Computing jumping numbers in higher dimensions

Hans Baumers and Ferran Dachs Cadefau

KU Leuven

22 June 2016
EACA, Logroño

Introduction

What is algebraic geometry?

Introduction

What is algebraic geometry?

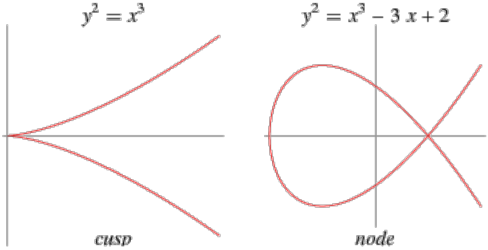
- Consider a polynomial $f \in \mathbb{C}[x_1, \ldots, x_n]$.

Introduction

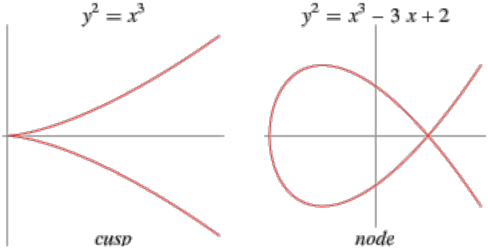
What is algebraic geometry?

- Consider a polynomial $f \in \mathbb{C}[x_1, \dots, x_n]$.
- Study the properties of $\{x \in \mathbb{C}^n \mid f(x) = 0\}$ (=geometry) using algebraic methods.

Introduction



Introduction



Which is the worst?

Introduction

- The jumping numbers of the node are 1, 2, 3...

Introduction

- The jumping numbers of the node are 1, 2, 3...
- The jumping numbers of the cusp are $\frac{5}{6}$, 1, $\frac{11}{6}$, 2...

Introduction

- The jumping numbers of the node are 1, 2, 3...
- The jumping numbers of the cusp are $\frac{5}{6}$, 1, $\frac{11}{6}$, 2...
- The cusp is worse!

Setting

- We consider a smooth complex algebraic variety X of arbitrary dimension, and a hypersurface (= effective divisor) $D \subset X$.

Setting

- We consider a smooth complex algebraic variety X of arbitrary dimension, and a hypersurface (= effective divisor) $D \subset X$.
- We study the singularities of D.

Setting

- We consider a smooth complex algebraic variety X of arbitrary dimension, and a hypersurface (= effective divisor) $D \subset X$.
- We study the singularities of D.
- Think of $X = \mathbb{C}^n$, and $D = \{x \in \mathbb{C}^n \mid f(x) = 0\}$ for some $f \in \mathbb{C}[x_1, \ldots, x_n]$.

Definitions and first properties

Definition

A **log resolution** of D is a birational morphism $\pi : Y \to X$ such that

- Y is smooth,
- π is an isomorphism over $X \backslash \mathrm{Sing}(D)$,
- the pullback $\pi^* D$ is a snc divisor, i.e., the irreducible components of $\pi^{-1}(D)$ are smooth and intersect transversally.

Definition

A **log resolution** of D is a birational morphism $\pi : Y \to X$ such that

- Y is smooth,
- π is an isomorphism over $X \setminus \mathrm{Sing}(D)$,
- the pullback $\pi^* D$ is a snc divisor, i.e., the irreducible components of $\pi^{-1}(D)$ are smooth and intersect transversally.

A log resolution ‘makes D smooth’.

Definition

If $\pi : Y \to X$ is a morhpism of smooth algebraic varieties, then the **relative canonical divisor** associated to π is the divisor

$$K_\pi = K_Y - \pi^* K_X .$$

Definition

If $\pi : Y \to X$ is a morhpism of smooth algebraic varieties, then the **relative canonical divisor** associated to π is the divisor

$$K_\pi = K_Y - \pi^* K_X.$$

The relative canonical divisors measures the difference between Y and X. Its class has a unique representative supported on $\mathrm{Exc}(\pi)$.

