ACSV: help wanted from computer algebra(ists)

Mark C. Wilson
University of Auckland

Computer Algebra in Combinatorics
Schrödinger Institute
Vienna
2017-11-14

Standing assumptions

- We use boldface to denote a multi-index: $\mathbf{z} = (z_1, \dots, z_d)$, $\mathbf{r} = (r_1, \dots, r_d)$. Similarly $\mathbf{z}^{\mathbf{r}} = z_1^{r_1} \dots z_d^{r_d}$.

Standing assumptions

- We use boldface to denote a multi-index: $\mathbf{z} = (z_1, \ldots, z_d)$, $\mathbf{r} = (r_1, \ldots, r_d)$. Similarly $\mathbf{z}^{\mathbf{r}} = z_1^{r_1} \ldots z_d^{r_d}$.
- A (multivariate) sequence is a function $a : \mathbb{N}^d \to \mathbb{C}$ for some fixed d. Usually write $a_{\mathbf{r}}$ instead of $a(\mathbf{r})$.

Standing assumptions

- We use boldface to denote a multi-index: $\mathbf{z} = (z_1, \ldots, z_d)$, $\mathbf{r} = (r_1, \ldots, r_d)$. Similarly $\mathbf{z}^{\mathbf{r}} = z_1^{r_1} \ldots z_d^{r_d}$.
- A (multivariate) sequence is a function $a : \mathbb{N}^d \to \mathbb{C}$ for some fixed d. Usually write $a_{\mathbf{r}}$ instead of $a(\mathbf{r})$.
- The generating function (GF) is the formal power series

$$F(\mathbf{z}) = \sum_{\mathbf{r} \in \mathbb{N}^d} a_{\mathbf{r}} \mathbf{z}^{\mathbf{r}}.$$

Standing assumptions

- We use boldface to denote a multi-index: $\mathbf{z} = (z_1, \ldots, z_d)$, $\mathbf{r} = (r_1, \ldots, r_d)$. Similarly $\mathbf{z}^{\mathbf{r}} = z_1^{r_1} \ldots z_d^{r_d}$.
- A (multivariate) sequence is a function $a : \mathbb{N}^d \to \mathbb{C}$ for some fixed d. Usually write $a_{\mathbf{r}}$ instead of $a(\mathbf{r})$.
- The generating function (GF) is the formal power series

$$F(\mathbf{z}) = \sum_{\mathbf{r} \in \mathbb{N}^d} a_{\mathbf{r}} \mathbf{z}^{\mathbf{r}}.$$

- Assume $F(\mathbf{z}) = G(\mathbf{z})/H(\mathbf{z})$ where G, H are polynomials. The singular variety $\mathcal{V} := \{\mathbf{z} : H(\mathbf{z}) = 0\}$ consists of poles.

Standing assumptions

- We use boldface to denote a multi-index: $\mathbf{z} = (z_1, \ldots, z_d)$, $\mathbf{r} = (r_1, \ldots, r_d)$. Similarly $\mathbf{z}^{\mathbf{r}} = z_1^{r_1} \ldots z_d^{r_d}$.
- A (multivariate) sequence is a function $a : \mathbb{N}^d \to \mathbb{C}$ for some fixed d. Usually write $a_{\mathbf{r}}$ instead of $a(\mathbf{r})$.
- The generating function (GF) is the formal power series

$$F(\mathbf{z}) = \sum_{\mathbf{r} \in \mathbb{N}^d} a_{\mathbf{r}} \mathbf{z}^{\mathbf{r}}.$$

- Assume $F(\mathbf{z}) = G(\mathbf{z})/H(\mathbf{z})$ where G, H are polynomials. The singular variety $\mathcal{V} := \{\mathbf{z} : H(\mathbf{z}) = 0\}$ consists of poles.
- To avoid discussing topology, assume all coefficients of F are nonnegative.

Outline of ACSV results

- Given direction $\overline{\mathbf{r}}$, asymptotics in that direction are determined by a (generically zero-dimensional) variety, of critical points.

Outline of ACSV results

- Given direction $\overline{\mathbf{r}}$, asymptotics in that direction are determined by a (generically zero-dimensional) variety, of critical points.
- We may restrict to a dominant point $\mathbf{z}_*(\overline{\mathbf{r}})$ lying in the positive orthant, which determines the exponential rate.

