a list of a list *allRings* and a list *EndCharts*, of which the first consists of all charts that are generated and the second consists of all final charts that are generated during the CJS-algorithm. Each chart has the data type *ring* in SINGULAR and in these rings we can find some global (for the ring) variables of the following form:

- an ideal $I_X$, which descrices the (strict/weak) transform of the input ideal,
- an ideal $I_Z \subset I_X$, which describes the ambient space,
- an ideal *desended_*$I_Z$, which describes the corresponding $\tilde{Z}$ in Construction A.1.13. $I_Z \subseteq$ descented_$I_Z \subseteq I_X$ holds,
- a list *boudary* a, which is a list containing three sublists. The first list is the *tower of boundarys*. The second list is a list of polynomials which describes the exceptional divisors of the sequence of blow-up. The third sublist has an encoded information whether a boundary component is old, new or the exceptional divisor of the latest blow-up, i.e.,

$$boundary[j][2][i] = \begin{cases} 1, & boundary[j][1][i] \text{ is a new component; not the latest} \\ -1, & boundary[j][1][i] \text{ is an old component; not the latest} \\ 2, & boundary[j][1][i] \text{ is a new component and the latest} \\ -2, & boundary[j][1][i] \text{ is an old component and the latest} \end{cases},$$

- a list *MaxIX* a, which is a list containing a list of integer vectors (i.e., the *tower of invariants*) and a sublist. The first sublist contains the invariants max-$\nu^O(\cdot)$, which we introduce in Definition 5.6.5. The second sublist contains of irreducible components of Max-$\nu^O(X)$ and their labels. Information about the labeling is described in Construction 5.6.11. The list of invariants has to be stored together with the locus of maximal refined order, because as an integer vector or a list of integer vectors, it could not be stored global in a ring in SINGULAR (everything that should be stored in a ring must contains a variable of the ring), i.e., because of technical reasons,
- a list *tower* which stores the centers of the several levels of the recursion. In each level of the tower we can find the irreducible components of the center with their labels,
- a pathmatrix $path = \begin{pmatrix} 0 & \cdots & k \\ -1 & \cdots & \ell \end{pmatrix}$ such that $k$ describes the number of the predecessor chart (cf. [42]). The successors of the predecessors are labeled from 1 to #successors. The number $\ell$ indicates which of these successors the given chart is. The first column is following the notation of the *resolve.lib* [37] which is part of the SINGULAR computer algebra system. That is why it is internal safed as a module and autoconverted to an integer matrix;
- an ideal *Centerideal*, which describes the image of the center of the parent chart under the canonical blow-up map,

- an ideal *last_ map*, which describes the last map and
- an ideal *total_ map*, which describes the map from the beginning of the sequence of blow-ups until this chart.

Algorithm 22 is the main method of the CJS-Algorithm.

**Algorithm 22** CJS(ideal $I_Z$, ideal $I_X$, ideal descented_IZ, list boundary)

```
1: allRings =init_list();
2: EndCharts = ∅;
3: if (isEndChart()) then return(list(allRings, allRings));
4: Y := list(Max-ν^O(I_X, I_Z)_m, 0)_m;                    ▷ irred. comp. with label 0
5: create_towers(Y);          ▷ create tower of boundary, invariant and center
6: allRings = allRings+CopiesOfCharts(1);                    ▷ blow-up method
7: for (i = 2; i ≤ size(allRings); i++) do
8:     setring(allRings[i]);
9:     invariant = MaxIX[1];                          ▷ invariant before last blow-up
10:    (invariant_new, Y_new) = Maxlocus(tower_of_centers,tower_of_boundary,I_Z, I_X)[1];
11:    descented_IZ = Maxlocus(tower_of_centers,tower_of_boundary,I_Z, I_X)[2];
12:    if (invariantHasImproved() == 1) then
13:        SetAllBoundaryComponentsOld();
14:        invariant = Maxlocus(tower_of_centers,tower_of_boundary,I_Z, I_X)[1][1];
15:        MaxIX[1] = invariant;
16:        Y_new = list((Maxlocus(tower_of_centers,tower_of_boundary,I_Z, I_X)[1][2])_m,0);
17:        for (j = 1; j ≤size(Y); j++) do
18:            MaxIX[2][j] = list(Y_j, 0);          ▷ Y_j is the j-th irred. comp of Y
19:        if (isEndChart() == 1) then
20:            Endcharts.append(allrings[i]);
21:        else
22:            Y := ⋃_m(Y_new)_m                          ▷ union of the irred. comp
23:            create_tower_of_centers(Y)
24:    else
25:        SetAllBoundaryComponentsOld();          ▷ if an invariant improves
26:        if (isEndChart() == 1) then
27:            Endcharts.append(allrings[i]);
28:        else
29:            labeling();
30:            if (size(tower) == 0) then
31:                Y_list = orderYbylabel(Y_new);
32:            else
33:                Y_list = orderYbylabel(tower[last]);
34:            Y = Y_list[k]                  ▷ with k minimal s.t. Y_list[k]≠ ∅
35:            create_tower(Y);
36:        allRings = allRings+CopiesOfCharts(i);
37: return(list(allRings,EndCharts));
```

First, the algorithm checks whether the first chart is already an end chart, so the algorithm is finished. If not, the algorithm computes the first center by calculating Max-$\nu^O(I_X, I_Z)$. The $m$-th irreducible component of Max-$\nu^O(I_X, I_Z)$ is denoted as $(\text{Max-}\nu^O(I_X, I_Z))_m$ and stored together with label 0 as a list in the Singular variable $Y$, for $m \in \mathbb{Z}_{\geq 0}$. Then the *towers* (of center ideals, boundaries, and invariants) are calculated by the *create_tower*-algorithm. For more details see Algorithm A.2.4. Now the blow-up in the center, which is the last entry of the *tower of center*, is calculated, and the successor charts are appended (after the blow-up) to the list of all charts. For more details, see Algorithm A.2.2. The input value of 1 explains that all of these charts are successors of the first chart.

After this, we get in a for-loop which computes the blow-up of each chart and chooses centers by applying rules of the theoretical CJS algorithm (see Construction 5.6.11). This can be seen in the for-loop in lines 7 to 36.

At each run through the for loop, we have to switch the underlying ring to the ring which describes the current chart, in which we perform the calculations. Then we store the invariant from the predecessor chart in order to be able to check the improvement of the invariant. The next step is to calculate the new locus of maximal log-refined order and the new invariants (max-$\nu^O(\cdots)$) at each level of the *tower*. Now, we have to distinguish two cases:

1. The invariant of the first level of recursion (the initial CJS-call) has improved.
2. The invariant of the first level of recursion (the initial CJS-call) has not improved.

In the first case, we mark all of the elements of the boundary of this level to be old (line 13).[1] Then we have to calculate Max-$\nu^O(I_X, I_Z)$ and the invariant max-$\nu^O(I_X, I_Z)$ again since the set of old boundary elements could have changed (lines 14-15).

The next step is to check whether the chart is an end chart. For more details, see Algorithm A.2.6. If it is the case, the chart does not need further blow-ups, and we go through the next run of the for loop. If it is not an end chart, we set $Y$ to the value of $Y\_new$ which coincides with Max-$\nu^O(I_X)$, and we have to calculate the next *tower* of recursive centers and calculate the subsequent blow-up.

In the second case, the singularity has not improved. The invariant of some level of recursion may improve. Then the level's boundary components must be marked as 'old' if the invariant of this level improves, too. That is what the submethod

[1] Only if max-$\nu(I_X)$ improves. The improvement of max-$\nu^O(I_X)$ is not enough.

*SetAllBoundaryComponentsOld* does. It checks the invariants of all levels, and if there is some improvement, the boundary components will be marked to be old. As in the first case, we must check whether we are in an end chart.
After that, we have to label the irreducible components of each recursive centers and the locus of maximal log-refined order Max-$\nu^O(I_X)$. Suppose we are in a recursive CJS call. In that case, the *tower* is not empty, so we have to calculate the next center by taking the union of the minimal label components of the last element of the *tower of centers* or the locus of maximal Max-$\nu^O(I_X)$. We write this information in the variable *Y_list* respectively $Y$.

The end of the calculation in this chart is analogue to the first case. We have to blow-up in center $Y$ and append the successor charts to the list of all charts.

If there are no more charts, we return the list of all charts together with a list of all end charts.

*(End of Remark A.2.1.)*

## A.2.3. Submethods of the CJS implementation

In this subsection, we consider the used submethods in the Singular implementation of the CJS algorithm.

We start with our method which calculates the blow-up and update the data in the successor charts. We will see such a submethod in RESBINOMIAL2 (Algorithm 34) and DET_RESOLVE (Algorithm 56), too.

**Remark A.2.2** (Algorithm 23)**.** Algorithm 23 describes the blow-up in the center, stored in the last element of the list *tower*.
The *input* consists of

- an integer *chartnumber*, which denotes the number of the parent chart, and
- an integer *mode* $\in \{1, 2\}$. If the value of *mode* equals 1, the Algorithm chooses the strict transform; otherwise, if the value equals 2, the Algorithm chooses the weak transform.

The *output* is a list of charts we get by the blow-up. Every chart in this list has the same data structure as the one described in Algorithm A.2.1.

**Algorithm 23** CopiesOfCharts(int chartnumber, int mode) (CJS)

```
1: successor_list = blowUp(⟨0⟩,tower.last);          ▷ ambient space and center
2: Center_this_chart = tower.last;
3: for (j = 1; j ≤size(successor_list); j++) do
4:     def newring = successor_list[j];
5:     map BlowUpMap = r,bM;                          ▷ BlowUpMap : r → newring
6:     I_X = BlowUpMap(I_X) + sT;
7:     if (elimpart(IX)[3] ≠ ⟨0⟩) then                ▷ elimpart() necessary
8:         I_X = elimpart(IX)[1] + elimpart(IX)[3];
9:         change ordering of ring s.t. std(IX) is eliminated by elimpart
10:        map BlowUpElimMap = r,bM ∘ elimpartmap;
11:        elimpartmap(eD);                           ▷ transform the exc div
12:        elimpartmap(sT);
13:    else
14:        map BlowUpElimMap = BlowUpMap;             ▷ elimpart() not used
15:    I_Z = sT;
16:    tower = BlowUpElimMap(tower);
17:    transform(tower,mode);                         ▷ build transform
18:    total_map = BlowUpElimMap(total_map);
19:    last_map = BlowUpElimMap(last_map);
20:    MaxIX = BlowUpElimMap(MaxIX);
21:    transform(MaxIX,mode);                         ▷ build transform
22:    boundary = BlowUpElimMap(boundary);
23:    transform(boundary,mode);                      ▷ build strict transform
24:    descented_I_Z = MaxOrdArithO(I_X, I_Z)[2];
25:    MaxIX[2] = MaxOrdArithO(I_X, I_Z)[1];
26:    path = BlowUpElimMap(path);
27:    path=path,[chartnumber,j];                     ▷ add last column to path
28:    tower = delete(tower,size(tower));             ▷ remove last entry of tower
29:    reset_boundary();                              ▷ reset former latest component
30:    for (i = 1; i ≤size(boundary); i++) do
31:        boundary[i][1][size(boundary[i][1])+1] = eD;          ▷ add exc. div
32:        boundary[i][2] = boundary[i][2] + list(2);   ▷ new and latest divisor
33: return (successor_list);
```

First of all, the map of the blow-up in this center is calculated with the method BlowUp() of the Singular-Library resolve.lib [37]. After that, the data is mapped into the new ring, generated by the *blowUp(·)*-method. The input of this method consists of an ambient space $\langle 0 \rangle$ and the center of the blow-up, which equals the last value of the list *tower*.

The internal method *blowUp(·)* from the Singular library resolve.lib calculates the several charts of the blow-up. Each chart has the data type *ring*, and the variables which are already in these rings are the following:

- $sT$, the strict transform of the blow-up of $\langle 0 \rangle$ in center *tower*.last,
- $eD$, the exceptional divisor of this blow-up and
- $bM$, the map of blow-up.

The parent chart's other elements must be mapped in the new successor chart by applying (bM $\circ$ elimpartmap)$(\cdot)$.

After blowing up the trivial ambient space, we add the ideal of the strict transform of this ambient space to our total transform of $I_X$. Then we have to distinguish two cases:

1. The elimpart-map is not the identity. We have to adapt the ordering of the ring data structure to get the same generators of $I_X$ if we call std$(I_X)$ and elimpart$(I_X)$. That is why we have the if-statement in line 7 and have to change and combine the map of the blowing up with the map of the elimpart-method.

2. The elimpart-map is the identity, so we do not need any changes but the name of the map.

After this (line 15), we only have to map each component without blow-up map. In order to do this, the Algorithm has to know in which ring these components live, so we have this declaration in line 4.
Lines 16 to 22 and 26 are the several mappings of the data structure components. The last lines of the Algorithm belong to little changes on the variables, so the path has to get a new column, the last element of the tower of centers, i.e., the last center has to be deleted, and the new exceptional divisor has to add to the boundary of each level of the tower. So this divisor is the latest divisor and not that one which has the entry 2 resp. $-2$ in the list. So this value has to change to 1 resp. $-1$ and the latest divisor gets the value 2.

The submethod *transform* only calculates the transform concerning the mode of the locus of maximal refined order and the tower. If mode equals 1, then we use the strict transform, and if mode equals 2, then we use the weak transform. The transform of the boundary is independent of the mode. Here we always use the strict transform. This method is treated as a black box here.

It is crucial to notice that this algorithm does not calculate the invariant. So after this Algorithm, in all of the successor charts, the stored invariant is the version of the parent chart. This has to be done later in the main algorithm when $j$ in the for-loop of the main algorithm gets the value of the number of this chart. First, we read the old value, and after this, we can calculate the new value and compare them concerning a lexicographical ordering. We cannot calculate the former value of the invariant after Algorithm 23. *(End of Remark A.2.2.)*

Before blowing up, we have to compute the center. For that calculation we need to label the irreducible components of our locus of maximal invariant.

**Remark A.2.3.** Algorithm 24 describes our labeling method. We label the irreducible components of the locus of maximal log-refined order and the irreducible components of the tower, which contains several centers. The input is

- a list, which encodes the boundary and
- an ideal *last_ Center*, which describes the center of the last blowing up.

The boundary is necessary because it contains information about the last exceptional divisor.
We assume that *last_ Center* $= V(\langle x_1, \ldots, x_\ell \rangle)$ has the label $k \in \mathbb{N}_0$, see Construction 5.6.11.
The output is the list of irreducible components of maximal singularity together with their labels and the *tower of centers*.

**Algorithm 24** labeling(list boundary, ideal last_Center)

```
1: maxlabel = ncols(path);
2: for Y_new ∈ { MaxIX[2], tower[1], ...,tower[size(tower)]} do
3:     for Y_i =: ⟨x_{i_1}, ... x_{i_m}⟩, irred. comp of Y_new contained in the last exc.div do
4:         for 1 ≤ j ≤ m do
5:             if x_{i_j} ∉ {x_1, ..., x_ℓ} then
6:                 label_of_Y_i = maxlabel;                    ▷ not dominating!
7:                 break;
8:         label_of_Y_i = k;
9: return list(MaxIX,tower)
```

Line 1 uses the fact that the number of columns of the pathmatrix equals number of the blow-up. *(End of Remark A.2.3.)*

Now we are able to compute the recursion via the above described *tower*.

**Remark A.2.4.** Algorithm 25 describes the generation of the towers.
The input is an ideal $Y$, which we want to insert in this tower. This $Y$ is a locus where the singularity is maximal or is a locus described by a union of minimal label components of the lowest center.

If the tower is empty, we insert the irreducible components of $Y$ with the label 0 in the list.

As long as the locus of maximal log-refined order differs from the latest component of the tower, the irreducible components of this locus are inserted with the label 0 in the lowest label of the tower.

Since this algorithm only manipulates global variables, there is no particular output.

**Algorithm 25** create_tower Methode(ideal Y)

```
1: if (tower == ∅) then
2:     tower[1] = ⋃_i list(Y_i,0);                ▷ Y_i i-th irred. comp. of Y
3:     boundary[1] = N(X), (1, ..., 1);
4:     invariant = max-ν^O(Y, I_Z);
5: while (Max-ν^O(tower[last]) ≠ tower[last]) do
6:     Y = Max-ν^O(tower[last]);
7:     boundary.append(N(tower[last]), (1, ..., 1));
8:     tower.append(⋃_i list(Y_i,0));
9:     invariant = max-ν^O(Y, I_Z);
10: stop;
```

*(End of Remark A.2.4.)*

The next pseudo code we will discuss shows how to detect the improvements of the invariant.

**Remark A.2.5** (Algorithm 26)**.** The input are two lists. The first list encodes the invariant in the considered chart. The second encodes the invariant in the parent chart. The output is a boolean. The algorithm returns 1, if the new invariant is smaller than the old invariant in the parent chart. Otherwise it returns 0.

**Algorithm 26** invariantHasImproved(intvec invariant, intvec invariant_new) (CJS)

```
1: if (size(invariant_new) > size(invariant)) then
2:     return(1);
3: for (i = 1; i ≤size(invariant); i++) do
4:     if (invariant_new[i] <lex invariant[i]) then
5:         return(1);
6: return(0);
```

In order to check whether the invariant of some level of the tower has improved, we first have to compare the length of both lists. If the list *invariant_new* is longer than the list *invariant*, then the invariant has improved because we have more generators of IX $f_i$, where max-ord$(f_i) = 1$ than before. If both lists' lengths are equal, then we have to compare all entries but the last lexicographically. This belongs to the fact that the last entry is the additional component which encodes $|O|$ in max-$\nu^O$, but we only consider here the improvement of max-$\nu$.

*(End of Remark A.2.5.)*

Finally, the main algorithm has to stop anywhere. So we present now the detection, if the considered chart is an end chart.

**Remark A.2.6** (Algorithm 27)**.** This method has neither input nor output, since the needed information is stored in global (global for each chart) variables.

**Algorithm 27** isEndchart() (CJS Algorithm)

```
1: if (I_Z ⊄ I_X or I_X == 1) then
2:     return (1);
3: if (max-ord(I_X, I_Z) > 1 or max-ν(I_X, I_Z).last > 1) then
4:     return (0);
5: if I_Z == I_X then return (1);
6: return (1);
```

The method isEndchart (Algorithm 27) checks if the maximal order is greater than 1 or if there is more than one old boundary component. In these cases, we are not finished with our CJS algorithm because the scheme is still singular, or we do not know whether we have simple normal crossings. Therefore this method returns in these cases 0 and otherwise 1.

*(End of Remark A.2.6.)*

Experiments on several examples can be found in Section C.

## A.3. New implementation of Blanco's algorithm

We need a seperated method for the calculation of the center of Blanco's algorithm and therefore a modular implementation of it. Unfortunately, the implementation of Blanco and Pfister [12] is only a proof of concept implementation and therefore it not focusses on the structure or a good worst-case complexity. For this reason, the author reimplements the algorithm of Blanco in the resbinomial2.lib. There she uses some methods of resolve.lib [37] as a black box. This is the reason why we assume that the underlying ring $R$ (in SINGULAR) is a field of characteristic 0.

First, we explain the main method, and afterwards, we present the important submethods of this method.

As seen before, every chart is represented by the singular data type RING. In this ring, we have the data of the basic object which is stored in the list BO. We use the notation of Section 5.3, here. The entries of the list are the following:

1. An ideal encoding the ambient space $W$,
2. the considered ideal $J$,
3. an intvec $b$, which contains the order of $J$ and the order of the (recursive) coefficient ideals of $J$,
4. the set $E$ of exceptional divisors,
5. an ideal encoding the map describing the blow-up map,
6. internal data we inherit from resolve.lib and
7. an intvec, where the entries (corresponding to the descent in dimension) point to the exceptional divisors with which we have to intersect.

This data type is already used by resolve.lib and allows us to use methods of resolve.lib on this list BO. Since we do not touch the sixth entry during the whole computation, we do not discuss it, here. See [37] for more information.
Furthermore, there is a global variable which contains the pathmatrix $path = \begin{pmatrix} 0 & \cdots & k \\ -1 & \cdots & \ell \end{pmatrix}$ such that $k$ describes the number of predecessor charts (cf. [42]). The successors of the predecessors are labeled from 1 to #successors. The number $\ell$ indicates which of these successors the given chart is. The first column is following

the notation of the *resolve.lib* [37]. We have already seen this concept of encoding the path in the implementation of the CJS algorithm in Section A.2.