Definition

If $\pi : Y \to X$ is a morhpism of smooth algebraic varieties, then the **relative canonical divisor** associated to π is the divisor

$$K_\pi = K_Y - \pi^* K_X.$$

The relative canonical divisors measures the difference between Y and X. Its class has a unique representative supported on $\mathrm{Exc}(\pi)$.

Example

If $\pi : Y \to X$ is a blow-up of a subvariety V of codimension $r \leq 2$ with exceptional divisor E (='replacing V by a hyperplane E') then $K_\pi = (r-1)E$.

Multiplier ideals

Definition

If $\pi : Y \to X$ is a log resolution of a hypersurface $D \subset X$, and $\lambda \in \mathbb{Q}_{\geq 0}$, then the **multiplier ideal** of (X, D) with coefficient λ is

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor),$$

where we denote $\lfloor \lambda \pi^* D \rfloor = \sum \lfloor \lambda a_i \rfloor E_i$ if $\pi^* D = \sum a_i E_i$.

First properties

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$$

First properties

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$$

- The multiplier ideals do not depend on the log resolution,

First properties

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$$

- The multiplier ideals do not depend on the log resolution,
- $\mathcal{J}(\lambda D) \subseteq \mathcal{O}_X$ for all λ,

First properties

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$$

- The multiplier ideals do not depend on the log resolution,
- $\mathcal{J}(\lambda D) \subseteq \mathcal{O}_X$ for all λ,
- $\mathcal{J}(\lambda D) = \mathcal{O}_X$ if $\lambda \ll 1$,

First properties

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$$

- The multiplier ideals do not depend on the log resolution,
- $\mathcal{J}(\lambda D) \subseteq \mathcal{O}_X$ for all λ,
- $\mathcal{J}(\lambda D) = \mathcal{O}_X$ if $\lambda \ll 1$,
- if $\lambda < \lambda'$, then $\mathcal{J}(\lambda D) \supseteq \mathcal{J}(\lambda' D)$,

First properties

$$\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$$

- The multiplier ideals do not depend on the log resolution,
- $\mathcal{J}(\lambda D) \subseteq \mathcal{O}_X$ for all λ,
- $\mathcal{J}(\lambda D) = \mathcal{O}_X$ if $\lambda \ll 1$,
- if $\lambda < \lambda'$, then $\mathcal{J}(\lambda D) \supseteq \mathcal{J}(\lambda' D)$,
- if $0 < \varepsilon \ll 1$, then $\mathcal{J}((\lambda + \varepsilon)D) = \mathcal{J}(\lambda D)$.

Jumping numbers

So there is a chain of rational numbers

$$0 = \lambda_0 < \lambda_1 < \cdots < \lambda_i < \lambda_{i+1} < \dots$$

such that

Jumping numbers

So there is a chain of rational numbers

$$0 = \lambda_0 < \lambda_1 < \cdots < \lambda_i < \lambda_{i+1} < \dots$$

such that

- $\mathcal{J}(\lambda_i D) \supsetneq \mathcal{J}(\lambda_{i+1} D)$ for all i,

Jumping numbers

So there is a chain of rational numbers

$$0 = \lambda_0 < \lambda_1 < \cdots < \lambda_i < \lambda_{i+1} < \ldots$$

such that

- $\mathcal{J}(\lambda_i D) \supsetneq \mathcal{J}(\lambda_{i+1} D)$ for all i,
- if $\lambda \in [\lambda_i, \lambda_{i+1})$ for some i, then $\mathcal{J}(\lambda D) = \mathcal{J}(\lambda_i)$.

Jumping numbers

So there is a chain of rational numbers

$$0 = \lambda_0 < \lambda_1 < \cdots < \lambda_i < \lambda_{i+1} < \dots$$

such that

- $\mathcal{J}(\lambda_i D) \supsetneq \mathcal{J}(\lambda_{i+1} D)$ for all i,
- if $\lambda \in [\lambda_i, \lambda_{i+1})$ for some i, then $\mathcal{J}(\lambda D) = \mathcal{J}(\lambda_i)$.

The numbers λ_i, $i > 0$, are called the **jumping numbers** of the pair (X, D).