Outline of ACSV results

- Given direction $\overline{\mathbf{r}}$, asymptotics in that direction are determined by a (generically zero-dimensional) variety, of critical points.
- We may restrict to a dominant point $\mathbf{z}_*(\overline{\mathbf{r}})$ lying in the positive orthant, which determines the exponential rate.
- For subexponential factors, there is an asymptotic series $\mathcal{A}(\mathbf{z}_*)$ depending on the type of singularity at $\mathbf{z}_*$. Each term is computable from finitely many derivatives of G and H at $\mathbf{z}_*$.

Outline of ACSV results

- Given direction $\overline{\mathbf{r}}$, asymptotics in that direction are determined by a (generically zero-dimensional) variety, of critical points.
- We may restrict to a dominant point $\mathbf{z}_*(\overline{\mathbf{r}})$ lying in the positive orthant, which determines the exponential rate.
- For subexponential factors, there is an asymptotic series $\mathcal{A}(\mathbf{z}_*)$ depending on the type of singularity at $\mathbf{z}_*$. Each term is computable from finitely many derivatives of G and H at $\mathbf{z}_*$.
- This yields an asymptotic expansion

$$a_{\mathbf{r}} \sim \mathbf{z}_*(\overline{\mathbf{r}})^{-\mathbf{r}}\, \mathcal{A}(\mathbf{z}_*)$$

that is uniform on compact subsets of directions, provided the geometry at $\mathbf{z}_*(\overline{\mathbf{r}})$ does not change.

Simplest asymptotic formulae

- Smooth point:

$$a_{\mathbf{r}} \sim \mathbf{z}_*(\overline{\mathbf{r}})^{-\mathbf{r}} \sqrt{\frac{1}{(2\pi|\mathbf{r}|)^{d-1}\kappa(\mathbf{z}_*(\overline{\mathbf{r}}))}} \frac{G(\mathbf{z}_*(\overline{\mathbf{r}}))}{|\nabla_{\log} H(\mathbf{z}_*(\overline{\mathbf{r}}))|}$$

where $|\mathbf{r}| = \sum_i r_i$ and κ is the Gaussian curvature of $\log \mathcal{V}$ at $\log \mathbf{z}_*(\overline{\mathbf{r}})$.

Simplest asymptotic formulae

- Smooth point:

$$a_{\mathbf{r}} \sim \mathbf{z}_*(\overline{\mathbf{r}})^{-\mathbf{r}} \sqrt{\frac{1}{(2\pi|\mathbf{r}|)^{d-1}\kappa(\mathbf{z}_*(\overline{\mathbf{r}}))}} \frac{G(\mathbf{z}_*(\overline{\mathbf{r}}))}{|\nabla_{\log} H(\mathbf{z}_*(\overline{\mathbf{r}}))|}$$

  where $|\mathbf{r}| = \sum_i r_i$ and κ is the Gaussian curvature of $\log \mathcal{V}$ at $\log \mathbf{z}_*(\overline{\mathbf{r}})$.
- The Gaussian curvature can be computed explicitly in terms of derivatives of H to second order.

Simplest asymptotic formulae

- Smooth point:

$$a_{\mathbf{r}} \sim \mathbf{z}_*(\overline{\mathbf{r}})^{-\mathbf{r}} \sqrt{\frac{1}{(2\pi|\mathbf{r}|)^{d-1}\kappa(\mathbf{z}_*(\overline{\mathbf{r}}))}} \frac{G(\mathbf{z}_*(\overline{\mathbf{r}}))}{|\nabla_{\log} H(\mathbf{z}_*(\overline{\mathbf{r}}))|}$$

  where $|\mathbf{r}| = \sum_i r_i$ and κ is the Gaussian curvature of $\log \mathcal{V}$ at $\log \mathbf{z}_*(\overline{\mathbf{r}})$.
- The Gaussian curvature can be computed explicitly in terms of derivatives of H to second order.
- Multiple point:

$$a_{\mathbf{r}} \sim \mathbf{z}_*(\overline{\mathbf{r}})^{-\mathbf{r}} G(\mathbf{z}_*(\overline{\mathbf{r}})) \det J(\mathbf{z}_*(\overline{\mathbf{r}}))^{-1}$$

  where J is the Jacobian matrix $(\partial H_i/\partial z_j)$.