Now we have a look at the main method of the new implementation of Blanco's algorithm.

**Remark A.3.1** (Algorithm 28)**.** The input is a binomial ideal $J$. The output is a list consisting of a the list of all final charts *EndCharts* and the list of all charts *BlowUpTree*.

**Algorithm 28** resbinomial2( ideal $J$)

```
1: def r = basering;
2: list BlowUpTree;
3: BlowUpTree[1] = r;
4: BO = createBO(J);
5: descent_BOlist, b, Center = descent_BO(BO);
6: if (finished(BO)==1) then
7:     Endcharts[1] = BlowUpTree[1];
8: else
9:     BlowUpTree.append(SuccessorCharts(descent_BOlist[1],Center,1));
10: for (i = 2; i ≤ size(BlowUpTree); i++) do
11:     actual_chart = L[i];
12:     setring(actual_chart);
13:     descent_BOlist, b, Center = descent_BO(BO);
14:     if (finished(BO)==1) then
15:         Endcharts.append(BlowUpTree[i]);
16:     else
17:         BlowUpTree.append(SuccessorCharts(descent_BOlist[1],Center,i));
18: return(Endcharts,BlowUpTree);
```

First, the algorithm creates the basic object and calculates the descent in dimension. Afterwards, we can store the center in an additional variable. Then the first chart (if necessary) will be blown up, and the successor charts will be appended to the list of all charts. Then the same procedure will be repeated for every successor chart until there is no more chart.

*(End of Remark A.3.1.)*

The following method we will look at is the DESCENTBO-method. In this method, the descent in dimension is computed.
The called subroutines DELTALIST, which calculates $\Delta^{i-1}(J)$ (Definition 2.4.9) and the maximal $i$ such that $\Delta^{i-1}(J)$ is not empty, and INTERS_E, which computes the intersection with the necessary exceptional divisors, were implemented in the

library resolve.lib.

This method is the most important method for the DET_RESOLVE-algorithm in Section A.5.2. We call this method from DET_RESOLVE, because it calculates the center and updates the whole basic object. So the RESBINOMIAL2-method is only the skeleton and here we have the heart of Blanco's algorithm:

**Remark A.3.2** (Algorithm 29)**.** The input is a list BO which encodes the basic object with the additional data described at the beginning of this Section. The output is a list consisting of a list of the associated coefficient ideals, an intvec $b$ which describes the order at each step of the descent in dimension, and an ideal which encodes the center of the next blow-up following the algorithm of Blanco.

The variable COUNTER stores the current level of the decent in dimension, where the considered basic object starts at level 1. First, we update the ambient space described in Remark 31. Afterwards, we store the basic object as the first element in the list CoeffBOList and calculate via the method DELTALIST of resolve.lib. The size of the returned list corresponds to the order of the considered ideal.
Next, we update the value of BO[7]. This is an internal pointer in which the method INTERS_E of resolve.lib uses to calculate the relevant exceptional divisors. Ultimately, we intersect the center and the ideal of the basic object with the relevant exceptional divisors.
Then we are finished considering this level of descent in dimension and check whether we have to consider the coefficient basic object or whether we are finished with the induction and return the center. If we have to consider the next level via the coefficient basic object of the current basic object, we repeat the same steps. However, before returning the center, we have to check the case in which we are, i.e., if we have to compute the intersection of basic objects or if we are in the monomial case. See Remark 32 for more information.

*(End of Remark A.3.2.)*

We also have to calculate whether the descent in dimension is finished. We use the following method for this calculation: the following DESCENTFINISHED-method.

**Remark A.3.3** (Algorithm 30)**.** The input is an ideal describing the current maximal locus $C_{temp}$, an integer btmp which equals the current maximal order of the current coefficient ideal and a list of type basic object which encodes the current coefficient ideal in the descent in dimension. The output is an integer $\{0, 1, 2, 3, 4\}$ which encodes with which status we end up or returns 0, if we are not finished. The status is 1, if Ctemp is a point, 2 if the coefficient ideal equals $\langle 0 \rangle$, 3 if the

**Algorithm 29** descent_BO(list BO)

counter = 1;
BO = UpdateAmbientSpace(BO);
CoeffBOList[1] := BOtmp:= BO;
intvec $b_{\text{old}}$ = BO[3];
DList = DeltaList(BOtmp); btmp = size(DList);
$b$[1] = btmp;
Ctemp = DList[btmp]; ▷ $\Delta^{\text{btmp}-1}(J)$
orderdecreased = 0;
**if** ($b$[1] == $b_{\text{old}}$) **then**
    BO7new[1] = BO[7];
**if** ($b$[1] < $b_{\text{old}}$) **then**
    BO7new[1] = size(BO[4]);
    orderdecreased = 1;
BOtmp[7] = BO7new;
list E = inters_E(BOtmp); ▷ $E$[1] are the relevant exc divs
CoeffBOList[1][2] = std($E[1]^{b[1]}$+ CoeffBOList[1][2]); ▷ Companion ideal
Ctemp = std(Dlist[btmp] + E[1]);
**while** (descent_finished(Ctemp, BOtmp) == 0) **do**
    BOtmp = UpdateAmbientSpace(Coeff(BOtmp,b[size(b)])); ▷ coefficient BO
    CoeffBOList[size(CoeffBOList)+1] = BOtmp;
    casenumber = DetectCase(BOtmp,DList);
    **if** (casenumber == 2) **then**
        BO[1..4] = (BO[1],($Icontrol^{b_{\text{counter}}}$+BO[2]$^{b_{\text{counter}}}$), $E$) ▷ intersection of BOs
    DList = DeltaList(BOtmp), btmp = size(DList);
    $b$[size($b$)+1] = btmp;
    **if** (orderdecreased == 0 and (size($b_{\text{old}}$) ≥ counter)) **then**
        **if** ($b$[counter] == $b_{\text{old}}$[counter]) **then**
            BO7new[counter] = BO7old[counter];
        **if** ($b$[counter] < $b_{\text{old}}$[counter]) **then**
            BO7new[counter] = size(BOtmp[4]);
            orderdecreased = 1;
    **else**
        BO7new[counter] = 0;
    BOtmp[7][1] = BO7new[counter];
    **if** (casenumber == 1 or casenumber == 2) **then**
        $E$ = inters_E(BOtmp); ▷ intersection with the exc divs
        CoeffBOList[counter][2] = std($E[1]^{b[1]}$+ CoeffBOList[counter][2]);
        Ctemp = std(Dlist[btmp] + $E$[1]);
    **else**
        BOtmp[2] = Icontrol;
        BOcontrol[3] = CoeffBOList[counter−1][3];
        center = MonomialCase(BOcontrol,DList);
    counter++;
**for** ($j = 1; j \leq$ size(BO7new); $j++$) **do** ▷ set all BO[7] entries
    CoeffBOList[$j$][7] = intvec(BO7new[$j$..size(BO7new)]);
return(CoeffBOList,Ctemp);

coefficient ideal equals $\langle 1 \rangle$ and 4 if Ctemp is smooth and simple normal crossing with the exceptional divisors, so it is a suitable center

**Algorithm 30** descent_finished(ideal $C_{temp}$, int btmp, list CoeffBO)

```
1: if isSmooth(C_temp) and normalCross(C_temp,CoeffBO[4]) == 1 then
2:     return(1);
3: else if CoeffBO[2] == ⟨0⟩ then
4:     return(2);
5: else if CoeffBO[2] == ⟨1⟩ then
6:     return(3);
7: else
8:     C_temp = √C_temp
9:     if isSmooth(C_temp) and normalCross(C_temp,CoeffBO[4]) == 1 then
10:        return(4);
11: return(0);
```

Case 4 abbreviates the vertical induction so we can take the radical as the center if it is smooth and has normal crossings with the exceptional divisors in the binomial case. *(End of Remark A.3.3.)*

We have already seen the main idea of the following method in the implementation of the refined order (see Remark A.1.12). If we have generators of the form, $1 - \underline{x}^{\underline{A}}$, we add them to the set of generators of the ambient space.

**Remark A.3.4** (Algorithm 31)**.** The input is a list describing the current (coefficient) basic object of the descent in dimension. The algorithm detects generators of BO[2] of the form $1 - \underline{x}^{\underline{A}}$ and adds it to the generators of the ambient space BO[1].

**Algorithm 31** UpdateAmbientSpace(list BO)

```
1: DList = DeltaList(BO);
2: for (j = 1; j ≤ size(BO[2]);j++) do
3:     if size(DList) == 1 then
4:         tmpBO[2] = BO[2][j];
5:         DListtmp = DeltaList(tmpBO[2]);     ▷ calculate order of the j-th entry
6:         if (size(DListtmp) == 1) then
7:             BO[1] = BO[1] + tmpBO[2];
8:             BO[2] = delete(BO[2],j);
9:             Dlist = DeltaList(BO);
10: return(BO);
```

*(End of Remark A.3.4.)*

Now, we have a distuingish the different three cases, which we detect in the following algorithm.

**Remark A.3.5** (Algorithm 32)**.** The input is the current (coefficient) basic object on level $i$ of the descent in dimension and the deltalist of the (coefficient) basic object on level $i-1$.
The output is an integer describing the current case. We return 1, if the order of the strict transform (of the coefficient ideal of level $i$) is greater or equal to $b!$, where $b$ is the order of the coefficient ideal of level $i-1$. In this case, we do not need to calculate the intersection of basic objects because it has no effort. So we can abbreviate the process.
In the second case, the order of the strict transform (of the coefficient ideal of level $i$) is between 1 and $b!$, where $b$ is again the order of the coefficient ideal of level $i-1$. And we return 3, if we are in the monomial case. That is the case, if the order of the strict transform equals 1.

**Algorithm 32** DetectCase(list BO, list DList)

```
1: int b = size(DeltaList(BO));
2: if (b ≥ size(DList)!) then
3:     return(1);
4: else if (1 < b < size(DList)!) then
5:     return(2);
6: return(3);
```

*(End of Remark A.3.5.)*

We discuss now the monomial case of the algorithm of Bravo, Encinas and Villamayor (see end of Section 5.3), which is our case number three.

**Remark A.3.6** (Algorithm 33)**.** The input is the current (coefficient) basic object on level $i$ of the descent in dimension and the deltalist of the (coefficient) basic object on level $i-1$. The output is the center of the next blow-up following the monomial case.

**Algorithm 33** MonomialCase(list BO, list DList)

```
int b = BO[3][1];
J = BO[2];
list Explist = leadexp(J[1]);
for (j = 1; j ≤ n; j++) do
    duplicate Explist n-times in list tmplist
    delete a single superfluent entry s.t. ∑_{i∈I} Explist_i ≥ b holds
    add list without dulicates to tmplist
delete sublists with not minimal codimension
only keep sublists with maximal sum of entries
return the lexicographical minimal element
```

*(End of Remark A.3.6.)*

We use the following example to illustrate the implementation of the monomial case. The example is inspired by [16, Example 20.4].

**Example A.3.7.** Let $(W, (J, 9), E = \{H_1, H_2, H_3, H_4\})$ be a basic object, where $W$ denotes the real analytic space $\mathbb{R}^4$, $H_i$, $1 \leq i \leq 4$ are the coordinate hyperplanes and

$$J = \mathcal{I}(H_1)^6 \mathcal{I}(H_2)^4 \mathcal{I}(H_3)^2 \mathcal{I}(H_4)^2.$$

The intersection $H_1 \cap H_2 \cap H_3 \cap H_4$ is a closed point. The singular locus of the basic object is

$$\mathrm{Sing}(J, 9) = (H_1 \cap H_2) \cup (H_1 \cap H_3 \cap H_4),$$

since $6 + 4 = 6 + 2 + 2 \geq 9$.
The function $h$ is given by

$$h(\xi) = \begin{cases} (-2, \frac{10}{9}, (1, 2, 0, 0)), & \text{if } \xi \in H_1 \cap H_2 \\ (-3, \frac{10}{9}, (1, 3, 4, 0)), & \text{if } \xi \in (H_1 \cap H_3 \cap H_4) \backslash (H_1 \cap H_2) \end{cases}.$$

The maximum value is $\max h = (-2, \frac{10}{9}, (1, 2, 0, 0))$ and therefore $\mathrm{Max}\, h = H_1 \cap H_2$.

Our Algorithm 33 first takes the leadexponents, so we have a list $(6, 4, 2, 2)$ and duplicate it 4 times. So the list tmplist consists of 4 times the list $(6, 4, 2, 2)$, i.e., tmplist[1] $= (6, 4, 2, 2)$, tmplist[2] $= (6, 4, 2, 2)$ and so on.
Afterwards, we delete elements such that the sum of the remaining elements have a sum greater or equal 9. So $(0, 4, 2, 2)$ is not a possibility since $4+2+2 = 8 < 9$. The next possibility is $(6, 0, 2, 2)$. The sum of the elements is $10 > 9$. So tmplist[5] $= (6, 0, 2, 2)$. The remaining possibilities yield tmplist[6] $= (6, 4, 0, 2)$ and tmplist[7] $=$

$(6, 4, 2, 0)$. Repeating this procedure, we get a further entry tmplist[8] $= (6, 4, 0, 0)$.

So we detect in the for loop the singular locus

$$\mathrm{Sing}(J, 9) = (H_1 \cap H_2) \cup (H_1 \cap H_3 \cap H_4).$$

Since $6 + 4 = 6 + 2 + 2 = 10$, the second component of $h(\xi)$ equals $\frac{10}{9}$, for $\xi \in \mathrm{Sing}(J, 9)$.
Finally, we detect the lexicographical minimal element (with respect to the indices) and return $(H_1 \cap H_2)$.

In the main method RESBINOMIAL2 the blow-up is fulfilled by the SUCCESSORCHARTS-method. In there we perform the blow-up and update the data of the basic object and the path of the successor charts. We have seen this type of method already in the implementation of the CJS algorithm in Algorithm 23.

**Remark A.3.8** (Algorithm 34)**.** The input consists of

- a list describing the basic object BO,
- the ideal describing the center and
- an integer which denotes the total number of the charts.

The output is a list of successor charts. This algorithm creates and fills the successor

**Algorithm 34** SuccessorCharts(list BO, ideal cent, int chartnumber)

```
 1: def r = basering;
 2: successor_list = blowUpBO(BO,cent,2);
 3: Numb_sucessors = size(successor_list);
 4: for (j = 1; j ≤ Numb_sucessors; j++) do
 5:     def newring = successor_list[j];
 6:     setring(newring);
 7:     n = nvars(basering);
 8:     ideal bM=BO[5];                          ▷ copied from resolve.lib
 9:     map BlowUpMap = r,bM;
10:     ideal last_map = bM;
11:     ideal total_map = BlowUpMap(total_map);
12:     def path = imap(r,path);
13:     path=path,[chartnumber,j];
14: setring(r);
15: return(successor_list);
```

charts with the necessary data. *(End of Remark A.3.8.)*

## A.4. Implementational aspects of Hu's algorithm

In this section we present our implementation of Hu's algorithm. First, we discuss how to implement the algorithm in general. Later on, we discuss how the implementation looks like, if we want to plug in our main implementation DET_RESOLVE in Section A.5.2.

**Remark A.4.1.** The *input* is

- an ideal $X$ describing the ambient space and
- $\{f_1, \ldots, f_m\}$ a list of binomials, with $f_j = f_j = 1 - \rho_j x^{A_j}, 1 \leq j \leq m$.

The *output* of Algorithm 35 is a list of total Charts and final charts for the determinantal algorithm.

**Algorithm 35** Hu(ideal $X$, list $\{f_1, \ldots, f_m\}$)

**INPUT:** $X$,$\{f_1, \ldots, f_m\}$, with $f_j = 1 - \rho_j \underline{x}^{\underline{A}_j}, 1 \leq j \leq m$

1: $Y := V(\prod_{j=1}^{m} f_j)$
2: $X^0 := X \backslash Y$;
3: $I = \mathfrak{P}\{1, \ldots, m\}$
4: $L = \text{list}(\text{list}())$;
5: **for** $i \in I$ **do**
6: $\quad D_i := \bigcap_{j \in i} V(f_j)$;
7: $\quad$ L[$n-$size($i$)].append($D_i$); ▷ The $\ell$-th element of $L$ is a list of $D_i$ of rank $\ell$.
8: **for** $(\ell = n,\ \ell \geq 1,\ \ell --)$ **do**
9: $\quad$ **for** $(k = 1,\ k \leq \text{size}(L[\ell]),\ k++)$ **do**
10: $\quad\quad$ **if** $(L[\ell][k] \neq \emptyset)$ **then**
11: $\quad\quad\quad X = B\ell_{L[\ell][k]}(X)$; ▷ Blow-up with center $L[\ell][k]$
12: $\quad\quad\quad L = B\ell_{L[\ell][k]}(L)$; ▷ short notation for blow-up every element in $L$
13: **return**

*(End of Remark A.4.1.)*

We will not focus on this general approach of implementing Hu's algorithm here since we only need the choice of the center of Hu's algorithm. Here, we will only calculate the intersection lattice and the covering for the normal crossing process.

Since our main goal is to resolve determinantal singularities of at most binomial type, we optimized the implementation of Hu's algorithm for the use as a black box in the determinantal resolution algorithm, i.e., we will not calculate the whole intersection lattice but the elements of this which we will need until our entries of the matrix become normal crossings. Furthermore, Hu's algorithm provides the

covering we need for the monomialization process. So we calculate this covering here and do not need to calculate it later a second time.

**Remark A.4.2** (Algorithm 36)**.** The *input* is a matrix $M$ which has principalized entries which do not need to have normal crossings. The *output* of Algorithm 36 is:

- a list IntersectionLattice, which contains the elements in the intersection lattice, where we have to blow-up and
- a list CoveringAndComponentList, that contains the relative extremal elements of the intersection lattice together with a list of elements in Entrylist which does not generate the covering. More information can be found in Section A.5.2.

**Algorithm 36** HuAlgorithm(matrix $M$)

```
1: Entrylist = removeDuplicates(ListForHu(M));
2: list IntersectionLattice = HuCalc(Entrylist);
3: list coveringlist = IntersectionLattice[2];
4: list CoveringAndComponentList;
5: if (size(IntersectionLattice[1][1]) == 0) then      ▷ Only calculate the covering
6:     for (i = 1; i ≤size(Entrylist);i + +) do
7:         if (size(Entrylist[i]) == 1) then
8:             CoveringAndComponentList.append(list(list(Entrylist[i]),{Entrylist[j]|
   i ≠ i}));
9: else
10:    for (i = 1; i ≤size(coveringlist);i + +) do
11:        CoveringAndComponenList[i][1] = coveringlist(i);
12:        for (j = 1; j ≤size(Entrylist);j + +) do
13:            if (MemberOfList(Entrylist[j], coveringlist[i]) == 0) then
14:                CoveringAndComponentList[i][2].append(list(Entrylist[j]));
15:                CoveringAndComponentList[2].append({Entrylist[j]| i ≠ i})
16: return(IntersectionLattice[1],CoveringAndComponentList);
```

*(End of Remark A.4.2.)*

The method ListForHu converts the principalized entries of the matrix into a list of the irreducible components without duplicates and 1 entries.

**Remark A.4.3** (Algorithm 37)**.** The *input* is a matrix $M$ which has principalized entries which do not need to have normal crossings. The *output* of Algorithm 37 is a list Entrylist, which contains the converted list.

**Algorithm 37** ListForHu(matrix $M$)

```
1: Entrylist = list();
2: for (i = 1; i ≤ #rows(M); i++) do
3:     for (j = 1; j ≤ #cols(M); j++) do
4:         if (M[i, j] ≠ 1) then
5:             Entrylist.append(minAssGTZ(M[i, j]));
6: deleteDuplicates(Entrylist);
7: return(Entrylist);
```

In line 5, the method minAssGTZ calculates the minimal associated prime ideals of proper ideal. *(End of Remark A.4.3.)*

**Remark A.4.4** (Algorithm 38)**.** The *input* is the list Entrylist, calculated in Algorithm 37. The *output* of Algorithm 38 is a list containing:

- a list centerlist, which contains the elements of the intersection lattice where we have to blow-up
- a list covering list, which contains the relative extremal components of the intersection lattice.