Theorem (Skoda's theorem)

If $\lambda > 1$, then λ is a jumping number if and only if $\lambda - 1$ is a jumping number.

Theorem (Skoda's theorem)

If $\lambda > 1$, then λ is a jumping number if and only if $\lambda - 1$ is a jumping number.

Hence it suffices to compute the jumping numbers in the interval $(0, 1]$.

Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_X(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.
If $\pi : Y \to X$ is a log resolution, and we denote

$$K_\pi = \sum k_i E_i$$

and

$$\pi^* D = \sum a_i E_i,$$

Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_X(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.
If $\pi : Y \to X$ is a log resolution, and we denote

$$K_\pi = \sum k_i E_i$$

and

$$\pi^* D = \sum a_i E_i,$$

then the jumping numbers are contained in the set

$$\left\{ \frac{k_i + m}{a_i} \middle| m \in \mathbb{Z}_{>0} \right\}.$$

Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_X(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.
If $\pi : Y \to X$ is a log resolution, and we denote

$$K_\pi = \sum k_i E_i$$

and

$$\pi^* D = \sum a_i E_i,$$

then the jumping numbers are contained in the set

$$\left\{ \frac{k_i + m}{a_i} \middle| m \in \mathbb{Z}_{>0} \right\}.$$

The numbers in this set are the **candidate jumping numbers**.
The smallest candidate is always a jumping number, it is the **log canonical threshold**, denoted $\mathrm{lct}(D)$.

The algorithm

- Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.

- Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.
- **Key**: Study when $\pi_* \mathcal{O}_Y(-D_1) = \pi_* \mathcal{O}_Y(-D_2)$ for two divisors D_1 and D_2, i.e., when they are **equivalent**.

- Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.
- **Key**: Study when $\pi_* \mathcal{O}_Y(-D_1) = \pi_* \mathcal{O}_Y(-D_2)$ for two divisors D_1 and D_2, i.e., when they are **equivalent**.
- We want to find the biggest divisor D_λ equivalent to $\lfloor \lambda \pi^* D \rfloor - K_\pi$.

- Recall: $\mathcal{J}(\lambda D) = \pi_* \mathcal{O}_Y(K_\pi - \lfloor \lambda \pi^* D \rfloor)$.
- **Key**: Study when $\pi_* \mathcal{O}_Y(-D_1) = \pi_* \mathcal{O}_Y(-D_2)$ for two divisors D_1 and D_2, i.e., when they are **equivalent**.
- We want to find the biggest divisor D_λ equivalent to $\lfloor \lambda \pi^* D \rfloor - K_\pi$.
- Then the jumping number following to λ is the smallest number λ' such that $\lfloor \lambda' \pi^* D \rfloor - K_\pi$ has a coefficient bigger than the one in D_λ.

π-antieffective divisors

Definition

A divisor D on Y is called π-antieffective if $-D|_E$ is an effective divisor for every π-exceptional divisor E.

π-antieffective divisors

Definition

A divisor D on Y is called π-antieffective if $-D|_E$ is an effective divisor for every π-exceptional divisor E.

Algorithm (Unloading procedure; B., Dachs Cadefau, after Laufer, Enriques, Reguera, AAD. . .)

`Input`: A divisor D on Y.
`Output`: The smallest π-antieffective divisor bigger than D.
While $-D|_E$ is not effective for some E, replace D by $D + E$.

π-antieffective divisors

Definition

A divisor D on Y is called π-antieffective if $-D|_E$ is an effective divisor for every π-exceptional divisor E.

Algorithm (Unloading procedure; B., Dachs Cadefau, after Laufer, Enriques, Reguera, AAD. . .)

`Input`: A divisor D on Y.

`Output`: The smallest π-antieffective divisor bigger than D.

While $-D|_E$ is not effective for some E, replace D by $D + E$.

The algorithm always stops. The result is the π-**antieffective closure** of D.