**Algorithm 38** HuCalc(list Entrylist)

```
1: IntersectionLattice = CalculateIntersectionLattice(Entrylist)
2: for (i = 1; i ≤size(IntersectionLattice[1]);i++) do
3:     calculate centerlist[i]          ▷ an ideal generated by the elements of
   IntersectionLattice[1][i];
4: coveringlist = IntersectionLattice[2];
5: return(centerlist,coveringlist);
```

*(End of Remark A.4.4.)*

**Remark A.4.5** (Algorithm 39)**.** The *input* is the list Entrylist, calculated in Algorithm 37. The *output* of Algorithm 39 is a list containing:

- a list centerlist, which contains the elements of the intersection lattice where we have to blow-up
- a list covering list, which contains the relative extremal components of the intersection lattice.

**Algorithm 39** CalculateIntersectionLattice(list Entrylist)

```
1: list IntersectionLattice = list();
2: emptyOrOrigin = 0, i = 2;
3: sizeL = size(Entrylist);
4: Listideal = ideal(Entrylist);
5: n = Codimension(Listideal);
6: while (emptyOrOrigin == 0) do
7:     emptyOrOrigin = 0, NotEmpty = 0;
8:     SubsetList = subsets(sizeL,i);          ▷ list of all sizeL-element subsets of
   {1,...,n}
9:     for (j = 1; j ≤ size(SubsetList);j++) do
10:        IntersectionLattice[i].append(list());
11:        for (k = 1; k ≤ i; k++) do
12:            IntersectionLattice[i][last].append(Entrylist[SubsetList[j][k]]);
13:            if (⟨ IntersectionLattice[i][last][j] | j ≤ size
   IntersectionLattice[i][last]⟩) then
14:                delete(IntersectionLattice[i][last]);
15:            if (IntersectionLattice[i] = ∅ or i == sizeL) then
16:                emptyOrOrigin = 1;
17: list coveringlist = list();
18: for (i = min{n,size(IntersectionLattice)}; i > 1; i − −) do
19:     for (j = 1; j ≤ size(IntersectionLattice[i];j++)) do
20:         contained = 0;
21:         for (k = 1; k ≤ size(coveringlist);k++) do
22:             if (isSubList(IntersectionLattice[i][j],coveringlist[k])) then
23:                 contained = 1;
24:             if (contained == 0) then
25:                 coveringlist.append(IntersectionLattice[i][j]);
26: for (i = 1; i ≤ n; i++) do
27:     for (j = 1; j ≤ size(IntersectionLattice[i]); j++) do
28:         if (allPairwiseTransversal(IntersectionLattice[i][j])) then
29:             delete(IntersectionLattice[i][j]);
30: delete emptylists in IntersectionLattice
31: return(IntersectionLattice,coveringlist);
```

The while loop calculates the intersection lattice. Afterwards, the covering is calculated as the relative extremal components, i.e., the codimension $n$ components, or if this component is empty, then we have to take the elements with lower codimension that are not empty lying above in a full (i.e., with empty components) intersection lattice.

Finally, we delete the superfluent components of the intersection lattice, namely if the codimension is $n$ or less and all components are pairwise transversal.

We use the notation 'delete(list[entry])' which is shorthand for the call of 'delete(list,

entry)'. *(End of Remark A.4.5.)*

# A.5. Implementational aspects of the determinantial resolution

We have discussed the methods that help us to present this thesis's main algorithms. In the first subsection, we discuss the implementational aspects of the generic cases. The theoretical background we gave in Chapter 6. Afterwards, we generalize the skeleton of the generic case for the determinantal resolution of Chapter 8.

## A.5.1. Implementational aspects of the generic case

In this subsection, we discuss an implementation of the generic, the generic symmetric, and the generic skew-symmetric case written in the computer algebra system SINGULAR [30].

Throughout this section, we assume that the underlying ring $R$ (because of the constraints of the computer algebra system SINGULAR and the fact that only fields and $\mathbb{Z}$ are implemented) is a field or $\mathbb{Z}$. Furthermore, we assume that $\mathrm{char}(R) \neq 2$ in the skewsymmetric case.

The main algorithm is for all cases the same. To distinguish the several cases, we add input parameter mode. So we do not need to duplicate the same code. So we get better readability.
For better usability, we add different methods *generic_resolve, symmetric_generic_resolve* and skewsymmetric_generic_resolve which only call the main algorithm with the suitable parameters.

**Remark A.5.1** (Algorithm 40)**.** The *input* is:

- a string *mode* $\in$ { "skew", "sym", "gen"} which encodes the desired algorithm,
- an integer $m$ which describes the number of rows,
- optional: an integer $n$ which describes the number of columns, if it is not given, then we assume the generic $m \times m$ matrix. Only availiable for the pure generic case and
- optional: an integer $r \in \{1, \ldots, \mathrm{rank}(M)\}$, which gives the information about what $I_{r,m,n}$ the algorithm has to resolve. Default: $r = \min\{m, n\}$.

The *output* of Algorithm 40 is a list of a list of all final charts *EndCharts* and a list of all charts *allRings*.

**Algorithm 40** generic_resolve(string *mode*, int $m$[, int $n$, int $r$])

```
 1: if (not defined(n)) then
 2:     n = m;
 3: matrix M = generate_matrix(mode,m,n); ▷ M = M_{m,n}, M = A_m or M = B_m
 4: list allRings =init_list(M,r);
 5: list EndCharts = list();
 6: if (finished(M)) then
 7:     return(list(allRings,allRings));
 8: allRings.append(blowup(M,minor(M,1)));
 9: for (i = 2; i ≤ size(allRings); i++) do
10:     setring(allRings[i]);
11:     M = gaussian_step(mode, M);
12:     M = submatrix(M, mode);
13:     r = min{#rows of M, #columns of M};
14:     if (finished(M)) then
15:         EndCharts.append(actual_ring);
16:         continue;
17:     else
18:         M = transformationOfVariables(mode,M);
19:         allRings.append(blowup(M,std(radical(minor(M,1)))));
20: return(list(EndCharts,allRings));
```

First, we have to initialize the ring structure with all its information. So every chart in the list *allRings* provides the information about the strict transform of $M$ (by abuse of notation, also called $M$), the dimension of the size of the relevant minors ($\text{rank}(M) - r$) and the linear transformation of the variables.
We are finished if the considered minor is regular, i.e., $r == 1$ in the generic case and $r == 2$ in the generic skew-symmetric case. If we are finished, there is nothing to do, and the algorithm returns a list with a list of the actual ring twice since it is the list of all charts during this resolution process, and it is also the list of all end charts.
Otherwise, we have to blow-up the origin, and in the next steps, we have to consider the successor charts, which are now stored in the list of all charts.
During the for-loop (lines 9 to 19), we first have to change our setting to the setting of the following unconsidered chart (*actual_ring*). In this chart, we first have to do the gaussian_step and reduce the value of $r$ by one or two depending on the mode and the considered chart. Now, we are in the same setting as we have seen in the first chart. If in the considered chart, the ideal generated by the interesting

minors is regular, this chart is an end chart, so we add it to the list of end charts. Otherwise, we blow-up in origin (after suitable linear transformation).
We repeated this until every chart in the list of all charts was considered. Then we return the list of all charts together with the list of end charts.

*(End of Remark A.5.1.)*

The only differences between all of the generic cases are

- the construction of the initial matrix $M$,
- the gaussian steps, we have discussed in Construction 6.1.2 and in the proofs of Main Theorem 4 and Main Theorem 5,
- the computation of the submatrix and
- the transformation of variables.

Since the construction of the initial matrix and the calculation of the right submatrix are obvious, we only have to handle the different Gaussian steps and transformation of variables implementations.

We want to take advantage of the structure to reduce complexity, here. So we do not have to calculate the blow-up via ideals, but we can manipulate the entries in every single chart by setting the 1 entry at the correct position. So we can consider less complex data structures here.

We now look at the implementations of the several gaussian steps and start with the generic case.

**Remark A.5.2** (Algorithm 41)**.** The *input* is:

- a string *mode* = "gen" which encodes the desired algorithm and
- a matrix $M$.

The *output* of Algorithm 41 is the corresponding matrix $M$ after the gaussian step.

The algorithm first finds the position of the 1 entry in $M$. This position is denoted as $(j, k)$. Afterwards, we generalize the Gaussian step discussed in 6.2 to all charts. We only need to perform the row operation since we have already seen that the column operations only set the entries of the $\ell$-th row to the value of 0, so we can reduce complexity here and we directly set these entries. *(End of Remark A.5.2.)*

The next more difficult mode is the generic skew-symmetric case since we need to perform two row resp. column operations.

**Algorithm 41** gaussian_step(string *mode="gen"*, matrix $M$)

```
1: (j, k) = PositionOfOneEntry(M);
2: for (i = 1; i ≤ #rows(M); i++) do
3:     for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
4:         if (i ≠ j) then
5:             M[i, ℓ] = M[i, ℓ] − M[i, k] · M[j, ℓ];
6: for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
7:     if (ℓ ≠ k) then
8:         M[j, ℓ] = 0;
9: return(M);
```

**Remark A.5.3** (Algorithm 42)**.** The *input* is:

- a string *mode* = "skew" which encodes the desired algorithm and
- a matrix $M$.

The *output* of Algorithm 42 is the corresponding matrix $M$ after the gaussian step.

**Algorithm 42** gaussian_step(string *mode="skew"*, matrix $M$)

```
1: (j, k) = PositionOfOneEntry(M);
2: for (i = 1; i ≤ #rows(M); i++) do
3:     for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
4:         if i ≠ j and i ≠ k then
5:             M[i, ℓ] = M[i, ℓ] − M[i, k] · M[j, ℓ];
6: for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
7:     if (ℓ ≠ k) then
8:         M[j, ℓ] = 0;
9: for (i = 1; i ≤ #rows(M); i++) do
10:    for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
11:        if (ℓ ≠ j and ℓ ≠ k) then
12:            M[i, ℓ] = M[i, ℓ] − M[i, j] · M[k, ℓ];
13: for (i = 1; i ≤ #rows(M); i++) do
14:    if (i ≠ k) then
15:        M[i, k] = 0;
16: return(M);
```

The algorithm first detects the position of the 1 entry in $M$. This position is denoted as $(j, k)$. Afterwards, we generalize the Gaussian step discussed in Section 6.2 to all charts. We only need to perform the first row operation and the second column operation since we have already seen that the first column operation and the second row operation only set the corresponding entries to the value of 0.

Again, we can reduce complexity by directly setting these entries to 0. *(End of Remark A.5.3.)*

The last and the most complicated mode is the symmetric mode since we have to distinguish 2 cases depending on the position of the 1 entry in the matrix $M$, namely the diagonal and the non-diagonal charts.

**Remark A.5.4** (Algorithm 43)**.** The *input* is:

- a string *mode* = "sym" which encodes the desired algorithm and
- a matrix $M$.

The *output* of Algorithm 43 is the corresponding matrix $M$ after the gaussian step.

After storing the position of the 1-entry in $(j, k)$, the algorithm checks if it is in a $X_{j,j}$-chart or a $X_{j,k}$-chart with $j \neq k$.

We have to perform only a single row operation in the $X_{j,j}$-charts. In the $X_{j,k}$-charts, we first have to perform the first row operation and set the corresponding entries afterwards to 0. The next step is to multiply and divide entries with $\varepsilon$. Then we have to perform the second column operation, and finally, we need to set the corresponding entries to the value of 0 instead of performing a second row operation. *(End of Remark A.5.4.)*

We now discuss the transformation of the variables in the different modes. The difference between the modes is the concrete substitution because we have to preserve the (skew-)symmetry in the (skew-)symmetric case and the generic structure in the generic case.

In the following remark, we discuss the generic skew-symmetric transformation of variables in the generic skew-symmetric case:

**Remark A.5.5** (Algorithm 44)**.** The *input* is:

- a string *mode* = "skew" which encodes the desired algorithm and
- a skewsymmetric generic matrix $M$.

The *output* of Algorithm 44 is the transformed matrix $M$.

**Algorithm 43** gaussian_step(string *mode*=*"sym"*, matrix $M$)

```
1: (j, k) := PositionOfOneEntry(M);
2: if (j = k) then                                          ▷ X_{j,j}-chart
3:     for (i = 1; i ≤ #rows(M); i++) do
4:         for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
5:             if (i ≠ j) then
6:                 M[i, ℓ] = M[i, ℓ] − M[i, j] · M[j, ℓ];
7:     for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
8:         if (ℓ ≠ j) then
9:             M[j, ℓ] = 0;
10: else                                                    ▷ X_{j,k}-chart, j ≠ k
11:     for (i = 1; i ≤ #rows(M); i++) do
12:         for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
13:             if (i ≠ k) then
14:                 M[i, ℓ] = M[i, ℓ] − M[i, j] · M[k, ℓ];
15:     for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
16:         if (ℓ ≠ j) then
17:             M[j, ℓ] = 0;
18:     ε := 1 − M[j, j] · M[k, k];
19:     M[j, k] = 1;                                        ▷ Divide M[j, k] by ε
20:     for (i = 1; i ≤ #rows(M); i++) do
21:         for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
22:             if (i ≠ j and i ≠ k) then
23:                 M[i, ℓ] = M[i, ℓ] · ε;
24:     for (i = 1; i ≤ #rows(M); i++) do
25:         for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
26:             if (i ≠ j) then
27:                 M[i, ℓ] = M[i, ℓ] − M[i, k] · M[j, ℓ];
28:     for (ℓ = 1; ℓ ≤ #cols(M); ℓ++) do
29:         if (ℓ ≠ k) then
30:             M[j, ℓ] = 0;
31: return(M);
```

**Algorithm 44** transformationOfVariables(string *mode="skew"*, matrix $M$)

```
1: var_counter = 1;
2: for (i = 1; i ≤ #rows(M); i++) do
3:     for (j = i + 1; j ≤ #cols(M); j++) do
4:         M[i, j] = var(var_counter);
5:         M[j, i] = − var(var_counter);
6:         var_counter++;
7: return(M);
```

So the algorithm only takes the entry $(i, j)$ of $M$ and substitute it with the variable with number *var_ counter*. Afterwards, we preserve the skew symmetry and substitute the entry $(j, i)$ of $M$ by $-1$ times the same variable.
Then we increase the value of *var_ counter* and repeat the procedure for $(i, j)$ is the entries of $M$ in the upper triangle of $M$.

*(End of Remark A.5.5.)*

The other modes are analogous. The symmetric case does not multiply the $(j, i)$-entry with $-1$ and also considers the diagonal entries.

**Remark A.5.6** (Algorithm 45)**.** The *input* is:

- a string *mode* = "sym" which encodes the desired algorithm and
- a symmetric generic matrix $M$.

The *output* of Algorithm 45 is the transformed matrix $M$.

**Algorithm 45** transformationOfVariables(string *mode="sym"*, matrix $M$)

```
1: var_counter = 1;
2: for (i = 1; i ≤ #rows(M); i++) do
3:     for (j = i; j ≤ #cols(M); j++) do
4:         M[i, j] = var(var_counter);
5:         M[j, i] = var(var_counter);
6:         var_counter++;
7: return(M);
```

So the algorithm only takes the entry $(i, j)$ of $M$ and substitute it by the variable with number *var_ counter*. We preserve the symmetry and substitute the entry $(j, i)$ of $M$ by the same variable.
Then we increase the value of *var_ counter* and repeat the procedure for $(i, j)$ is the entries of $M$ in the upper triangle of $M$. *(End of Remark A.5.6.)*

In the generic case, the two for loops iterate over all entries $(i, j)$ and only set $M[i, j]$ to the variable with number *var_ counter*.

**Remark A.5.7** (Algorithm 46)**.** The *input* is:

- a string *mode* = "gen" which encodes the desired algorithm and
- a generic matrix $M$.

The *output* of Algorithm 46 is the transformed matrix $M$.

**Algorithm 46** transformationOfVariables(string *mode="gen"*, matrix $M$)

```
1: var_counter = 1;
2: for (i = 1; i ≤ #rows(M); i++) do
3:     for (j = 1; j ≤ #cols(M); j++) do
4:         M[i, j] = var(var_counter);
5:         var_counter++;
6: return(M);
```

The algorithm only takes the entry $(i, j)$ of $M$ and subtitute it by the variable with number *var_ counter*. *(End of Remark A.5.7.)*

## A.5.2. Implementational aspects of resolution of determinantal singularities of at most binomial type

In this subsection, we will generalize the generic skeleton for the determinantal resolution algorithm. Since we will use methods of resolve.lib [37] which is an implementation of Villamayors's algorithm, that resolves singularities in characteristic zero, we restrict the implementation to the case of determinantal singularities of at most binomial type in characteristic zero. We have seen in Section 8 that we can substitute several methods by other algorithms, so the implementation is also modular and we are able to substitute several algorithms in the future.

First, we discuss the considered data types and data sets.

The list DATALIST consists of the main data of the ring. The order of the entries is influenced by the order of the elements of the list BO in resolve.lib. We can find the following data in the variable DATALIST:

1. an ideal defining the ambient space,
2. the considered matrix in the actual chart,

3. a list which consists of the STATE and the size of the minors which generate the considered ideal,

4. a list of exceptional divisors and

5. an ideal defining the map ob blow-up.

Further variables living in the underlying ring are the variables HUList and BO. Whereas the first consists of the intersection lattice resulting from Hu's algorithm, the second is the basic object of resbinomial2.lib. They only exist when we are in the process of resolution of singularities binomial ideals resp. arrangements of hypersurfaces.

As we have seen in Chapter 8, we have to divide our Algorithm into five parts, namely the determinantal binomial case, the usage of the Algorithm of Hu, the linear transformation for getting monomial entries in the matrix, the determinantal monomial case and the usage of the Gaussian Algorithm.

So the idea is to implement the main Algorithm as a state machine, i.e., we have to check the state and depending on this state we perform the calculation. After the calculation in a state is finished we have to recheck the current state. In particular, we have to notice if we are in the middle of a calculation in a state or not.

This is realized with the additional data BO and HUList and that they are only defined if we are in the middle of a calculation in resbinomial2 or Hu's algorithm.

**Remark A.5.8** (Algorithm 47)**.** The *input* is:

- a matrix $M \in K[\underline{x}]^{n\times m}$, where every entry of $M$ is at most binomial, $K$ is a field of characteristic zero and

- an integer sizeOfMinor $\in \{1,\ldots,\text{rank}(M)\}$, which gives the information about what $I_{\text{sizeOfMinor},n,m}$ the algorithm has to resolve.

The *output* of Algorithm 47 is a list of a list *allRings* and a list *EndCharts*. Algorithm 47 is the main method.

**Algorithm 47** determinantal_resolve(matrix $M$,int sizeOfMinor)

```
1: list datalist = createDatalist(M,sizeOfMinor);
2: list allRings; allRings[1] = basering;
3: list EndCharts;
4: string state = datalist[3][1];
5: for (i = 1; i ≤ size(allRings); i++) do
6:     actual_ring = allRings[i];
7:     if (defined(BO) or defined(HuList)) then
8:         if (defined(BO)) then
9:             ideal center = calculate_center();
10:            if (center == ⟨1⟩ or center ==BO[1] or center ==BO[2]) then
11:                kill(center); kill(BO);                ▷ Endchart in resbinomial2
12:                datalist[3][1] = state = calculate_state(M,datalist[4]);
13:        else
14:            if HuList[1] ≠ ∅ then
15:                ideal center = calculate_center();
16:            else                                       ▷ Endchart in Hu's Algorithm
17:                datalist[3][1] = stat = "transformation";
18:                ideal center = ⟨0⟩          ▷ Go directly to state 'transformation'
19:    if !defined(center) then
20:        while (state == "gauss") do
21:            M = generic_gauss(M,nrows(M),ncols(M));
22:            state = calculate_state(M,datalist[4]);
23:            datalist[3][2] = minorsize;
24:        if (state == "binresol") then
25:            list binomiallist = GenerateListOfBinomials(M);
26:            ideal J = binomiallist[1];
27:            list BO = createBO(J);
28:        if (state == "Hu") then
29:            list centerlist = HuAlgorithm(M);
30:        if (state == "transformation") then
31:            ideal cent = ⟨0⟩;      ▷ dummy value for detecting the case in blowup
   method
32:        if (state == "monomial") then
33:            ring r_new, ideal J = CreateBinomialFromMinomialMatrix(M);
34:            setring(r_new);
35:            list BO = createBO(J);
36:            allRings[i] = r_new;
37:        if (min{ncols(M), nrows(M)} == 1) then
38:            list BO = createBO(minor(M, 1));
39:        if defined(BO) or defined(centerlist) then
40:            ideal center = compute_center();
41:    if (finished(allRings[i])== 1) then
42:        EndCharts.append(actualRing[i]);
43:    else
44:        allRings.append(BlowUpMethodDetresolve(datalist,center));
45: return(list(allRings,EndCharts));
```

The skeleton of this pseudo code is very similar to the one of Algorithm 40. The differences are that

- we do not know wheather the initial matrix is arbitrary, locally monomial or simultaneously locally monomial, so we have to handle this chart in the for-loop, too.
- we have to check first, at which state of the algorithm we are and afterward we have to use one of the subroutines.
- We use the fact that BO or HuList is only defined if we are not finished with tha calculation in a state using resbinomial2 or Hu's algorithm.