π-antieffective divisors

Theorem (B., Dachs Cadefau)

If $\tilde{D}$ is the π-antieffective closure of D, then

$$\pi_*\mathcal{O}_Y(-\tilde{D}) = \pi_*\mathcal{O}_Y(-D).$$

π-antieffective divisors

Theorem (B., Dachs Cadefau)

If $\tilde{D}$ is the π-antieffective closure of D, then

$$\pi_* \mathcal{O}_Y(-\tilde{D}) = \pi_* \mathcal{O}_Y(-D).$$

So if λ' is the smallest jumping number bigger than λ, $\lfloor \lambda' \pi^* D \rfloor - K_\pi$ must exceed the π-antieffective closure of $\lfloor \lambda \pi^* D \rfloor - K_\pi$ in at least one coefficient.

π-antieffective divisors

Theorem (B., Dachs Cadefau)

If $\tilde{D}$ is the π-antieffective closure of D, then

$$\pi_*\mathcal{O}_Y(-\tilde{D}) = \pi_*\mathcal{O}_Y(-D).$$

So if λ' is the smallest jumping number bigger than λ, $\lfloor \lambda' \pi^* D \rfloor - K_\pi$ must exceed the π-antieffective closure of $\lfloor \lambda \pi^* D \rfloor - K_\pi$ in at least one coefficient.

$\Rightarrow$ Lower bound for the next jumping number!

The two-dimensional case

Theorem (Lipman, 1969)

Suppose X is a surface. The mapping $D \mapsto \pi_* \mathcal{O}_Y(-D)$ determines a one-to-one correspondence between

1. *π-antieffective divisors on Y, and*
2. *complete ideal sheaves on X.*

The two-dimensional case

Theorem (Lipman, 1969)

Suppose X is a surface. The mapping $D \mapsto \pi_* \mathcal{O}_Y(-D)$ determines a one-to-one correspondence between

1. *π-antieffective divisors on Y, and*
2. *complete ideal sheaves on X.*

Theorem (Alberich-Carramiñana, Àlvarez-Montaner, Dachs Cadefau)

If $D_\lambda = \sum a_i^\lambda E_i$ is the π-antieffective closure of $\lfloor \lambda \pi^* D \rfloor - K_\pi$, then the smallest jumping number bigger then λ is

$$\lambda' = \min \left\{ \frac{k_i + a_i^\lambda + 1}{a_i} \right\},$$

where $K_\pi = \sum k_i E_i$ and $\pi^* D = \sum a_i E_i$.

The two-dimensional case

Algorithm (Alberich-Carramiñana, Àlvarez-Montaner, Dachs Cadefau)

`Input`: A curve C on X.

`Output`: The jumping numbers of (X, C).

The two-dimensional case

Algorithm (Alberich-Carramiñana, Àlvarez-Montaner, Dachs Cadefau)

`Input`: A curve C on X.

`Output`: The jumping numbers of (X, C).

1. Compute the minimal log resolution $\pi : Y \to X$ of C.

The two-dimensional case

Algorithm (Alberich-Carramiñana, Àlvarez-Montaner, Dachs Cadefau)

`Input`: A curve C on X.

`Output`: The jumping numbers of (X, C).

1. Compute the minimal log resolution $\pi : Y \to X$ of C.
2. Compute $\operatorname{lct}(C)$.

The two-dimensional case

Algorithm (Alberich-Carramiñana, Àlvarez-Montaner, Dachs Cadefau)

`Input:` A curve C on X.

`Output:` The jumping numbers of (X, C).

1. Compute the minimal log resolution $\pi : Y \to X$ of C.
2. Compute $\mathrm{lct}(C)$.
3. Having computed a jumping number λ_j, compute the π-antieffective closure $D_\lambda = \sum a_i^{\lambda_j} E_i$ of $\lfloor \lambda_j \pi^* C \rfloor - K_\pi$ using the unloading procedure. Then the next jumping number is

$$\lambda_{j+1} = \min \left\{ \frac{k_i + a_i^{\lambda_j} + 1}{a_i} \right\}.$$

The two-dimensional case

Algorithm (Alberich-Carramiñana, Àlvarez-Montaner, Dachs Cadefau)

`Input`: A curve C on X.

`Output`: The jumping numbers of (X, C).

1. Compute the minimal log resolution $\pi : Y \to X$ of C.
2. Compute $\mathrm{lct}(C)$.
3. Having computed a jumping number λ_j, compute the π-antieffective closure $D_\lambda = \sum a_i^{\lambda_j} E_i$ of $\lfloor \lambda_j \pi^* C \rfloor - K_\pi$ using the unloading procedure. Then the next jumping number is

$$\lambda_{j+1} = \min \left\{ \frac{k_i + a_i^{\lambda_j} + 1}{a_i} \right\}.$$

4. Repeat until we arrive at 1.

The higher dimensional case

Theorem (B., Dachs Cadefau)

If we run the algorithm of [AAD] until we arrive at 1, then we get a set of so-called **supercandidates**, *containing all the jumping numbers.*

The higher dimensional case

Theorem (B., Dachs Cadefau)

If we run the algorithm of [AAD] until we arrive at 1, then we get a set of so-called **supercandidates***, containing all the jumping numbers.*

Remark

No examples are known where not all supercandidates are jumping numbers!

Checking supercandidates

Definition

The *minimal jumping divisor* of a supercandidate λ' is the reduced divisor $G_{\lambda'}$ supported on those E_i where the minimum

$$\lambda' = \min \left\{ \frac{k_i + a_i^{\lambda} + 1}{a_i} \right\}$$

is achieved.

Checking supercandidates

Definition

The *minimal jumping divisor* of a supercandidate λ' is the reduced divisor $G_{\lambda'}$ supported on those E_i where the minimum

$$\lambda' = \min \left\{ \frac{k_i + a_i^{\lambda} + 1}{a_i} \right\}$$

is achieved.

Proposition (B., Dachs Cadefau)

If the minimum jumping divisor of a supercandidate λ' has an irreducible connected component, then λ' is a jumping number.

Checking supercandidates

Definition

The *minimal jumping divisor* of a supercandidate λ' is the reduced divisor $G_{\lambda'}$ supported on those E_i where the minimum

$$\lambda' = \min \left\{ \frac{k_i + a_i^\lambda + 1}{a_i} \right\}$$

is achieved.

Proposition (B., Dachs Cadefau)

If the minimum jumping divisor of a supercandidate λ' has an irreducible connected component, then λ' is a jumping number.

Good news: this happens very often!

Checking supercandidates

Otherwise, there are some more techniques to check whether a supercandidate is a jumping number.

Examples

Example 1

- Take $X = \mathbb{C}^3$ and D given by

$$x(yz - x^4)(x^4 + y^2 - 2yz) + yz^4 - y^5 = 0.$$

Example 1

- Take $X = \mathbb{C}^3$ and D given by

$$x(yz - x^4)(x^4 + y^2 - 2yz) + yz^4 - y^5 = 0.$$

- We find a resolution $\pi : Y \to X$ by six point blow-ups, with

$$K_\pi = 2E_1 + 4E_2 + 8E_3 + 14E_4 + 6E_5 + 6E_6$$

and

$$\pi^* D = D_{aff} + 5E_1 + 9E_2 + 16E_3 + 27E_4 + 11E_5 + 11E_6.$$

Example 1

λ	G_λ
$\frac{5}{9}$	$E_2 + E_4$
$\frac{2}{3}$	$E_2 + E_4$
$\frac{20}{27}$	E_4
$\frac{7}{9}$	$E_2 + E_4$
$\frac{23}{27}$	E_4
$\frac{8}{9}$	$E_2 + E_4$
$\frac{25}{27}$	E_4
$\frac{26}{27}$	E_4
1	$E_1 + E_2 + E_3 + E_4 + E_5 + E_6 + D_{aff}$.

Example 2

- Take $X = \mathbb{C}^3$ and D given by

$$(x^3 + y^3 + z^3)^2 + g(x, y, z) = 0,$$

where $g(x, y, z)$ is a generic homogeneous polynomial of degree 7.