Note, that one has to guarantee that maximal one of BO and HuList is defined. And the lists only have to be defined if we are not ready with applying resbinomial2 or Hu's algorithm together with the corresponding translation.

*(End of Remark A.5.8.)*

Theoretically, one can implement this algorithm with an additional input parameter which encodes the algorithms we want to use since the algorithm is modular. In practice, the bottleneck is the implementation of other resolution algorithms, like, e.g., resolve.lib. We need a submethod which generates the considered data types and data sets, and in addition, we need a method which calculates the next center and modifies the data set. That is another reason why we reimplement the algorithm of Blanco in Section A.3.

Now we discuss how to calculate the current state of the process.

**Remark A.5.9** (Algorithm 48)**.** The *input* of Algorithm 48 is the considered matrix $M$ in the actual chart together with the list of exceptional divisors $E$. The *output* is a string which encodes the state of the resolution process.

**Algorithm 48** calculate_state(matrix $M$, list $E$)

```
1:  matrix N = strictTransformsOf(M,E);
2:  if (not isLocallyMonomial(N)) then
3:      return("binresolve");
4:  if (isMonomial(N) and not isresolvedMonomial(N)) then
5:      return("monomial");
6:  if (isresolvedMonomial(N) ) then
7:      return("gauss");
8:  if (isLocallyMonomial(N) and not isSimultaneouslyLocallyMonomial(N)) then
9:      return("Hu");
10: if (isSimultaneouslyLocallyMonomial(N) and not isMonomial(N)) then
11:     return("transformation");
```

First, we need a matrix $N$ whose entries are the strict transforms of the input matrix's $M$ entries. For a better readability, the single decisions, if a state is right, are outsourced in sub-methods.
Note that *method()* is shorthand for *method()* $==$ 1 and *not method()* is shorthand for *method()* $==$ 0, if the return type of the method is a boolean. *(End of Remark A.5.9.)*

Next we discuss the submethods of the calculation of the current state.

**Remark A.5.10** (Algorithm 49)**.** The *input* of Algorithm 49 is the matrix $N$ in Algorithm 48. The *output* is a boolean. The method returns 1, if every entry of the matrix is of the form $1 - \underline{x}^{\underline{A}}$ and 0 otherwise. Line 5 is singular notation for the

**Algorithm 49** isLocallyMonomial(matrix $M$)

```
1: LocallyMonomial = 1;
2: for (i = 1; i ≤ ncols(M); i++) do
3:     for (j = 1; j ≤ nrow(M); j++) do
4:         if (size(M[i, j]) == 2) then
5:             if (|M[i, j][1]| ≠ 1 and |M[i, j][2]| ≠ 1) then
6:                 LocallyMonomial = 0;
7: return(LocallyMonomial);
```

term $M[i,j] \notin \{1 \pm \underline{x}^{\underline{A}}, \underline{x}^{\underline{A}} \pm 1\}$. *(End of Remark A.5.10.)*

**Remark A.5.11** (Algorithm 50)**.** Since this algorithm is the last which is called in CALCULATE_STATE we can assume that the input matrix $M$ is not a monomial matrix but that has principalized entries.
The *input* of Algorithm 50 is the matrix $N$ in Algorithm 48. The *output* is a boolean. The method returns 1, if there exists a linear transformation such that every entry of $N$ is monomial. We use the fact that our implementation of Hu's algorithm only

**Algorithm 50** isSimultaneouslyLocallyMonomial(matrix $M$)

```
1: list HuList = HuAlgorithm(M);
2: list centerlist = HuList[1];
3: if (size(centerlist) == 1) then
4:     return(1);
5: return(0);
```

returns the necessary centers until our entries become normal crossing.
In case that we have normal crossing entries Hu's algorithm calculates the covering we use in the next 'transformation' step. Since this is stored in the list HuList which is a global variable, we do not have to return it. *(End of Remark A.5.11.)*

**Remark A.5.12** (Algorithm 51)**.** The *input* of Algorithm 51 is the matrix $N$ in Algorithm 48. The *output* is a boolean. The method returns 1, if every entry of $N$ is monomial.

**Algorithm 51** isMonomial(matrix $N$)

```
1: Monomial = 1;
2: for (i = 1; i ≤ ncols(N); i++) do
3:     for (j = 1; j ≤ nrow(N); j++) do
4:         if (size(N[i,j]) ≥ 2) then          ▷ more than one term in N[i,j]
5:             Monomial = 0;
6: return(0);
```

*(End of Remark A.5.12.)*

**Remark A.5.13** (Algorithm 52)**.** The *input* of Algorithm 52 is the matrix $N$ in Algorithm 48. The *output* is a boolean. The method returns 1, if every entry of $N$ is monomial and the ideal generated by the entries of $N$ equals $\langle 1 \rangle$.

**Algorithm 52** isresolvedMonomial(matrix $M$)

```
1: if isMonomial(M) == 0 then
2:     return(0);
3: ideal J = minor(M,1);
4: return(J == ⟨1⟩);
```

*(End of Remark A.5.13.)*

After discussing how to choose the right state, we present how to modify the data to use the black box RESBINOMIAL2.LIB as a black box. First, we discuss the modification in the determinantal binomial case.

**Remark A.5.14** (Algorithm 53)**.** The *input* of Algorithm 53 is the matrix $M$ in the considered chart. The *output* is a a list of all entries of $M$ which are binomial and not locally monomial.

**Algorithm 53** GenerateListOfBinomials(matrix $M$)

```
1: list returnlist;
2: for (i = 1; i ≤ #cols; i++) do
3:     for (j = 1; j ≤ #rows; j++) do
4:         if M[i,j] is binomial and not (locally) monomial then
5:             returnlist.add(M[i,j]);
6: return(returnlist);
```

*(End of Remark A.5.14.)*

In the determinantal monomial case, we have to adapt the ideal generated by the entries.

**Remark A.5.15** (Algorithm 54)**.** The *input* of Algorithm 54 is the matrix $M$ in the considered chart. The *output* is a ring with the additional $\underline{y}$-variables and the binomial ideal corresponding to the determinantal monomial case.

**Algorithm 54** CreateBinomialIdealFromMonomialMatrix(matrix $M$)

1: $\langle f_1, \ldots, f_m \rangle$ := std(minor($M$, 1));
2: $d$ := Max-ord($\langle f_1, \ldots, f_m \rangle$);
3: ring r_new = actual_ring[$\underline{y}$];
4: setring(r_new);
5: **for** ($i = 1$; $i \leq m$ ; $i++$) **do**
6: $\quad f_i = y_i^d + f_i$;
7: $I = \langle f_1, \ldots, f_m \rangle$
8: return(r_new,$I$);

Since our implementation is written for characteristic 0, we do not need two new variables $y_{2\ell}$ and $y_{2\ell+1}$ like discussed in Remark 8.1.1 and Remark 8.1.2. For a better worst-case time complexity, we will use as few as possible many variables. *(End of Remark A.5.15.)*

**Remark A.5.16** (Algorithm 55)**.** This method has no input data since every data has to be stored globally in the underlying ring. The output is an ideal describing the next center of the resolution process.

**Algorithm 55** calculate_center()

1: int $n$ = nvars(basering);
2: centerlist = HuList[1];
3: **if** (defined(centerlist)) **then**
4: $\quad$ center = centerlist.last;
5: $\quad$ centerlist = delete(centerlist, last);
6: **if** (defined(BO)) **then**
7: $\quad$ BO, $b$, center = resbinomial2.DescentInDimension(BO,1);
8: $\quad$ **if** (state == "monomial") **then**
9: $\quad\quad V(\underline{x}, \underline{y}) \mapsto V(\underline{x})$; ▷ delete $\underline{y}$-components
10: return(center);

The algorithm distinguishes two cases:

1. The center resulting from the algorithm of Hu. We have to choose the last entry, which is not empty in our center list, as the upcoming center.

Afterwards, this entry has to be deleted from the center list. If the center list is empty, we have to delete it.

2. The usage of resbinomial2.lib in the determinantally binomial case and the determinantally monomial case. Since both cases deal with the same algorithm, which only distinguishes the initial ideal, we can consider them together. Note that in the determinantal monomial case, we have to delete the $\underline{y}$-components of the center.

If there is a defined basic object, we use the center calculation method of resbinomial2.lib directly. We can delete the basic object variable, if we are in a final chart of a resbinomial2-call. *(End of Remark A.5.16.)*

We have discussed how to calculate the next center, so we are ready to do the blow-up.

**Remark A.5.17** (Algorithm 56)**.** The *input* of Algorithm 56 is the list datalist together with the center of the blow-up. The *output* the list of successor charts of the actual chart.

**Algorithm 56** BlowUpMethodDetresolve(list datalist, ideal center)

```
1: if (center ≠ ⟨0⟩) then
2:     successorList = blowUp3(datalist[1],center);
3:     numbSuccessors = size(successorList);
4:     for (i = 1; i ≤ numbSuccessors; i++) do
5:         setring(successorList[i]);
6:         datalist = blowUp(datalist);
7:         datalist[4].append(eD);
8:         path = path,[chartnumber,i];
9:         if (defined(BO)) then
10:            BO = blowUp(BO);
11:        else
12:            if defined(HuList) then
13:                successorlist = blowUp(HuList);
14:                HuList[3].append(eD);
15:            else                          ▷ Hu's algorithm finished just now
16:                state = "transformation";
17: else
18:     successorList = generateCovering(datalist);
19: return(successorList);
```

We first check whether we have to blow-up or if we have to cover our chart for the transformation of variables. Afterwards, we blow-up the necessary data. If Hu's

algorithm finishes at the moment, we store the state directly to reduce complexity. In the real implementation, there are more technical details, like reducing variables of the ring, if the determinantal monomial case finishes. But it would work without these technical details, so we do not focus on them here. *(End of Remark A.5.17.)*

**Remark A.5.18** (Algorithm 57)**.** The *input* of Algorithm 57 is the list datalist together with the center of the blow-up. The *output* the list of successor charts of the actual chart.

**Algorithm 57** generateCovering(list datalist, int chartnumber)

```
1: def r = basering;
2: for (i = 1; i ≤ size(HuList[2]);i++) do
3:     list parametersystem[i] = HuList[2][i][1];
4:     list complementlist[i] = HuList[2][i][2];
5: for (i = 1; i ≤ size(parametersystem);i++) do
6:     for (j = 1; j ≤ #HuList[3]; j++) do
7:         parametersystem[i].append(HuList[3][j]);
8:     parametersystem[i] = AddComponentToSystemOfParameters(parametersystem[i],
   complementlist[i]);
9: for (member = 1; member ≤ size(parametersystem); member++) do
10:    N = parametersystem[member];
11:    normcrossideal = ⟨N[1][i] | 1 ≤ i ≤ size(N[1])⟩
12:    complementideal = ⟨N[2][i] | 1 ≤ i ≤ size(N[2])⟩
13:    def newring = ℂ[t]; ▷ new variables t, same number as in r
14:    map f = newring, normcrossideal;
15:    datalist = sat(datalist,complementideal);
16:    setring(newring);
17:    datalist = preimage(r,f,datalist);
18:    path = path,[chartnumber,member];
19:    successorList.add(newring);
20: return(successorList);
```

First, we read the data from HuList[2]. The first entry is a list of system of parameters and the second list describes the generators of the covering $D(h_1 \cdots h_r)$. Every system of parameters need exactly $n$ (variables of the unserlying ring) normal crossing hypersurfaces. So we have to add the exceptional divisors and maybe some more components which are calculated in the AddComponentToSystemOfParameters-method.
Afterwards, we can cover the actual chart by the different $D(h_1 \cdots h_r)$-charts with the corresponding system of parameters.
We transfer the data (like in the blow-up method) to the chart of the covering and saturate with $h_1 \cdots h_r$. *(End of Remark A.5.18.)*

Note, that a covering with a single chart is also stored as a successor chart although we could do it in the same chart in theory. This fact yield a higher number of total chart in our implementation. We will see this in the unit tests concerning the ((skew-)symmetric) generic cases in Section C.2.

**Remark A.5.19** (Algorithm 58)**.** The *input* of Algorithm 58 is the list parametersystem together with the list complementlist in Algorithm 57. The *output* the updated list parametersystem

**Algorithm 58** AddComponentToSystemOfParameters(list parameterlist, list complementlist)

```
1: def r = basering;
2: int n = nvars(r);
3: if (size(parametersystem) == n) then
4:     return(parametersystem);
5: parametersystemideal = ⟨N[1][i] | 1 ≤ i ≤ size(N[1])⟩;
6: complementideal = ⟨N[2][i] | 1 ≤ i ≤ size(N[2])⟩;
7: for (i = 1; i ≤ n; i++) do
8:     if (size(parametersystem) == n) then
9:         return(parametersystem);
10:    if (x_i is not contained in parametersystem) then      ▷ x_i denotes the i-th
   variable in r
11:        if (parametersystemideal + ⟨x_i⟩ == ⟨1⟩) then
12:            parametersystem.append(x_i);
13:            parametersystemideal += ⟨x_i⟩;
14:        if (parametersystemideal + ⟨x_i + 1⟩ == ⟨1⟩) then
15:            parametersystem.append(x_i + 1);
16:            parametersystemideal += ⟨x_i + 1⟩;
17:        if (parametersystemideal + ⟨x_i − 1⟩ == ⟨1⟩) then
18:            parametersystem.append(x_i − 1);
19:            parametersystemideal += ⟨x_i − 1⟩;
20: return(parametersystem);
```

If the system of parameters has the right size, we do not need to add anything. Otherwise, the idea is to add $f_i = x_i + c_i$, where $c_i \in \{0, 1, -1\}$ and $x_i$ is a variable in the basering which is not contained in the system of parameters. Every $f_i$ has to fullfil the following conditions:

- $f_i$ has to lie in $D(h_1 \cdots h_r)$ and
- $f_i$ has to intersect each of the other components in the intersection point.

*(End of Remark A.5.19.)*

The last thing we have to discuss is the detection of the final charts.
Since we have used the algorithm of Blanco and Hu as a black box and after the algorithm of Blanco is finished we go on at the next step (Hu or Gauß) of our calculation, if the considered singularity is not resolved yet. We should not interrupt the execution of the algorithm of Blanco. That is the reason why we only call the FINISHED-method in Algorithm 47 after an application of Hu or Blanco finished or before application of Hu or Blanco started and nowhere inbetween.
At the end of an application of Blanco and Hu, we have normal crossings with the exceptional divisors. Afterwards we check, if the current chart is a final chart by testing if the singular locus (SLOCUS-method in SINGULAR) of the variety generated by the minors of the considered size is empty or not.

So we have discussed all parts of the implementation of the resolution of determinantal singularities of at most binomial type. Comparisons with state of the art libraries can be found in Section B.6.

# B. Some remarks on the complexity of the algorithms

Before stating some remarks on the complexity of our algorithms in Section B.3, we give in Section B.1 the background to complexity theory in general.

## B.1. Theoretical background in complexity theory

We define the usual complexity classes in the typical complexity theory and give a context to the results we have seen in the thesis. We refer to [4] for more information and details.

**Definition B.1.1.**

- The complexity class P is the class of problems which are solvable in deterministic polynomial time ([4, Definition 1.20]).
- The complexity class NP is the class of problems which are verifiable in determinitic polynomial time, i.e., solvable in nondeterministic polynomial time ([4, Definition 2.1]).
- The complexity class EXPTIME is the class of problems which are solvable in determinitic exponential time, i.e., running time is bounded by some $f(n) = 2^{n^{\mathcal{O}(1)}}$ ([4, Definition 2.24]).
- The complexity class EXPSPACE is the class of problems which are solvable in determinitic exponential space, i.e., the space is bounded by some $f(n) = n^{\mathcal{O}(1)}$ (analogous to [4, Definition 2.24]).

**Remark B.1.2.** The following chain of inclusion holds

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{EXPTIME} \subseteq \mathrm{EXPSPACE},$$

where $\mathrm{P} \subsetneq \mathrm{EXPTIME}$ holds.

For practical reasons, the classes P and NP are the classes of the most interesting problems because problems in P are efficiently solvable. It is a famous open problem

whether P $\neq$ NP or not.

In order to compare problems and classes of problems, we need the concept of reduction.

**Definition B.1.3.** ([4, Definition 2.7]). We say a problem $A \subseteq \Sigma^*$ is polynomial many-one reducible (pm-reducible) to a problem $B \subseteq \Gamma^*$ denoted by $A \leq_m^{\mathrm{P}} B$, if there is a polynomial-time computable function $f\colon \Sigma^* \to \Gamma^*$ such that for every $x \in \Sigma^*$, $x \in A \iff f(x) \in B$.

**Definition B.1.4.** ([4, Definition 2.7]) Let $\mathcal{C}$ be a complexity class. We say a problem $A$ is $\mathcal{C}$-hard, if for every problem $B \in \mathcal{C}$ there is a pm-reduction such that $B \leq_m^{\mathrm{P}} A$ holds.
We say a problem $A$ is $\mathcal{C}$-complete, if $A \in \mathcal{C}$ and $A$ is $\mathcal{C}$-hard.

So the notion of hardness gives a lower bound on the complexity, and the notion of completeness gives an exact classification in this class. In other words, we have stated in Remark 5.1.10, that the Gröbner basis computation is EXPSPACE-complete, so there is no running time $\leq 2^{n^{\mathcal{O}(1)}}$ and no space requirement $\leq 2^{n^{\mathcal{O}(1)}}$.

In theoretical computer science and complexity theory, people say such an algorithm for Gröbner basis computing cannot be efficient. Nevertheless, we consider more inefficient classes when dealing with Hironaka-style resolutions, so we have to introduce the Grzegorczyk hierarchy.

**Definition B.1.5.** [47]. Let $E_0(x, y) = x + y$, let $E_1(x) = x^2 + 2$ and we define iteratively for $n \geq 2$

$$E_n = E_{n-1}^x(2).$$

We define the Grzegorczyk hierarchy as follows. Let $\varepsilon^n$ be the $n$-th set of the hierarchy, then it contains the following functions

1. $E_k$, for $k < n$,
2. $Z(x) \equiv 0$,
3. $S(x) = x + 1$,
4. $p_i^m(t_1, \ldots, t_m) = t_i$,
5. the composition of functions in this set and
6. results of limited (primitive) recursion applied to functions in the set.

**Remark B.1.6.** The inclusion chain of the sets in the hierarchy is a chain of strict inclusions, i.e.,

$$\varepsilon^0 \subsetneq \varepsilon^1 \subsetneq \varepsilon^2 \subsetneq \dots.$$

This belongs to the fact that the $n$-th hyper operation is in $\varepsilon^n$ but not in $\varepsilon^{n-1}$.

**Example B.1.7.**

- $\{x+1, x+2, \dots\} \subseteq \varepsilon^0$
- $\{x+y, 6x, \dots\} \subseteq \varepsilon^1$, so $\varepsilon^1$ provides all addition functions.
- $\{x \cdot y, x^6, \dots\} \subseteq \varepsilon^2$, i.e., $\varepsilon^2$ provides all multiplication functions.
- $\{x^y, 2^{2^{2^x}}, \dots\} \subseteq \varepsilon^3$, i.e., $\varepsilon^3$ provides all exponential functions.
- $\varepsilon^4$ provides all tetration functions.

These examples show that the $(\ell+3)$-th level of the hierarchy can potentially contain very fast-growing functions (depending on $\ell$). This may illustrate the complexity of the Hironaka-style resolution of large dimensional singularities.

## B.2. Complexity analysis of the generic determinantal case

For interests of computational complexity, it might be interesting to know if this specific algorithm is less complex than a general algorithm for the resolution of singularities. An effective Hironaka resolution of a scheme $X$ is classified to be an element of the $(\dim(X)+3)$-rd level of the Grzegorczyk hierarchy [5]. So for practical reasons, a general algorithm is very unusable.

Therefore we give some remarks on the number of considered charts and the number of final charts in the three cases. We start with the generic case.

**Remark B.2.1.** The first blow-up of $M_{n,m}$ is covered by $n \cdot m$ charts. In there, we can interpret the transform of $M$ as a generic $(n-1) \times (m-1)$ matrix.
A second blow-up gives us $(n-1) \cdot (m-1)$ additional charts in every of the $n \cdot m$ charts, so for resolution of the 2-minors we need

$$\underbrace{n \cdot m}_{\text{first blow-up}} + \underbrace{n \cdot m \cdot (n-1) \cdot (m-1)}_{\text{second blow-up}}$$

charts where the last $n \cdot m \cdot (n-1) \cdot (m-1)$ are the final charts and so on. In total for a generic matrix, we need to consider

$$\sum_{i=1}^{\min\{r,k\}-1} \prod_{j=i}^{\min\{r,k\}} (r-(r-j)) \cdot (k-(k-j))$$

charts. Only the charts which are created by the last blow-up are final charts so we have

$$\prod_{j=0}^{\min\{r,k\}-1} (r-j) \cdot (k-j)$$

final charts.
Thus for a generic quadratic matrix the described resolution procedure (see Example 6.1.1) has

$$\sum_{i=1}^{n-1} \prod_{j=i}^{n} (n-(n-j))^2$$

charts and

$$\prod_{j=0}^{n-1} (n-j)^2$$

final charts.

We continue with the generic skew-symmetric case.

**Remark B.2.2.** Let $M$ be the generic skew-symmetric matrix of dimension $n \times n$, where $n = 2m$ is an even positive number. The goal is to resolve the ideal generated by the maximal pfaffians of $M$. We have seen that we the first blow-up generates $t_{n-1} = \frac{n(n-1)}{2}$ charts. Afterwards, in all of these charts we have the transform $M'$ of $M$ which is the generic skew-symmetric matrix of dimension $(n-2) \times (n-2)$. In all of these charts, the second blow-up generates $t_{n-3}$ further charts. For a skew-symmetric matrix we have

$$\prod_{j=0}^{m-1} t_{2j+1}$$

final charts, where $t_i = \frac{i(i+1)}{2}$ denotes the $i$-th triangular number and

$$\underbrace{1}_{\text{initial chart}} + \sum_{i=0}^{m-1} \prod_{j=1}^{m-1} t_{2j+1}$$

total charts. For quadratic skew-symmetric matrices $M$ of odd dimension, $\det(M) = 0$ holds, so there is no blow-up needed for resolution.

The quadratic symmetric generic matrix case is more complicate and we have to distinguish several types of charts.

**Remark B.2.3.** Let us have a look at the generic symmetric matrix $B_m$. Our subgoal is to find a sequence $(c_i)_{0\leq i\leq m-2}$ for the number of charts where the transform is of the form $B_{m-i}$. This matrix lives in the initial chart. This implies $c_0 = 1$. After one blow-up, we have $t_{n-1}$ of $X_{j,k}$-charts with $j \neq k$ where we can consider the transform of $B_m$ as some matrix $B_{m-1}$. Therefore $c_1 = t_{n-1}$. In addition to these charts, we have $m$ charts of the type $X_{j,j}$-chart. In there, we can consider the transform as a matrix of the form $B_{m-2}$.
The second blow-up is only necessary for the charts with the transform as a matrix $B_{m-1}$. In each of the $t_{m-1}$ charts, we obtain $t_{n-2}$ charts of the $X_{j,k}$-type and $m-1$ charts of the $X_{j,j}$-type. This leads to

$$c_2 = \underbrace{n}_{\#X_{j,j}\text{-charts after first blow-up}} + \underbrace{t_{n-1}\cdot t_{n-2}}_{\#X_{j,k}\text{-charts after second blow-up}}.$$

By applying this argument inductively, we get

$$c_i = \underbrace{c_{i-2}(n-(i-1))}_{\#X_{j,j}\text{-charts}} + \underbrace{c_{i-1}\cdot t_{n-i}}_{\#X_{j,k}\text{-charts}}, \text{ for } i \geq 2.$$

The total number of charts is

$$\sum_{i=0}^{m-2} c_i$$

and the number of final charts is $c_{m-2}$.

## B.3. Comparing the variants of local monomialization

Let us compare the discussed algorithms for monomializing a binomial in Section 7. First, we analyze the complexity of the algorithms by estimating the maximal possible number of blowups needed to monomialize a given binomial. After that we turn out attention to explicit examples, where we compare the numbers of charts appearing along the monomialization and as well as the number of final charts. At the end, we briefly look at the question, whether the different choices for the centers in a fixed method have an impact on the resulting numbers.

During this section we assume to $K$ be a field and $f = \underline{x}^A - \rho\underline{x}^B \in K[\underline{x}] := K[x_1,\ldots,x_n]$ be a binomial, where $\rho \in K^\times$ and $\underline{x}^A = x_1^{A_1}\cdots x_n^{A_n}$ for $A =$

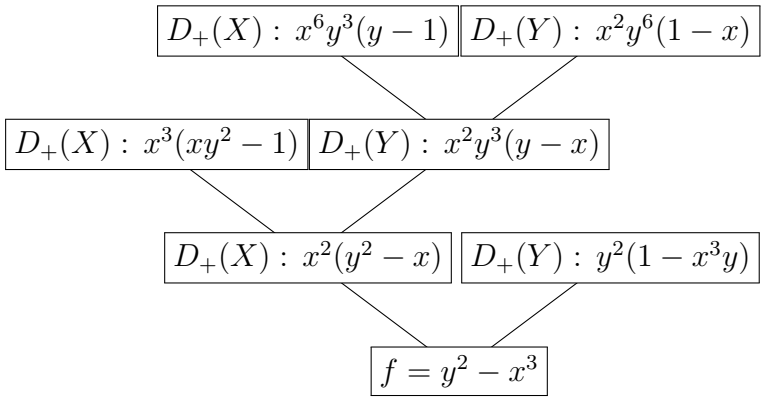



Figure B.1.: Example for the tree structure of a blowup process for $f = y^2 - x^3$.

$(A_1, \ldots, A_n) \in \mathbb{Z}^n_{\geq 0}$. Furthermore, we use the notation of Chapter 7.

**Remark B.3.1.** We can interpret the process of blowing up as a tree structure. The vertices correspond to the charts, where we put the original data on level 0 and all charts, which arise after the $\ell$-th blowup are put on level $\ell$. Two vertices $v$ on level $\ell$ and $w$ on level $\ell + 1$ are connected by an edge if the chart corresponding to $w$ is one of the charts of the blowup in $v$. The unique vertex on level 0 is called the *root* of the tree and vertices on level $\ell$, which are not connected to any vertex of a higher level, are called *leaves* of the tree. The latter correspond to the final charts of the blowup procedure. In Figure B.1, we illustrate the tree structure for a simple example. In the boxes we indicate, which of the charts of the blowup we are considering, and we provide the total transform of $f$. We abuse notation and denote the coordinates in each chart by $(x, y)$. In every case, where a blowup is performed, the ideal of the center is $\langle x, y \rangle$. The number $(\ell)$ on the right marks the level. There are two leaves on level 3 and one leaf each on level 2 and 1.

The number of charts is delimited by the number of charts, which we newly create after a blowup, and by the longest path from the root to any leaf. The first of these numbers is determined by the codimension of the center. This provides the following bounds:

| mode | maximal codimension of a possible center |
|---|---|
| 1 (max.ord.) | $n =$ (number of variables) |
| 2 (codim.2) | 2 |
| 3 (min.codim.) | 4 |
| 4 (exc.) | 4 |

(B.3.1)

Using the invariant, which we introduced to prove the termination of the respective variant for monomialization, we can bound the length of the longest path of the resulting tree of blowups.

**Lemma B.3.2.** *Let $g = \underline{x}^A - \rho\underline{x}^B \in K[\underline{x}] = K[x_1, \ldots, x_n]$ with $\rho \in K^\times$, $A, B \in \mathbb{Z}^n_{\geq 0}$ such that $A_iB_i = 0$ for all $i \in \{1, \ldots, n\}$. The following are upper bounds for the longest path from the root to any leaf in the blowup tree of the respective mode:*

| *mode* | *upper bound for the longest path* |
|---|---|
| 1 *(max.ord.)* | $2^{m-1}M + m - 1 - \sum_{\ell=1}^{m-1} 2^{m-\ell-1}(m-\ell+1)$ |
| 2 *(codim.2)* | $(\alpha+\beta-4)(n-1) + \mathfrak{a} + \mathfrak{b} + 1$ |
| 3 *(min.codim.)* | $(\alpha+\beta-4)(n-1) + \mathfrak{a} + \mathfrak{b} + 1$ |
| 4 *(exc.)* | $(\alpha+\beta-4)(n-1) + \mathfrak{a} + \mathfrak{b} + 1$ |

(B.3.2)

*where*

$$\begin{aligned} m &:= \min\{|A|, |B|\}, & M &:= \max\{|A|, |B|\}, \\ \alpha &:= \max\{A_i \mid i \in \{1, \ldots, n\}\}, & \mathfrak{a} &:= \#\{i \in \{1, \ldots, n\} \mid A_i = \alpha(g)\}, \\ \beta &:= \max\{B_i \mid i \in \{1, \ldots, n\}\}, & \mathfrak{b} &:= \#\{i \in \{1, \ldots, n\} \mid B_i = \beta(g)\}. \end{aligned}$$

*Proof.* Recall that for centers contained in maximal order locus of $g$ (mode $= 1$), we introduced $\mathrm{inv}(g) = \big(\min\{|A|, |B|\}, \max\{|A|, |B|\}\big) \in \mathbb{Z}^2_{\geq 0}$ as measure for the complexity of the singularity (Definition 7.2.6), which strictly decreases with respect to the lexicographical ordering on $\mathbb{Z}^2_{\geq 0}$ after each blowup. We have $\mathrm{inv}(g) = (m, M)$. Observe that $m \leq M$. Let $(m', M')$ be the value of $\mathrm{inv}(g')$ in a chart after the blowup in a center contained in the locus of maximal order of $g$. The proof of Proposition 7.2.7 provides that either

(*i*) $(m', M') \leq_{\ell p} (m - 1, 2M - m)$, or

$(ii)$ $(m', M') \leq_{\ell p} (m, M-1)$.

If we are $k_1$ times in case $(ii)$ and then once in $(i)$, we obtain that the value of inv(.) is bounded by

$$(m-1,\ 2(M-k_1)-m).$$

Note that $k_1 \in \{0, \ldots, M-m\}$ since we cannot have $M-1 < m$ in $(ii)$. At this stage, we performed $k_1+1$ blowups. Iterating this, the upper bound for the invariant becomes

$$\Big(m-s,\ 2^s M - \sum_{\ell=1}^{s} 2^{s-\ell+1} k_\ell - \sum_{\ell=1}^{s} 2^{s-\ell}(m-\ell+1)\Big)$$

after $s$ steps, where, for $\ell \in \{1, \ldots, s\}$,

$$k_\ell \in \{0, \ldots, 2^{\ell-1}M - \sum_{q=1}^{\ell-1} 2^{\ell-q} k_q - \sum_{q=1}^{\ell-1} 2^{\ell-1-q}(m-q+1) - (m-\ell+1)\},$$

is the number of times that we are in case $(ii)$ until the first entry drops from $m-\ell+1$ to $m-\ell$ in case $(i)$. The number of blowups up to this point is $k_1 + \ldots + k_s + s$.

If $s = m-1$, the first entry of the bound for inv(.) is 1. In particular, we can only be in case $(ii)$ for the remaining decreases if we want to determine the maximal length of a path in the blowup tree. Hence, the second entry for the bound determines the number of blowups remaining. In total, we obtain the bound

$$\sum_{\ell=1}^{m-1} k_\ell + (m-1) + 2^{m-1}M - \sum_{\ell=1}^{m-1} 2^{m-\ell} k_\ell - \sum_{\ell=1}^{m-1} 2^{m-\ell-1}(m-\ell+1) =$$

$$= 2^{m-1}M + m - 1 - \sum_{\ell=1}^{m-1} (2^{m-\ell}-1) k_\ell - \sum_{\ell=1}^{m-1} 2^{m-\ell-1}(m-\ell+1).$$

We aim to maximize the bound for the number of blowups. The only variation in the expression are the numbers $k_1, \ldots, k_{m-1}$. Since $2^{m-\ell} - 1 \geq 1$ the maximum is obtained if $k_\ell = 0$, for all $\ell \in \{1, \ldots, m-1\}$. In conclusion, we have proven the bound of (B.3.2) for mode $= 1$.

Let us consider the remaining three cases (mode $\in \{2, 3, 4\}$). For each of them, we used $\iota(g) = (\alpha(g), \mathfrak{a}(g), \beta(g), \mathfrak{b}(g)) \in \mathbb{Z}^4_{\geq 0}$ of Definition 7.1.1 to measure the improvement of the singularity after the blowup following the respective strategy. We have $\iota(g) = (\alpha, \mathfrak{a}, \beta, \mathfrak{b})$. Note that $\mathfrak{a} + \mathfrak{b} \leq n$. Let us fix the value of mode $\in \{2, 3, 4\}$.

Let $(\alpha', \mathfrak{a}', \beta', \mathfrak{b}')$ be the value of $\iota(g')$ after the blowup in the center, which is determined by the strategy given by mode. Let $j \in \{1, \ldots, n\}$ be such that we

are in the $X_j$-chart. By Propositions 7.3.5, 7.4.2, and Corollary 7.5.2, we have $(\alpha', \mathfrak{a}', \beta', \mathfrak{b}') <_{\ell p} (\alpha, \mathfrak{a}, \beta, \mathfrak{b})$. The decrease can be made more precise depending on which chart we are. Since $g = \underline{x}^A - \rho \underline{x}^B$ and $A_i B_i = 0$ for all $i \in \{1, \ldots, n\}$, we either have $A_j \neq 0$ (case $(I)$) or $B_j \neq 0$ (case $(II)$). In these two cases, the proofs of Propositions 7.3.5, 7.4.2, and Corollary 7.5.2 provide that we have:

$$(I) \quad \begin{cases} (\alpha', \mathfrak{a}', \beta', \mathfrak{b}') = (\alpha, \mathfrak{a} - 1, \beta, \mathfrak{b}), & \text{if } \mathfrak{a} > 1; \\ (\alpha', \mathfrak{a}', \beta', \mathfrak{b}') = (\alpha', \mathfrak{a}', \beta, \mathfrak{b}), & \text{with } \alpha' < \alpha, \quad \text{if } \mathfrak{a} = 1; \end{cases}$$

$$(II) \quad \begin{cases} (\alpha', \mathfrak{a}', \beta', \mathfrak{b}') = (\alpha, \mathfrak{a}, \beta, \mathfrak{b} - 1), & \text{if } \mathfrak{b} > 1 \\ (\alpha', \mathfrak{a}', \beta', \mathfrak{b}') = (\alpha, \mathfrak{a}, \beta', \mathfrak{b}'), & \text{with } \beta' < \beta, \quad \text{if } \mathfrak{b} = 1. \end{cases}$$

Note that $\mathfrak{a}' \leq n - \mathfrak{b}$ in $(I)$ and $\mathfrak{b}' \leq n - \mathfrak{a}$ in $(II)$. Since in $(I)$ (resp. $(II)$), the last two (resp. first two) entries remain the same, the operations determined by $(I)$ and $(II)$ are independent of each other.

Due to the bound of $\mathfrak{a}'$ and $\mathfrak{b}'$, we get the longest path in the resolution tree if both the value of $\mathfrak{a}(.)$ and $\mathfrak{b}(.)$ decrease to the value 1 first. This is achieved after $\mathfrak{a} + \mathfrak{b} - 2$ blowups following the respective monomialization procedure. We denote the strict transform of $g$ at this step by $g''$, which is obtained after factoring the monomial part from the total transform of $g$. Notice that we have $\iota(g'') = (\alpha, 1, \beta, 1)$. After the next blowup the invariant is at most $(\alpha - 1, n - 1, \beta, 1)$ (in case $(I)$) resp. $(\alpha, 1, \beta - 1, n - 1)$ (in case $(II)$). If the maximum is attained, then we get after $n - 2$ further blowups the value $(\alpha - 1, 1, \beta, 1)$ resp. $(\alpha, 1, \beta - 1, 1)$. Therefore, after at most $\mathfrak{a} + \mathfrak{b} + (\alpha - 2)(n - 1) + (\beta - 2)(n - 1)$ blowups, the value of $\iota(.)$ becomes $(1, 1, 1, 1)$. The latter means that the total transform of $g$ is of the form $\underline{x}^C(x_1 - \rho x_2)$, for some $C \in \mathbb{Z}^n_{\geq 0}$ and appropriately chosen variables. After one more blowup, the total transform of $g$ fulfills condition (7.0.1) and thus is locally monomial and the assertion follows. □

As an immediate consequence of (B.3.1) and Lemma B.3.2, we obtain:

**Corollary B.3.3.** *Let $g = \underline{x}^A - \rho \underline{x}^B \in K[x_1, \ldots, x_n]$ with $\rho \in K^\times$ and $A, B \in \mathbb{Z}^n_{\geq 0}$ such that $A_i B_i = 0$ for all $i \in \{1, \ldots, n\}$. Using the notation of Lemma B.3.2, the following are upper bounds for the number of charts in the respective variant for*

*monomializing g:*

| *mode* | *upper bound for the number of charts* |
|---|---|
| 1 *(max.ord.)* | $n^{d(m,M)}$ <br> *for* $d(m,M) := 2^{m-1}M + m - 1 - \sum_{\ell=1}^{m-1} 2^{m-\ell-1}(m-\ell+1)$ |
| 2 *(codim.2)* | $2^{(\alpha+\beta-4)(n-1)+\mathfrak{a}+\mathfrak{b}+1}$ |
| 3 *(min.codim.)* | $4^{(\alpha+\beta-4)(n-1)+\mathfrak{a}+\mathfrak{b}+1}$ |
| 4 *(exc.)* | $4^{(\alpha+\beta-4)(n-1)+\mathfrak{a}+\mathfrak{b}+1}$ |

Moreover, this worst case number of charts yields to a non-polynomial time algorithm in the number of variables and the degree of the binomial. The running time results as a product of the running time per chart times the number of charts and for each procedure the numer of charts is potentially exponential in $\alpha, \beta, \mathfrak{a}, \mathfrak{b}$ resp. $m, M$, where $\alpha$ and $\beta$ are bounded by the degree of the binomial, $\mathfrak{a}$ and $\mathfrak{b}$ are bounded by the number of variables and $M$ and $m$ are bounded by the degree of $g$, too.

Of course, the upper bounds are quite rough and the concrete number of blowups can be far smaller for explicit examples. For example, in the variant, where we choose the centers in the locus of maximal order, the codimension of the center is not necessarily always $n$ in every blowup.

Let us come to the study of explicit examples. In Figures B.2 and B.3, we provide several examples, where we consider the number of leaves and the number of total charts for each method for choosing the center. All examples are computed via an explicit implementation in SINGULAR of the algorithms described in the previous sections and the base field is always $\mathbb{Q}$. In the following, we discuss patterns, which can be observed in the examples, and we provide some indications for the noticed behavior of the different methods. Nonetheless, we do not provide rigorous proofs for the patterns in general.