Example 2

- Take $X = \mathbb{C}^3$ and D given by
$$(x^3 + y^3 + z^3)^2 + g(x, y, z) = 0,$$
where $g(x, y, z)$ is a generic homogeneous polynomial of degree 7.
- A resolution can be obtained after 22 point blow-ups, followed by two blow-ups centered at an elliptic curve C.

Example 2

- Take $X = \mathbb{C}^3$ and D given by
$$(x^3 + y^3 + z^3)^2 + g(x, y, z) = 0,$$
where $g(x, y, z)$ is a generic homogeneous polynomial of degree 7.
- A resolution can be obtained after 22 point blow-ups, followed by two blow-ups centered at an elliptic curve C.
- E_2 and E_3 are ruled surfaces over C, so their Picard groups are very complicated!

Example 2

- Take $X = \mathbb{C}^3$ and D given by
$$(x^3 + y^3 + z^3)^2 + g(x, y, z) = 0,$$
where $g(x, y, z)$ is a generic homogeneous polynomial of degree 7.
- A resolution can be obtained after 22 point blow-ups, followed by two blow-ups centered at an elliptic curve C.
- E_2 and E_3 are ruled surfaces over C, so their Picard groups are very complicated!

$$\pi^* D = D_{aff} + 6E_1 + 8\sum_{i=1}^{21} E_i^p + 7E_2 + 14E_3,$$

$$K_\pi = 2E_1 + 4\sum_{i=1}^{21} E_i^p + 3E_2 + 6E_3.$$

Example 2

It turns out that for the divisors we encounter in the unloading procedure, we can check effectivity.

Example 2

It turns out that for the divisors we encounter in the unloading procedure, we can check effectivity.
We find that the set of supercandidates in $(0, 1]$ is

$$\left\{ \frac{7}{14}, \frac{9}{14}, \frac{11}{14}, \frac{13}{14}, 1 \right\}$$

Example 2

It turns out that for the divisors we encounter in the unloading procedure, we can check effectivity.
We find that the set of supercandidates in $(0, 1]$ is

$$\left\{ \frac{7}{14}, \frac{9}{14}, \frac{11}{14}, \frac{13}{14}, 1 \right\}$$

and all of them are jumping numbers contributed by E_3.

Implementation

- For the computation of a log resolution, use “resolve.lib” or “reszeta.lib” in Singular.

- For the computation of a log resolution, use "resolve.lib" or "reszeta.lib" in Singular.
- If we are able to describe effective divisors on an exceptional divisor, the computation of supercandidates and their minimal jumping divisors is purely combinatorial.

- For the computation of a log resolution, use “resolve.lib” or “reszeta.lib” in Singular.
- If we are able to describe effective divisors on an exceptional divisor, the computation of supercandidates and their minimal jumping divisors is purely combinatorial.

Obstructions:

- For the computation of a log resolution, use “resolve.lib” or “reszeta.lib” in Singular.
- If we are able to describe effective divisors on an exceptional divisor, the computation of supercandidates and their minimal jumping divisors is purely combinatorial.

Obstructions:

- Describing when $-D|_E$ is effective is a very hard problem in general.

- For the computation of a log resolution, use “resolve.lib” or “reszeta.lib” in Singular.
- If we are able to describe effective divisors on an exceptional divisor, the computation of supercandidates and their minimal jumping divisors is purely combinatorial.

Obstructions:

- Describing when $-D|_E$ is effective is a very hard problem in general.
- Checking whether a supercandidate is a jumping number can be hard if our methods do not apply.

- For the computation of a log resolution, use “resolve.lib” or “reszeta.lib” in Singular.
- If we are able to describe effective divisors on an exceptional divisor, the computation of supercandidates and their minimal jumping divisors is purely combinatorial.

Obstructions:

- Describing when $-D|_E$ is effective is a very hard problem in general.
- Checking whether a supercandidate is a jumping number can be hard if our methods do not apply.

However, in many examples, our methods are sufficient.

Thank you!