**Example B.3.4** (Figure B.2, Examples 1–10)**.** In the first block of examples, all exponents of the starting binomial are one. This has the effect that the variant choosing centers of codimension two has a strong advantage, as can be easily seen in the number of charts. The reason for this is that the codimension of the centers in the other methods is very large, but the effect of the blowup is not much different than with a center of codimension two.

| | binomial | max.ord. | codim.2 | min.codim | exc. |
|---|---|---|---|---|---|
| 1. | $x_1x_2 - x_3x_4$ | 4 / 5 | 2 / 3 | 4 / 5 | 4 / 5 |
| 2. | $x_1x_2 - x_3x_4x_5$ | 10 / 13 | 3 / 5 | 10 / 13 | 10 / 13 |
| 3. | $x_1x_2 - x_3x_4x_5x_6$ | 22 / 29 | 4 / 7 | 22 / 29 | 22 / 29 |
| 4. | $x_1x_2x_3 - x_4x_5x_6$ | 60 / 79 | 6 / 11 | 40 / 53 | 40 / 53 |
| 5. | $x_1x_2x_3 - x_4x_5x_6x_7$ | 246 / 325 | 10 / 19 | 124 / 165 | 124 / 165 |
| 6. | $x_1x_2x_3 - x_4 \cdots x_8$ | 876 / 1.159 | 15 / 29 | 340 / 453 | 340 / 453 |
| 7. | $x_1 \cdots x_4 - x_5 \cdots x_8$ | 1.968 / 2.601 | 20 / 39 | 496 / 661 | 496 / 661 |
| 9. | $x_1 \cdots x_4 - x_5 \cdots x_9$ | 11.376 / 15.041 | 35 / 69 | 1.672 / 2.229 | 124 / 165 |
| 10. | $x_1 \cdots x_5 - x_6 \cdots x_{10}$ | 113.760 / 150.411 | 70 / 139 | 6.688 / 8.917 | 6.688 / 8.917 |
| 11. | $x_1x_2 - x_3^2$ | 3 / 4 | 3 / 5 | 3 / 4 | 3 / 4 |
| 12. | $x_1x_2x_3 - x_4^2$ | 7 / 10 | 7 / 13 | 7 / 10 | 7 / 10 |
| 13. | $x_1x_2x_3x_4 - x_5^2$ | 15 / 22 | 15 / 29 | 15 / 22 | 15 / 22 |
| 14. | $x_1x_2x_3 - x_4^4$ | 21 / 33 | 21 / 41 | 21 / 31 | 21 / 38 |
| 15. | $x_1x_2x_3x_4 - x_5^4$ | 85 / 134 | 85 / 169 | 85 / 127 | 85 / 162 |
| 16. | $x_1 \cdots x_5 - x_6^4$ | 341 / 538 | 341 / 681 | 341 / 511 | 341 / 666 |
| 17. | $x_1 \cdots x_6 - x_7^4$ | 1.365 / 2.154 | 1.365 / 2.729 | 1.365 / 2.047 | 1.365 / 2.698 |
| 18. | $x_1x_2 - x_3^3$ | 4 / 6 | 4 / 7 | 4 / 6 | 4 / 6 |
| 19. | $x_1x_2x_3 - x_4^3$ | 13 / 20 | 13 / 25 | 19 / 30 | 13 / 22 |
| 20. | $x_1x_2x_3x_4 - x_5^3$ | 40 / 62 | 40 / 79 | 104 / 164 | 40 / 72 |
| 21. | $x_1x_2x_3 - x_4^5$ | 31 / 47 | 31 / 61 | 43 / 67 | 31 / 58 |
| 22. | $x_1x_2x_3x_4 - x_5^5$ | 236 / 364 | 156 / 311 | 364 / 565 | 156 / 304 |
| 23. | $x_1 \cdots x_5 - x_6^5$ | 1.181 / 1.822 | 781 / 1.561 | 3.381 / 5.223 | 781 / 1.546 |
| 24. | $x_1 \cdots x_6 - x_7^5$ | 5.906 / 9.112 | 3.906 / 7.811 | 32.782 / 50.521 | 3.906 / 7.780 |

Figure B.2.: List of examples. In the last four columns, the entries are “the number of leaves/total number of charts".

Let us illustrate this for $g = x_1x_2x_3 - x_4x_5x_6$. The codimension two center, which will choose is $V(x_1, x_4)$. In the $X_1$-chart of the blowup, the strict transform of $g$ is $g' = x_2x_3 - x_4x_5x_6$. (Here, we abuse notation and denote the variables in the chart of the blowup also by $x_1, \ldots, x_6$.) In the $X_4$-chart, we obtain the strict transform $x_1x_2x_3 - x_5x_6$.

On the other hand, the variant choosing a center in the locus of maximal order determines the origin $V(x_1, \ldots, x_6)$ as the unique center. In the $X_1$-chart of the corresponding blowup, the strict transform of $g$ is $g' = x_2x_3 - x_4x_5x_6$. This is the same as the one before. In the remaining 5 charts, the strict transforms are of the same (up to renaming the variables). In contrast to the codimension two center, we have 6 different charts instead of only 2.

For the other two methods, the center is $V(x_1, x_2, x_4, x_5)$ and the analogous behavior appears, as the reader may verify.

**Example B.3.5** (Figure B.2, Examples 11–17)**.** In the second block of examples, all binomials have a term $x_n^{2k}$ appearing and the other monomial is of the form as in Examples 1–10. The number of leaves is in all cases the same, the total number of charts varies. The variant which seems to be most efficient is the one, where we choose the centers in the singular locus with minimal codimension. Note that the centers in this case are all of codimension three.

If $k = 1$, the maximal order is two. Therefore, the centers for mode $\in \{1, 3, 4\}$ coincide. Since the codimension of the center is three in these cases, none of the variables remaining in the strict transforms are exceptional. Moreover, already for $x_1x_2 - x_3^2$ one can verify that the codimension two centers requires more blowups to monomialize than the other methods. In the latter, the blowup with center $V(x_1, x_2, x_3)$ is sufficient.

This changes slightly for $k = 2$ and the differences in the number of total charts increases. The variant for mode $= 3$ remains the most efficient one. For example, the maximal order locus of $x_1x_2x_3 - x_4^4 = 0$ is $V(x_1, x_2, x_3, x_4)$, while $V(x_1, x_2, x_4)$ is the chosen center of the minimal codimension contained in the singular locus.

**Example B.3.6** (Figure B.2, Examples 18–24)**.** The last block of examples in Figure B.2 is of the same form as the previous one, but we have $x_n^{2k+1}$ as a monomial. As one can see in the numbers, the behavior changes, except for Example 18.

If the maximal order of the binomial is $\leq 3$, then the variant, which chooses the center in the locus of maximal order, is the best choice. One can verify that in this method, the behavior is the same as in the previous block.

One of the reasons for the large numbers for mode $= 3$ (centers of minimal codimension contained in the singular locus) is that the codimension is three at the

beginning. For example, if $k = 1$, the strict transform of the binomial in the $X_n$-chart of the first blowup is $x_1 \cdots x_{n-1} - x_n$. This is not transversal to the exceptional divisor, which is given by $x_n = 0$ in this chart. Therefore $n - 1$ more blowups are necessary in this chart. Moreover, in the $X_1$-chart, we obtain $x_2 \cdots x_{n-1} - x_1 x_n^3$ as strict transform, so $x_1$ just switch the side in the binomial. In contrast to this, the variable $x_1$ disappears in the $X_1$-chart of the blowup in $V(x_1, x_2, x_3, x_n)$ (which is the center for mode = 1).

The phenomenon that a variable is switching sides also appears for the other two methods (mode $\in \{2, 4\}$), but it has less impact for the codimension two centers as we are creating less charts, where the mentioned blowups arise. On the other hand, for examples of this kind of larger maximal order, the codimension 2 center become more efficient and the last variant (mode = 4) is slightly better. The reasons for the latter are the same as in Example B.3.5.

**Example B.3.7** (Figure B.3, Examples 25–29)**.** Let us turn to examples, where the appearing exponents are larger. More precisely, we consider binomials of the form $x_1^2 - x_2^3 x_3^4 \cdots x_n^{n+1}$, for $n \in \mathbb{Z}_{\geq 3}$. The large exponents have the effect that the centers are of codimension two at the beginning of all variants. Due to the exceptional divisors created there, the centers for mode = 4 coincide with the centers in the codimension two variant (mode = 2).

For centers in the locus of maximal order and centers of minimal codimension (mode $\in \{1, 3\}$), there appear eventually centers of higher codimension. For example, in $x_1^2 - x_2^3 \cdots x_5^6$, we obtain after five blowups with centers of codimension two the strict transform $x_1^2 - x_2 x_4 x_5^6$ and $(x_2, x_3, x_4)$ are exceptional. Our variant for choosing centers in the locus of maximal order, determines $V(x_1, x_2, x_4)$ as the next center, while, for mode = 3, the next center is $V(x_1, x_5)$. This leads to more charts in the first variant. Analogous to Example B.3.5, the centers of minimal codimension are slightly better in the number of total charts than the centers of codimension two, but the number of leaves are the same.

**Example B.3.8** (Figure B.3, Examples 30–34)**.** In contrast to the previous block of examples, the difference between centers of codimension two and centers of higher codimension becomes more clear for binomials of the type $x_1 x_2 - x_3^3 x_4^4 \cdots x_n^n$, for $n \in \mathbb{Z}_{\geq 4}$. As a consequence the variants with mode $\in \{2, 4\}$ are more efficient for large $n \gg 4$, while mode = 4 is slightly better as the center of codimension three at the beginning provide a fast improvement. On the other hand, the number of charts are larger for the remaining two variants, where the reason for the large numbers if mode = 1 are the same as in Example B.3.7.

| | binomial | max.ord. | codim.2 | min.codim | exc. |
|---|---|---|---|---|---|
| 25. | $x_1^2 - x_2^3x_3^4$ | 6 / 11 | 6 / 11 | 6 / 11 | 6 / 11 |
| 26. | $x_1^2 - x_2^3x_3^4x_4^5$ | 12 / 22 | 12 / 23 | 12 / 22 | 12 / 23 |
| 27. | $x_1^2 - x_2^3x_3^4x_4^5x_5^6$ | 18 / 34 | 15 / 29 | 15 / 28 | 15 / 29 |
| 28. | $x_1^2 - x_2^3x_3^4x_4^5x_5^6 \cdots x_{11}^{12}$ | 288 / 560 | 98 / 195 | 98 / 180 | 98 / 195 |
| 29. | $x_1^2 - x_2^3x_3^4x_4^5x_5^6 \cdots x_{16}^{17}$ | 2.304 / 4.480 | 582 / 1.163 | 582 / 1.036 | 582 / 1.163 |
| 30. | $x_1x_2 - x_3^3x_4^4$ | 8 / 12 | 8 / 15 | 8 / 12 | 8 / 13 |
| 31. | $x_1x_2 - x_3^3x_4^4x_5^5$ | 16 / 24 | 13 / 25 | 16 / 24 | 13 / 22 |
| 32. | $x_1x_2 - x_3^3x_4^4x_5^5x_6^6$ | 28 / 42 | 19 / 37 | 22 / 33 | 19 / 33 |
| 33. | $x_1x_2 - x_3^3x_4^4x_5^5x_6^6 \cdots x_{12}^{12}$ | 512 / 768 | 76 / 151 | 132 / 198 | 76 / 141 |
| 34. | $x_1x_2 - x_3^3x_4^4x_5^5x_6^6 \cdots x_{17}^{17}$ | 4.096 / 6.144 | 151 / 301 | 652 / 978 | 151 / 286 |
| 35. | $x_1^2 - x_2^2x_3$ | 2 / 3 | 2 / 3 | 2 / 3 | 2 / 3 |
| 36. | $x_1^2 - x_2^2x_3^2x_4$ | 3 / 5 | 3 / 5 | 3 / 5 | 3 / 5 |
| 37. | $x_1^2 - x_2^2x_3^2x_4^2x_5^2 \cdots x_{11}^2x_{12}$ | 11 / 21 | 11 / 21 | 11 / 21 | 11 / 21 |
| 38. | $x_1^2 - x_2^2x_3^2x_4^2x_5^2 \cdots x_{16}^2x_{17}$ | 16 / 31 | 16 / 31 | 16 / 31 | 16 / 31 |
| 39. | $x_1x_2^2 - x_3x_4^2$ | 6 / 9 | 2 / 3 | 2 / 3 | 2 / 3 |
| 40. | $x_1^2x_2^3 - x_4x_5^2x_6^2$ | 146 / 264 | 24 / 47 | 22 / 42 | 24 / 47 |
| 41. | $x_1x_2^2x_3^3 - x_4x_5^2x_6^3$ | 385 / 677 | 26 / 51 | 16 / 29 | 16 / 29 |
| 42. | $x_1^2x_2^2x_3^2 - x_4x_5^2x_6^3$ | 1.486 / 2.677 | 154 / 307 | 108 / 191 | 115 / 218 |
| 43. | $x_1^2x_2^3x_3^3 - x_4x_5^2x_6^2x_7^3$ | 18.702 / 34.262 | 126 / 251 | 104 / 196 | 124 / 246 |
| 44. | $x_1x_2^2x_3^3x_4^4 - x_5x_6^2x_7^3x_8^4$ | 107.062 / 196.798 | 260 / 519 | 206 / 371 | 213 / 392 |

Figure B.3.: List of examples (continued).

**Example B.3.9** (Figure B.3, Examples 35–38)**.** There are also types of binomials, for which all variants choose the same centers. For example, for binomials of the form $x_1^2 - x_2^2 \cdots x_{n-1}^2 x_n$, for some $n \in \mathbb{Z}_{\geq 3}$, all centers in the monomialization procedures are of codimension two. If we blowup with center $V(x_1, x_2)$, the total transform of the binomial in the respective charts are $x_1^2(1 - x_2^2 \cdots x_{n-1}^2 x_n)$ and $x_2^2(x_1^2 - x_3^2 \cdots x_{n-1}^2 x_n)$. While the first chart is locally monomial, the second one is of the same form as the original binomial with the difference that $x_2$ does not appear anymore. Hence, the total number of charts is $2(n-1)-1$ and the number of leaves is $n-1$.

**Example B.3.10** (Figure B.3, Examples 39–44)**.** This block of examples consists of homogeneous polynomials. Hence, the variant choosing the centers in the locus of maximal order will first blow-up the closed point, which creates many charts. For increasing degree of the homogeneous binomial, we obtain a fast growing number of charts and leaves. The other three variants are more efficient, where all of them choose first centers of codimension two. In contrast to mode = 1, the centers of larger codimension are more efficient than the codimension two centers towards the end of the monomialization procedure, where the appearing exponents are at most one. This phenomenon has already been observed in Example B.3.5.

**Example B.3.11** (Figure B.4)**.** As we have seen, it may appear that we have to make a choice for the center in the respective variant for monomialization. Let us have a glimpse into the question, how different choices affect the number of charts. Instead of modifying the implementations, we explore this by interchanging the appearing exponents appropriately in a given example.

If mode $\in \{2, 3, 4\}$, the numbers do not change, except for the block of example 41 in Figure B.4. In 41, i.e. $x_1x_2^2x_3^3 - x_4x_5^2x_6^3$ the numbers are larger compared to the other choices 41.2 and 41.3. The reason for this is that the codimension two centers (mode $\in \{2, 4\}$) for example 41 are of the form $V(x_1, x_i)$ or $V(x_j, x_4)$ (for $i \in \{4, 5, 6\}$ and $j \in \{1, 2, 3\}$) at the beginning of the monomialization process. Hence, the improvement of the exponent of $x_i$, resp. $x_j$, is only by one and more blowups are needed. On the other hand, mode = 3 is less affected by this, for example, the first center for example 41, $x_1x_2^2x_3^3 - x_4x_5^2x_6^3$, is $V(x_2, x_5)$. Example 41.3 is slightly better if mode = 3, as the first appearing powers are even.

The first method (via centers contained in the locus of maximal order) varies more if we interchange the exponents. In the cases, where the maximal order is two, the number of charts is significantly larger if we the first exponents are odd. The reason for this can be seen in Example B.3.7, where the binomial became $x_1^2 - x_2x_4x_5^6$.

| | binomial | max.ord. | codim.2 | min.codim | exc. |
|---|---|---|---|---|---|
| 27. | $x_1^2 - x_2^3x_3^4x_4^5x_5^6$ | 18 / 34 | 15 / 29 | 15 / 28 | 15 / 29 |
| 27.2 | $x_1^2 - x_2^5x_3^3x_4^6x_5^4$ | 20 / 38 | 15 / 29 | 15 / 28 | 15 / 29 |
| 27.3 | $x_1^2 - x_2^6x_3^5x_4^4x_5^3$ | 15 / 28 | 15 / 29 | 15 / 28 | 15 / 29 |
| 28. | $x_1^2 - x_2^3x_3^4x_4^5x_5^6 \cdots x_{11}^{12}$ | 288 / 560 | 98 / 195 | 98 / 180 | 98 / 195 |
| 28.2 | $x_1^2 - x_2^{11}x_3^9 \cdots x_6^3x_7^{12}x_8^{10} \cdots x_{11}^4$ | 414 / 812 | 98 / 195 | 98 / 180 | 98 / 195 |
| 28.3 | $x_1^2 - x_2^{12}x_3^{11}x_4^{10}x_5^9 \cdots x_{11}^3$ | 141 / 266 | 98 / 195 | 98 / 180 | 98 / 195 |
| 28.4 | $x_1^2 - x_2^{12}x_3^{10} \cdots x_6^4x_7^{11}x_8^9 \cdots x_{11}^3$ | 114 / 212 | 98 / 195 | 98 / 180 | 98 / 195 |
| 31. | $x_1x_2 - x_3^3x_4^4x_5^5$ | 16 / 24 | 13 / 25 | 16 / 24 | 13 / 22 |
| 31.2 | $x_1x_2 - x_3^3x_4^5x_5^4$ | 20 / 30 | 13 / 25 | 16 / 24 | 13 / 22 |
| 32. | $x_1x_2 - x_3^3x_4^4x_5^5x_6^6$ | 28 / 42 | 19 / 37 | 22 / 33 | 19 / 33 |
| 32.2 | $x_1x_2 - x_3^6x_4^5x_5^4x_6^3$ | 22 / 33 | 19 / 37 | 22 / 33 | 19 / 33 |
| 33. | $x_1x_2 - x_3^3x_4^4x_5^5x_6^6 \cdots x_{12}^{12}$ | 512 / 768 | 76 / 151 | 132 / 198 | 76 / 141 |
| 33.2 | $x_1x_2 - x_3^{12}x_4^{11}x_5^{10} \cdots x_{12}^3$ | 218 / 327 | 76 / 151 | 132 / 198 | 76 / 141 |
| 38. | $x_1^2 - x_2^2x_3^2x_4^2x_5^2 \cdots x_{16}^2x_{17}$ | 16 / 31 | 16 / 31 | 16 / 31 | 16 / 31 |
| 38.2 | $x_1^2 - x_2x_3^2x_4^2x_5^2 \cdots x_{17}^2$ | 16 / 31 | 16 / 31 | 16 / 31 | 16 / 31 |
| 41. | $x_1x_2^2x_3^3 - x_4x_5^2x_6^3$ | 385 / 677 | 26 / 51 | 16 / 29 | 16 / 29 |
| 41.2 | $x_1^3x_2x_3^2 - x_4^3x_5x_6^2$ | 244 / 427 | 12 / 23 | 16 / 29 | 12 / 23 |
| 41.3 | $x_1^2x_2x_3^3 - x_4^2x_5x_6^3$ | 274 / 483 | 19 / 37 | 14 / 25 | 14 / 25 |

Figure B.4.: List of examples, where the different choice within one method are considered.

The next center following our way of choosing the center (for mode = 1), would be $V(x_1, x_2, x_4)$. Hence, we create three new charts and in two of them, we have to blow-up $V(x_1, x_5)$ three times. In contrast to this, if we blowup first in $V(x_1, x_5)$, we would get a smaller number of charts, since we only have to blow-up $V(x_1, x_2, x_4)$ at the end, when we reach $x_1^2 - x_2x_4$ as strict transform.

Note that we interchanged the exponents only at the beginning of the monomialization process. In principle, one could interchange them after each blowup in order to optimize the choice of the center, but we do not address this here.

In conclusion, the approach by blowing up centers contained in the locus of maximal order provides a significant larger number of charts than the other variants if the exponents appearing in the binomial increase. Most of the time, choosing only centers of codimension two (mode = 2) leads to a small number of charts, while a particular structure of the binomial may give a small advantage to the other variants (mode $\in \{3, 4\}$) in some cases. But since the advantage is only small, our choice for a first investigation of a local monomialization of a binomial and the data resulting from it is via centers of codimension two.

## B.4. Comparing the implementation of the generic determinantal resolution with general state of the art algorithms

In this section, we want to compare our implementation of generic determinantal resolution (described in Section A.5.1) with the implementation of Frühbis-Krüger and Pfister [37] of the Algorithm in the computer algebra system SINGULAR and the implementation of Blanco's algorithm of Blanco and Pfister [12]. These implementations can be found in the libraries RESOLVE.LIB and RESBINOMIAL.LIB.

There is also an implementation of Bodnár and Schicho of the algorithm of Villamayor, but [16, page 43] says that in practice, examples for dimension less or equal 4 can be computed, and the dimension of our determinantal singularities are larger. That is the reason why we focus on the implementation of Frühbis-Krüger and Pfister, here. Originally, there is a boundary for the order $b \leq 5$ when calculating the coefficient ideal with the COEFF-method. For testing, we have deleted this boundary in the local Singular installation.

First, we start with comparing the generic_resolve with the general resolve-method of RESOLVE.LIB which is a singular implementation of Villamayor's algorithm. So every calculation has to be done in a ring of characteristic 0. We choose the singular ring with characteristic 0 and the minimal possible number of variables (namely $n \cdot m$ for a $n \times m$ generic matrix) for complexity reasons. In Figure B.5, some examples are given. In the first column, the considered matrix is given. The second column gives the $r \in \mathbb{Z}_{>0}$ that says which ideal of $r$-minors we want to resolve. The following two columns give the running time in seconds of the generic_resolve and the charts in the form final charts/ total charts, and the last two columns give the analogous data for the resolve algorithm.

With the results of Figure B.5 we see, that the high complexity of a general Hironaka resolution is a bottleneck in practice, so we cannot resolve a singularity which is more complex than the determinantal singularity that is generated by the 4-minors of the generic $(4 \times 4)$-matrix.
Our new implementation is constructed for exactly the case of resolving generic determinantal singularities, i.e., we do not deal with the complex data structure of ideals and the high complexity of Gröbner base computations. Therefore, the runtime of all calculations of our new algorithm is much better, i.e., smaller than the RESOLVE.LIB-calculations.

Another interesting point is that for every generic determinantal singularity which is generated by 2-minors of a generic matrix, all center calculation and therefore the whole resolution process is the same, if we start our new algorithm or the algorithm RESOLVE.

If we consider determinantal singularities which are generated by $r$-minors of a generic matrix, where $r > 2$ (and where the dimension of the generic matrix is big enough), our new algorithm considers less charts and final charts then RESOLVE. This belongs to the fact that if we consider minors of size $> 2$, i.e., in Section 6 we have seen that we can apply a change of variables to obtain new variables $y_{i,j} := x_{i,j} - x_{i,1}x_{1,j}$ which reduces the complexity. So we take advantage of this change of variables in the new algorithm. The general implementation RESOLVE.LIB does not use ths change of variables, so the complexity of the singularity seems to be higher than it really is. This change of variables is not an advantage if the size of the considered minors is $\leq 2$ since we have to blow-up at most one time and then we see in each chart, that the singularity is resolved and do not need this change of variables afterwards.

| Matrix | minor size | charts: generic | time: generic | charts: resolve | time: resolve |
|---|---|---|---|---|---|
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix}$ | 1 | 1 / 1 | 0 | 1 / 1 | 0 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix}$ | 2 | 9 / 10 | 0 | 9 / 10 | 3 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix}$ | 3 | 36 / 46 | 0 | 180 / 226 | 27 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \\ x_{4,1} & x_{4,2} \\ x_{5,1} & x_{5,2} \end{pmatrix}$ | 2 | 10 / 11 | 0 | 10 / 11 | 8 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \end{pmatrix}$ | 3 | 72 / 85 | 1 | 504 / 589 | 15482 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{pmatrix}$ | 4 | 576 / 737 | 10 | – / – | killed process |

Figure B.5.: List of examples, where we compare our generic implementation with the resolve.lib w.r.t charts and running time

| Matrix | minor | charts: generic | time: generic | charts: resolve |
|---|---|---|---|---|
| $\begin{pmatrix} 0 & x_{1,2} & x_{1,3} & x_{1,4} \\ -x_{1,2} & 0 & x_{2,3} & x_{2,4} \\ -x_{1,3} & -x_{2,3} & 0 & x_{3,4} \\ -x_{1,4} & -x_{2,4} & -x_{3,4} & 0 \end{pmatrix}$ | 2 | 1 / 1 | 0 | 1 / 1 |
| $\begin{pmatrix} 0 & x_{1,2} & x_{1,3} & x_{1,4} \\ -x_{1,2} & 0 & x_{2,3} & x_{2,4} \\ -x_{1,3} & -x_{2,3} & 0 & x_{3,4} \\ -x_{1,4} & -x_{2,4} & -x_{3,4} & 0 \end{pmatrix}$ | 4 | 6 / 7 | 0 | 6 / 7 |
| skewsymmetric $6 \times 6$ matrix | 4 | 15 / 16 | 0 | killed |
| skewsymmetric $6 \times 6$ matrix | 6 | 90 / 106 | 1 | killed |
| skewsymmetric $8 \times 8$ matrix | 8 | 2520 / 2969 | 30 | killed |

Figure B.6.: List of examples, where we compare our skewsymmetric generic implementation with the resolve.lib w.r.t charts

In Figure B.6, we do not write the running time of resolve, since it is zero or has was killed. We see at the first glance, that we do not have much comparisons. This relies to the fact that we have seen in Section 6, that the $\sqrt{\langle 2\ell\text{-minors of } A_m\rangle} = \sqrt{\langle (2\ell-1)\text{-minors of } A_m\rangle}$. So we have to double the dimension of the matrix (in comparison to the generic case) to get the same number of blow-ups of the minors. The bigger dimension is bad for RESOLVE, since the complexity is bounded by the dimension. Therefore, the complexity in this case is extremly high so we are able to resolve the 2 simplest cases of skewsymmetric generic matrices with RESOLVE, namely the trivial cases of 2-minors and the determinantal singularity generated by the 4 matrix of the skewsymmetric generic matrix of size $(4 \times 4)$.

Figure B.7 shows comparisons for symmetric generic determinantal singularities. We see, that for every symmetric generic determinantal singularity which is generated by 2-minors of a symmetric generic matrix, we need only a single blow-up with our new algorithm and with RESOLVE, but RESOLVE returns less total and final charts.

If we consider symmetric generic determinantal singularities which are generated by $r$-minors, where $r > 2$ (and where the dimension of the generic matrix is big enough), our new algorithm considers less charts and final charts then RESOLVE.

| Matrix | minor | charts: generic | time: generic | charts: resolve | time: resolve |
|---|---|---|---|---|---|
| $\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{1,2} & x_{2,2} \end{pmatrix}$ | 2 | 3 / 4 | 0 | 2 / 3 | 0 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{1,2} & x_{2,2} & x_{2,3} \\ x_{1,3} & x_{2,3} & x_{3,3} \end{pmatrix}$ | 2 | 6 / 7 | 0 | 3 / 4 | 0 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{1,2} & x_{2,2} & x_{2,3} \\ x_{1,3} & x_{2,3} & x_{3,3} \end{pmatrix}$ | 3 | 21 / 28 | 0 | 48 / 73 | 2423 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{1,2} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{1,3} & x_{2,3} & x_{3,3} & x_{3,4} \\ x_{1,4} & x_{2,4} & x_{3,4} & x_{4,4} \end{pmatrix}$ | 2 | 10 / 11 | 0 | 4 / 5 | 1523 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{1,2} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{1,3} & x_{2,3} & x_{3,3} & x_{3,4} \\ x_{1,4} & x_{2,4} & x_{3,4} & x_{4,4} \end{pmatrix}$ | 3 | 64 / 75 | 1 | – / – | killed |
| $\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{1,2} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{1,3} & x_{2,3} & x_{3,3} & x_{3,4} \\ x_{1,4} & x_{2,4} & x_{3,4} & x_{4,4} \end{pmatrix}$ | 4 | 222 / 297 | 5 | – / – | killed |

Figure B.7.: List of examples, where we compare our symmetric generic implementation with the resolve.lib w.r.t charts and running time

This could explained with the same argument, we applied above concerning the change of variables.

The high complexity of a general Hironaka resolution let us not resolve a singularity which is more complex than the determinantal singularity that is generated by the 3-minors of the symetric generic $(4 \times 4)$-matrix. Although the number of charts is less, we see by comparing the runtime of RESOLVE for generic determinantal singularities and for symmetric generic determinantal singularities, that this case and the calculation of each single chart is more complex than the generic case.

Altogether, we see that we can reduce complexity by taking advantage of the determinantal structure, if the considered size of a minor is bigger than 2. If the size of the minors is $\leq 2$, the considered ((skew)-symmetric) generic determinantal singularity is a binomial singularity. We have discussed in the theory sections of this thesis, that it is much easier to resolve binomial ideals than general ideals. Furthermore, the structure of the binomial ideals are very simple for these cases, since the order of such a binomial ideal is at most 2 and we need at most a single blow-up. Therefore RESOLVE is not worse than our new algorithm with respect to the considered number of charts.
If the size of minors is $> 2$, the considered ((skew)-symmetric) generic determinantal singularity is more complex and we can take advantage of the change of coordinates to reduce complexity.
Furthermore, the use of complex data structures like ideals and standard basis decrese complexity, so our new algorithm needs less runtime than RESOLVE.

The next algorithm with which we want to compare our implementation is the BINRESOLVE method of the library RESBINOMIAL.LIB [12], which is a singular implementation of the algorithm of Blanco.
Note, that we only can compare our implementation with the algorithm of Blanco if we resolve binomials, i.e., if we resolve the 2-minors of a generic matrix.

The comparisons in Figure B.8 show that the library RESBINOMIAL.LIB considers more total and final charts and that the runtime is much higher than the considered charts and runtime of our new algorithm.
We do not have more examples here because the process was killed in all other cases of determinantal singularities generated by 2- minors of ((skew)-symmetric) generic matrices.
The author reimplements this library in Section A.3.

| Matrix | charts: generic | time: generic | charts: resbinomial | time: resbinomial |
|---|---|---|---|---|
| $\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix}$ | 4 / 5 | 0 | 28 / 49 | 0 |
| $\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{pmatrix}$ | 6 / 7 | 0 | 264 / 661 | 13 |

Figure B.8.: List of examples, where we compare our generic implementation with the resbinomial.lib w.r.t charts and running time

| ideal | CJS | resbinomial2 | resolve |
|---|---|---|---|
| $z^2 - x^2y^2$ | 3 / 5 | 15 / 26 | 4 / 9 |
| $z^3 - x^2y^2$ | 19 / 34 | 15 / 24 | 70 / 141 |
| $\langle w^2z^2 - x^2y^2, x - y\rangle$ | 16 / 23 | 18 / 29 | 4 / 9 |
| $\langle z^2 - x^4y^4\rangle$ | 5 / 9 | 33 / 62 | 10 / 21 |
| $\langle z^2x^3 - y^4\rangle$ | 47 / 86 | 12 / 19 | 96 / 202 |

Figure B.9.: List of examples, where we compare our implementation of the CJS algorithm with the resbinomial2.lib and resolve.lib w.r.t the number of charts

## B.5. Comparison of CJS, resolve.lib and resbinomial2.lib

In this subsection, we compare the implementations of RESOLVE.LIB and the new implementations RESOLVE2.LIB and the CJS-algorithm.

We see in Figure B.9 that we can find for every of these implementations an example such that every of these implementation is more efficient than the other implementations (with respect to the number of total and final charts).

So one cannot say for surface singularities generated by binomial ideals, that there is the unique best implementation.

In some examples the computation of the TOWER (see Section A.2.2 for more details) yield a less number of charts than the coefficient ideal. Then CJS is more efficient than the other implementations. Sometimes the coefficient ideal construction yield less charts. Then it depend if the implementation of RESOLVE or

| associated Hypersurface | Normal form | CJS | resolve | detresolve |
|---|---|---|---|---|
| $D_4$ | $\begin{pmatrix} y & x & 0 \\ x & y & 0 \\ 0 & 0 & y \end{pmatrix}$ | 2 / 3 | 2 / 3 | 2 / 3 |
| $E_6$ | $\begin{pmatrix} 0 & x & y \\ x & y & 0 \\ y & 0 & xy \end{pmatrix}$ | 2 / 5 | 3 / 5 | 2 / 3 |
| $E_7$ | $\begin{pmatrix} x & 0 \\ 0 & x^2+y^3 \end{pmatrix}$ | 3 / 7 | 3 / 6 | 3 / 5 |
| $\Delta_{1,1}$ | $\begin{pmatrix} w & y & x \\ z & w & y \end{pmatrix}$ | − / − | 2 / 3 | 4 / 5 |

Figure B.10.: List of examples, where we compare our new implementation with resolve.lib w.r.t the number of charts

In Figure B.10, we see that for easy cases of determinantal singularities, there is no outlier and all of the algorithms considers more or less the same number of total and final charts.

the binomial case implementation RESBINOMIAL2 possesses an abbreviation.

## B.6. Comparing the implementation of the resolution of determinantal singularities at most binomial type with general state of the art algorithms

In this section, we give a short list of examples of comparisons of the considered number of charts of RESOLVE, CJS and our algorithm for resolution of determinantal singularities of at most binomial type. We begin with a list of simple square and symmetric matrices in [38, Table 7], [38, Table 10], [38, Table 12] and [38, Table 13].

In Figure B.11, we see that the new algorithm which take advantage of the determinantal structure is not more efficient, if the ideal of $r$-minors is a binomial ideal generated by less binomials than we have entries in the matrix.

This is not surprising since it uses more steps than an algorithm which only resolves the binomial ideal. Here, we maybe first principalize a binomial generator and

| Singularity | resolve | detresolve |
|---|---|---|
| 2-minors of $\begin{pmatrix} x^2 & y^3 \\ xy & z \end{pmatrix}$ | 48 / 102 | 8 / 14 |
| 1-minors of $\begin{pmatrix} x^2 & y^3 \\ xy & z \end{pmatrix}$ | Error | 3 / 4 |
| 2-minors of $\begin{pmatrix} x^3 & -y^4z^4 + x^2 \\ y^4 & z^2 \end{pmatrix}$ | 90 / 180 | 35 / 67 |
| 2-minors of $\begin{pmatrix} x^3 & -y^4z^4 + x^2 \\ y - x & z^2 \end{pmatrix}$ | 10 / 19 | 33 / 62 |
| 2-minors of $\begin{pmatrix} x^3 & -yz + x \\ y - x & z^2 \end{pmatrix}$ | 3 / 6 | 6 / 9 |

Figure B.11.: List of examples, where we compare our new implementation with resolve.lib w.r.t the number of charts

afterwards principalize the monomial ideal generated by the entries via another call of a binomial resolution procedure. An in this step, we need more variables which increases the worst-case complexity.

We see that our new algorithm could be more efficient, if the ideal of the considered minors is not a binomial, anymore.

Altogether, it was not easy to find examples where RESOLVE and DETRESOLVE finish the calculation. Both implementations uses the $\Delta$-operator and the construction of the coefficient ideal, which potentially deal with big numbers and they could enlarge the complexity of the calculation.
Since one could not predict with which input the complexity of constructing a coefficient ideal after some blow-ups is too high, one can not predict if the computation of RESOLVE or of DETRESOLVE ends.
In general, we have seen that DETRESOLVE could be more efficient. But this is only the case if the representation as a matrix is less complex than the representation as an ideal.

## B.7. Complexity remarks on Hu's Algorithm

We refer to the implementation described in Section A.4.

**Remark B.7.1.** Let $C = D_i \uplus D_j$ for some $D_i$ and $D_j$ be a center appearing in the algorithm of Hu. Note that for practical reasons, it is easier to blow-up in $D_i$ and $D_j$ separately than to blow-up in $C$.
The reason is that $I(C)$ has more generators than $I(D_i)$ and $I(D_j)$, and the complexity of the computation of the blow-up increases non-linearly in the number of generators.

**Remark B.7.2** (Worst-case number of blow-ups)**.** The number of blow-ups only depends on the number of Elements $f_j$. We assume we have $f_1, \ldots, f_m$, so we have $m$ such elements. Then we have the index set $I = \mathfrak{P}\{1, \ldots, m\}$ with $2^m$ element. Therefore, the maximum number of the $D_i$ elements is $2^m$. The list $L$ only handles the right order of the blow-ups and does not change the number of elements. If none of the $D_i$ is empty, we have to blow-up in each of these elements, so we need, in the worst case, $2^m$ blow-ups.

In our context, we have these $f_j$ elements in a matrix of dimension $n \times m$, so we have at most $2^{n \cdot m}$ blowups in the first call of Hu's Algorithm. After reduction to dimension $(n-1) \times (m-1)$, we have still (at most) $2^{(n-1)\cdot(m-1)}$ blow-ups.
In total, our applications of Hu's algorithm need at most

$$\sum_{i=0}^{\min\{n,m\}-1} 2^{(n-i)\cdot(m-i)}$$

blow-ups.

# C. Unit Tests

The most important task of unit tests is to ensure that the respective code under test realizes the functionality correctly and completely. The functionality is synonymous with the input and output behaviour of the code under test. In software projects, this functionality is defined by the requirements specification. This is not the basis for our code because the mathematics already supplies the correct results (see [73]).

In this chapter, we will use so-called white-box tests to show that our algorithms work correctly in many cases. Finally, we will try to cover the entire code using test cases to show the correctness of the code.

We follow the procedures of [69] in this chapter.

When testing, one must proceed systematically to cover all classes of possibilities since there are usually an infinite number of input possibilities, but one can only process a finite number of test cases.
A white-box test is a test that deals with the exact appearance of the code. Thus, a concrete test is the one made so that a certain path is traversed in the code. So, using white-box tests, we can cover certain paths of the code and measure the code coverage in percentage to get a measure of the ratio of the tested code. Our goal is to achieve a code coverage of 100%.

Various test cases can be derived from the program flow graph.

Like in the theory chapter, we start with the most easiest cases and go bottom up. First, we begin with trivial inputs, i.e., minors which are not singular. Afterwards, we test the following classes of determinantal singularities:

1. generated by ((skew)-symmetric) generic matrices

2. generated by monomial entries

3. generated by at most binomial entries

## C.1. Test cases for trivial inputs

The first class of inputs we want to test are regular minors of matrices $M$ with at most binomial entries.

The considered test cases use a matrix where the entries could be a unit, zero, variables, monomials and at most binomials. Furthermore, we consider dense and sparse matrices and maximal, minimal and other sizes of the minors.

Table C.1 shows the matrix generating the trivial determinantal singularity, the considered size of minor and the number of final and total charts.

## C.2. Test cases of ((skew-)symmetric) generic determinantal singularities

The following cases are the different classes of generic determinantal singularities. We start with the generic determinantal singularities, followed by the symmetric generic singularities and the skewsymmetric generic singularities.

In Section B.4, we have seen how many charts we consider with our specific implementation for resolution of ((skew)-symmetric) generic determinantal singularities.
In this chapter, we also discussed that the change of variables is an advantage of the specific implementation. Our algorithm for resolution of determinantal singularities of at most binomial type generates a covering with a single chart at this part of the algorithm. That explains the number of extra charts here.

Table C.2 illustrates the test case for generic determinantal singularities in the same format we have seen above.

| matrix | size of minor | final charts | total charts |
|---|---|---|---|
| $M = (1)$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x \\ 1 \end{pmatrix}$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x & 1 \end{pmatrix}$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x & 1 \\ 1 & 1 \end{pmatrix}$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x & 1 \\ 1 & 1 \end{pmatrix}$ | 2 | 1 | 1 |
| $M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 2 \\ 0 & 1 & 1 \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & x & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & y & 0 \\ -x & 0 & 0 \\ 0 & 0 & z \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & y & 0 \\ 0 & 0 & -x & 0 & 0 \\ 0 & 0 & 0 & 0 & z \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & y & 0 \\ 0 & 0 & -x & 0 & 0 \\ 0 & 0 & 0 & 0 & z \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & x & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & y & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & z \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & y & z \\ -y & 0 & u \\ -z & -u & 0 \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & y^2 & z^3 \\ -y^2 & 0 & u^4 \\ -z^3 & -u^4 & 0 \end{pmatrix}$ | 3 | 1 | 1 |

Table C.1.: Test cases of regular minor ideals

| matrix | size of minor | final charts | total charts |
|---|---|---|---|
| $M = (x)$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x & y \\ z & a \end{pmatrix}$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x & y \\ z & a \end{pmatrix}$ | 2 | 4 | 5 |
| $M = \begin{pmatrix} x & y & z \\ a & b & c \end{pmatrix}$ | 2 | 6 | 7 |
| $M = \begin{pmatrix} x & y & z \\ a & b & c \\ d & e & f \end{pmatrix}$ | 2 | 9 | 10 |
| $M = \begin{pmatrix} x & y & z \\ a & b & c \\ d & e & f \end{pmatrix}$ | 3 | 36 | 46 |

Table C.2.: Test cases of generic determinantal singularities

In Table C.3, we can see the test cases for symmetric generic determinantal singularities in the same format we have seen above. Here, we see the additional charts in the last test case. We only have them in this example since all other examples are resolved after a single blow-up, and we do not need to generate a covering.

| matrix | size of minor | final charts | total charts |
|---|---|---|---|
| $M = \begin{pmatrix} x & y \\ y & x \end{pmatrix}$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} x & y \\ y & x \end{pmatrix}$ | 2 | 3 | 4 |
| $M = \begin{pmatrix} x & y & z \\ y & a & b \\ z & b & c \end{pmatrix}$ | 2 | 9 | 10 |
| $M = \begin{pmatrix} x & y & z \\ y & a & b \\ z & b & c \end{pmatrix}$ | 3 | 45 | 55 |

Table C.3.: Test cases of symmetric generic determinantal singularities

Table C.4 illustrates the test cases where the input are skewsymmetric generic determinantal singularities. We consider the trivial singularities, and the 3- and 4-minors of the skewsymmetric generic $(4 \times 4)$-matrix.
Here, we see that our new algorithm can resolve as much singularities as RESOLVE can resolve.

| matrix | size of minor | final charts | total charts |
|---|---|---|---|
| $M = \begin{pmatrix} 0 & x \\ -x & 0 \end{pmatrix}$ | 1 | 1 | 1 |
| $M = \begin{pmatrix} 0 & x \\ -x & 0 \end{pmatrix}$ | 2 | 1 | 1 |
| $M = \begin{pmatrix} 0 & y & z \\ -y & 0 & u \\ -z & -u & 0 \end{pmatrix}$ | 3 | 1 | 1 |
| $M = \begin{pmatrix} 0 & x_{1,2} & x_{1,3} & x_{1,4} \\ -x_{1,2} & 0 & x_{2,3} & x_{2,4} \\ -x_{1,3} & -x_{2,3} & 0 & x_{3,4} \\ -x_{1,4} & -x_{2,4} & -x_{3,4} & 0 \end{pmatrix}$ | 3 | 6 | 7 |
| $M = \begin{pmatrix} 0 & x_{1,2} & x_{1,3} & x_{1,4} \\ -x_{1,2} & 0 & x_{2,3} & x_{2,4} \\ -x_{1,3} & -x_{2,3} & 0 & x_{3,4} \\ -x_{1,4} & -x_{2,4} & -x_{3,4} & 0 \end{pmatrix}$ | 4 | 6 | 7 |

Table C.4.: Test cases of skewsymmetric generic determinantal singularities

Regarding path coverage, we have covered the trivial return path, the easiest cases of the determinantal-monomial case, the gauss-state, the translation state and the blow-up method. It remains to cover the determinantal binomial state.

## C.3. Test cases of determinantal singularities of at most monomial type

In this section, we want to test determinantal singularities of at most monomial type. Since we test singularities of at most binomial type later in this chapter, we consider only minors of size 2.

| matrix | size of minor | final charts | total charts |
|---|---|---|---|
| $M = (x^3y)$ | 1 | 3 | 5 |
| $M = \begin{pmatrix} x^2 & y^3 \\ x & y \end{pmatrix}$ | 2 | 3 | 5 |
| $M = \begin{pmatrix} x^2 & y^3 \\ xy & z^3 \end{pmatrix}$ | 2 | 8 | 14 |

Table C.5.: Test cases of determinantal singularities of at most monomial type

Table C.5 illustrates the test cases for determinantal singularities of at most monomial type in the same format as we have seen above.

## C.4. Test cases of determinantal singularities of at most binomial type

In this section, we want to tests determinantal singularities of at most binomial type. Since we already discussed the high worst case complexity and since we have already covered the gauss-state in the path by other test cases, we restrict the test cases here to 2-minors.

| matrix | size of minor | final charts | total charts |
|---|---|---|---|
| $M = (x^2 - y^2z^2)$ | 1 | 15 | 26 |
| $M = \begin{pmatrix} -y^2z^2 + x^2 & x^3 \\ y^4 & z^2 \end{pmatrix}$ | 2 | 34 | 65 |
| $M = \begin{pmatrix} x^3 & -y^4z^4 + x^2 \\ y^4 & z^2 \end{pmatrix}$ | 2 | 35 | 67 |
| $M = \begin{pmatrix} x^3 & -y^4z^4 + x^2 \\ x - y & z^2 \end{pmatrix}$ | 2 | 33 | 62 |

Table C.6.: Test cases of determinantal singularities of at most binomial type

Table C.6 illustrates the test cases for determinantal singularities of at most binomial type.

Altogether, we reach a high path coverage with these test cases.

# List of Figures

# List of Algorithms

# List of Tables

# Bibliography


[1] S. Abhyankar. *Resolution of singularities of embedded algebraic surfaces*, volume 24. Pure and Applied Mathematics, 1966.

[2] S. Abhyankar. Good points of a hypersurface. *Advances in Math. 68*, pages 87–256, 1988.

[3] D. Abramovich and B. Schober. Desingularization of binomial varieties using toric artin stacks. 2022. `arXiv:arXiv:2208.08951`.

[4] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2006. URL: `https://theory.cs.princeton.edu/complexity/book.pdf`.

[5] E. Bierstone, D. Grigoriev, P. D. Milman, and J. Włodarczyk. Effective hironaka resolution and its complexity (with appendix on applications in positive characteristic). 2012. `arXiv:arXiv:1206.3090`.

[6] E. Bierstone and P. D. Milman. A simple constructive proof of canonical resolution of singularities. *Effective methods in algebraic geometry (Castiglioncello, 1990)*, 94:11–30, 1991.

[7] E. Bierstone and P. D. Milman. Canonical desingularization in characteristic zero by blowing up the maximum strata of a local invariant. *Invent. Math.* **128(2)**, pages 207–302, 1997.

[8] E. Bierstone and P. D. Milman. Desingularization of toric and binomial varieties. *J. Algebraic Geom.* **15**, 3:443 – 486, 2006.

[9] R. Blanco. Desingularization of binomial varieties in arbitrary characteristic. part i. a new resolution function and their properties. *Math. Nachrichten* **285**, 11-12:1316–1342, 2012.

[10] R. Blanco. Desingularization of binomial varieties in arbitrary characteristic. part ii: Combinatorial desingularization algorithm. *Q. J. Math.* **63**, 4:771–794, 2012.

[11] R. Blanco and S. Encinas. Embedded desingularization of toric varieties. *Journal of Symbolic Computation, 46(11)*, pages 1229–1241, 2011. doi:10.1016/j.jsc.2011.08.005.

[12] R. Blanco and G. Pfister. resbinomial.lib. *A Singular library for resolution of binomial ideals*, 2010. URL: https://github.com/Singular/Singular/blob/spielwiese/Singular/LIB/resbinomial.lib.

[13] G. Bodnár and J. Schicho. A computer program for the resolution of singularities. 01 2000. doi:10.1007/978-3-0348-8399-37.

[14] J. Boehm, W. Decker, A. Fruehbis-Krueger, F.-J. Pfreundt, M. Rahn, and L. Ristau. Towards massively parallel computations in algebraic geometry. *Algebraic Geometry. Found Comput Math 21*, page 767–806, 2021. URL: https://doi.org/10.1007/s10208-020-09464-x.

[15] J. Böhm and A. Frühbis-Krüger. A smoothness test for highercodimension. *J. Symbol. Comp. 86*, pages 153–165, 2018.

[16] A. Bravo, S. Encinas, and O. Villamayor. A simplified proof of desingularisation and applications. *Rev. Math. Iberoamericana 21*, pages 349–458, 2005.

[17] J. Bruce, V. Goryunov, and G. Haslinger. Families of skew-symmetric matrices of even size. 2022. URL: https://arxiv.org/abs/2206.00596.

[18] J. Bruce and U. Vetter. Determinantial rings. *Lecture Notes in Mathematics Vol. 1327*, 1988.

[19] A. Conca. Gröbner bases of minors of a symmetric matrix. *J. Algebra 166*, pages 406 – 421, 1994.

[20] V. Cossart, U. Jannsen, and S. Saito. *Desingularization: Invariants and Strategy: Application to Dimension 2.* Lecture Notes in Mathematics. Springer International Publishing, 2020.

[21] V. Cossart, U. Jannsen, and B. Schober. Invariance of hironaka's characteristic polyhedron. *Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. RACSAM,* **113**, 4:4145–4169, 2019.

[22] V. Cossart and O. Piltant. Resolution of singularities of threefolds in positive characteristic i. *Journal of Algebra*, 320(7):1051–1082, 2008.

[23] V. Cossart and O. Piltant. Resolution of singularities of arithmetical threefolds. *Journal of Algebra*, 2014.

[24] V. Cossart and O. Piltant. Resolution of singularities of threefolds in mixed characteristic: case of small multiplicity. *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, 108:113–151, 2014.

[25] V. Cossart and B. Schober. A strictly decreasing invariant for resolution of singularities in dimension two. *Publ. Res. Inst. Math. Sci.*, 56:217–280, 2020.

[26] S. D. Cutkosky. *Resolution of Singularities*. Graduate studies in mathematics. American Mathematical Soc., 2004.

[27] S. D. Cutkosky. Resolution of singularities for 3-folds in positive characteristic. *Amer. J. Math.,131(1)*, pages 59–127, 2009.

[28] S. D. Cutkosky. A skeleton key to abhyankar's proof of embedded resolution of characteristic p surfaces. *Asian J. Math., 15(3)*, pages 369–416, 2011.

[29] W. Decker, C. Fieker, M. Horn, and M. Joswig. OSCAR *open source computer algebra research system for computations in algebra, geometry, and number theory*. 2023. URL: `https://oscar.computeralgebra.de/`.

[30] W. Decker, G.-M. Greuel, G. Pfister, and H. Schoenemann. *SINGULAR 4-1-1, A computer algebra system for polynomial computations*. 2018. URL: `http://www.singular.uni-kl.de`.

[31] M. du Sautoy and F. Grunewald. Analytic properties of zeta functions and subgroup growth. *Annals of Mathematics Second Series, Vol. 152, No. 3*.

[32] D. Eisenbud and B. Sturmfels. Binomial ideals. *Duke mathematical journal*, 84(1):1–45, 1996. `doi:10.1215/S0012-7094-96-08401-X`.

[33] S. Encinas and H. Hauser. Strong resolution of singularities in characteristic zero. *Comment. Math. Helv., 77(4)*, pages 821–845, 2002.

[34] S. Encinas and O. Villamayor. *A Course on Constructive Desingularization and Equivariance*. 2000. `doi:10.1007/978-3-0348-8399-3_6`.

[35] A. Fruehbis-Krueger, L. Ristau, and B. Schober. Embedded desingularization for arithmetic surfaces - toward a parallel implementation. Math. Comp. 90 (2021):1957–1997, 2021. URL: `https://doi.org/10.1090/mcom/3624`.

[36] A. Frühbis-Krüger. Computational aspects of singularities. *J.-P. Brasselet, J.Damon et al.: Singularities in Geometry and Topology*, pages 253–327, 2007.

[37] A. Frühbis-Krüger and G. Pfister. resolve.lib. *A Singular library for resolution of singularities*, 2004. URL: `https://github.com/Singular/Singular/blob/spielwiese/Singular/LIB/resolve.lib`.

[38] A. Frühbis-Krüger and M. Zach. Determinantal singularities. 2021. URL: `https://arxiv.org/abs/2106.04855`.

[39] T. Gaffney and M. Molino. Symmetric determinantal singularities i: The multiplicity of the polar curve. 2020. URL: `https://arxiv.org/abs/2003.12543`.

[40] T. Gaffney and M. Molino. Symmetric determinantal singularities ii: Equisingularity and seids. 2021. URL: `https://arxiv.org/abs/2103.03195`.

[41] S. A. Gaube. Algorithmen zum ort maximaler ordnung. Master thesis, Leibniz Universität Hannover, 2018.

[42] S. A. Gaube and B. Schober. Algorithmic local monomialization of a binomial: A comparison of different approaches. *International Journal of Algebra and Computation*, 33:161–195, 2023. `doi:10.1142/S0218196723500108`.

[43] J. Giraud. Contact maximal en caractéristique positive. *Ann. Sci. Ec. Norm. Sup. 4ème série 8*, pages 201–234, 1975.

[44] P. D. González Perez and B. Teissier. Embedded resolution of non necessaryily normal affine toric varieties. *C. R. Math. Acad. Sci. Paris, 334(5)*, pages 379–382, 2002.

[45] R. A. Goward. A simple algorithm for principalization of monomial ideals. *Trans. Amer. Math. Soc. 357(12):4805-4812*, 2005.

[46] G.-M. Greuel and G. Pfister. *A Singular Introduction to Commutative Algebra*. Springer Science, Business Media, Berlin Heidelberg, 2nd edition, 2012.

[47] A. Grzegorczyk. Some classes of recursive functions. *Rozprawy Matematiczne*, pages 1–44, 1953.

[48] J. Harris. *Algebraic Geometry – A First Course*. Graduate Texts in Mathematics, Springer, 1992.

[49] R. Hartshorne. *Algebraic geometry*. Springer-Verlag, New York, 1977. Graduate Texts in Mathematics, No. 52.

[50] H. Hauser. Seventeen obstacles for resolution of singularities. *The Brieskorn anniversary (ed. V.I. Arnold, G.-M. Greuel, J. Steenbrink). Progress in Math. 162*, 1997.

[51] H. Hauser. The hironaka theorem on resolution of singularities (or: A proof we always wanted to understand). *Bull. Amer. Math. Soc., 40(3)*, pages 323–403, 2003.

[52] H. Hauser. Why the characteristic zero proof of resolution fails in positive characteristic. 2003. URL: `https://homepage.univie.ac.at/herwig.hauser/Publications/characteristic-p.pdf`.

[53] J. Herzog and N. V. Trung. Gröbner bases and multiplicity of determinantal and pfaffian ideals. 96, Issue 1:1–37, 1992.

[54] H. Hironaka. Resolution of singularities of an algebraic variety over a field of chararteristic zero I, II. *Ann. of Math. (2)* **79**, pages 109–203, 205–326, 1964.

[55] H. Hironaka. Characteristic polyhedra of singularities. *J. Math. Kyoto Uni.* **7**, pages 251–293, 1967.

[56] H. Hironaka. Idealistic exponents of singularity. *Algebraic geometry, John Hopkins Univ. Press*, pages 52–125, 1977.

[57] H. Hironaka. Desingularization of excellent surfaces. *Adv. Sci. Sem. in Alg. Geo, Bowdoin College, Summer 1967, Notes by Bruce Bennett, appendix of* Resolution of surface singularities, *Lecture Notes in Mathematics 1101*, 1984.

[58] H. Hironaka. Theory of infinitely near singular points. *J. Korean Math Soc 40*, pages 901–920, 2003.

[59] H. Hironaka. Three key theorems of infinitely near singularities. *Singularités Franco-Japonaises, Semin. Congr. 10, Soc. Math. France, Paris*, pages 87–126, 2005.

[60] Y. Hu. A compactification of open varieties. *Trans. Amer. Math. Soc.* **355**, 12:4737–4753, 2003.

[61] E. L. Kayser. Gröbner bases and their complexity. Master's thesis, Leibniz Universität Hannover, 2022. URL: `https://www.thi.uni-hannover.de/`

fileadmin/thi/abschlussarbeiten/2022/Kayser_-_Groebner_Bases_and_Their_Complexity_Website.pdf.

[62] V. Kodiyalam, T. Y. Lam, and R. G. Swan. Determinantal ideals, pfaffian ideals, and the principal minor theorem. *Noncommutative rings, group rings, diagram algebras and their applications*, 456:35–60, 2008.

[63] J. Kollár. Lectures on resolution of singularities. *Annals of Mathematics Studies, 166, Princeton University Press*, 2007.

[64] L. Li. Wonderful compactification of an arrangement of subvarieties. Michigan Math. J. **58**, 2:535–563, 2009.

[65] J. Lipman. Desingularization of two-dimensional schemes. *Annals of Mathematics*, 107(2):151–207, 1978.

[66] E. Miller and B. Sturmfels. *Combinatorial commutative algebra*, volume 227 of *Graduate texts in mathematics*. Springer, 2005. doi:10.1007/b138602.

[67] R. Narasimhan. Monomial equimultiple curves in positive characteristic. *Proc. Amer. Math. Soc. 89*, 3:402–406, 1983.

[68] F.-J. Pfreundt, M. Rahn, and et al. *GPI-space*. Fraunhofer ITWM Kaiserslautern. URL: http://www.gpi-space.de/.

[69] K. Schneider. *Abenteuer Softwarequalität - Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement*. dpunkt.verlag, Heidelberg, 2. auflage edition, 2012.

[70] B. Schober. Characteristic polyhedra of idealistic exponents with history. Dissertation, Universität Regensburg, 2013. URL: https://epub.uni-regensburg.de/28877/.

[71] B. Schober. Partial local resolution by characteristic zero methods. *Results Math 73, 48*, 2018. URL: https://doi.org/10.1007/s00025-018-0808-7.

[72] B. Schober. Singularities and polyhedra. Habilitationsschrift, Carl von Ossietzky Universität Oldenburg, 2021.

[73] A. Spillner and T. Linz. *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester; Foundation Level nach ISTQB-Standard*. dpunkt-Verlag, 2005.

[74] A. Ulyanov. A polydiagonal compactification of configuration spaces. *J. Algebraic Geom.* **11**, pages 129–159, 2002.

[75] I. Vainsencher. Complete collineations and blowing up determinantal ideals. *Math. Ann.* **267**, 3:417–432, 1984.

[76] O. Villamayor. Constructiveness of hironaka's resolution. *Ann. Sci. cole Norm. Sup. (4) 22(1)*, pages 1–32, 1989.

[77] O. Villamayor. Patching local uniformizations. *Ann. Sci. cole Norm. Sup. (4) 25(6)*, pages 629–677, 1992.

[78] R. J. Walker. Reduction of the singularities of an algebraic surface. *Annals of Mathematics*, 36(2):336–365, 1935. URL: `http://www.jstor.org/stable/1968575`.

[79] J. Włodarczyk. Hironaka resolution in characteristic zero. *J. Amer. Math. Soc., 18(4)*, pages 779–822, 2005.

[80] O. Zariski. The reduction of the singularities of an algebraic surface. *Annals of Mathematics*, 40(3):639–689, 1939. URL: `http://www.jstor.org/stable/1968949`.

[81] O. Zariski. The compactness of the riemann manifold of an abstract field of algebraic functions. *Bulletin of the American Mathematical Society*, 50(10):683 – 691, 1944.

[82] O. Zariski. The concept of a simple point of an abstract algebraic variety. *Transactions of the American Mathematical Society*, 62:1–52, 1947. `doi:10.2307/1990628`.

[83] O. Zariski and P. Samuel. *Commutative algebra, Volumes I and II*. Springer Science, 1958 and 1960.