Why we need computer algebra

1. Conversion of G/H to various forms.

Why we need computer algebra

1. Conversion of G/H to various forms.
2. Classification of singularities.

Why we need computer algebra

1. Conversion of G/H to various forms.
2. Classification of singularities.
3. Setting up critical point equations on the various strata of $\mathcal{V}$.

Why we need computer algebra

1. Conversion of G/H to various forms.
2. Classification of singularities.
3. Setting up critical point equations on the various strata of $\mathcal{V}$.
4. Determining which critical points contribute.

Why we need computer algebra

1. Conversion of G/H to various forms.
2. Classification of singularities.
3. Setting up critical point equations on the various strata of $\mathcal{V}$.
4. Determining which critical points contribute.
5. Solution of equations: find critical point explicitly, so exponential rate is determined.

Why we need computer algebra

1. Conversion of G/H to various forms.
2. Classification of singularities.
3. Setting up critical point equations on the various strata of $\mathcal{V}$.
4. Determining which critical points contribute.
5. Solution of equations: find critical point explicitly, so exponential rate is determined.
6. Simplifying algebraic expressions involving values of G, H at critical points.

Why we need computer algebra

1. Conversion of G/H to various forms.
2. Classification of singularities.
3. Setting up critical point equations on the various strata of $\mathcal{V}$.
4. Determining which critical points contribute.
5. Solution of equations: find critical point explicitly, so exponential rate is determined.
6. Simplifying algebraic expressions involving values of G, H at critical points.
7. Differential operators applied to G, H at critical points, using parametrized data, for higher order asymptotics.

Some conceptual difficulties

- Our methods are analytic, so computations should be carried out in the analytic local ring (the ring of germs of holomorphic functions at a point).

Some conceptual difficulties

- Our methods are analytic, so computations should be carried out in the analytic local ring (the ring of germs of holomorphic functions at a point).
- Computation in this ring is trickier than in polynomial rings. However there is a theory of computation in local rings and apparently SINGULAR implements some of it.

Some conceptual difficulties

- Our methods are analytic, so computations should be carried out in the analytic local ring (the ring of germs of holomorphic functions at a point).
- Computation in this ring is trickier than in polynomial rings. However there is a theory of computation in local rings and apparently SINGULAR implements some of it.
- In many applications we have a global factorization in the polynomial ring that coincides with the local analytic factorization. We call this the simple case. The current implementation considers only this case.

Some conceptual difficulties

- Our methods are analytic, so computations should be carried out in the analytic local ring (the ring of germs of holomorphic functions at a point).
- Computation in this ring is trickier than in polynomial rings. However there is a theory of computation in local rings and apparently SINGULAR implements some of it.
- In many applications we have a global factorization in the polynomial ring that coincides with the local analytic factorization. We call this the simple case. The current implementation considers only this case.
- Help wanted in finding the state of the art!

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.
  - Classification of singularities.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.
  - Classification of singularities.
  - Setting up critical point equations on the various strata.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.
  - Classification of singularities.
  - Setting up critical point equations on the various strata.
  - Determining which critical points contribute.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.
  - Classification of singularities.
  - Setting up critical point equations on the various strata.
  - Determining which critical points contribute.
  - Solution of equations: find critical point explicitly, so exponential rate is determined.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.
  - Classification of singularities.
  - Setting up critical point equations on the various strata.
  - Determining which critical points contribute.
  - Solution of equations: find critical point explicitly, so exponential rate is determined.
  - Simplifying formulae involving values of G, H at critical points.

Implementation

- Our basic results for smooth and multiple points are implemented in open source software mostly written by Alex Raichev. This was a standalone package but thanks to Daniel Krenn it is now part of the core Sage distribution. It has a few thousand lines of code.
- Status:
  - Conversion of G/H to various forms.
  - Classification of singularities.
  - Setting up critical point equations on the various strata.
  - Determining which critical points contribute.
  - Solution of equations: find critical point explicitly, so exponential rate is determined.
  - Simplifying formulae involving values of G, H at critical points.
  - Differential operators applied to G, H at critical points, using derived data, for higher order asymptotics.

Example

- Example: let $H = H_1 H_2 H_3 := (1-x)(1-y)(1-xy)$. In the local ring at $(1,1)$, each factor should be in the ideal generated by the other two (Nullstellensatz).

Example

- Example: let $H = H_1 H_2 H_3 := (1-x)(1-y)(1-xy)$. In the local ring at $(1,1)$, each factor should be in the ideal generated by the other two (Nullstellensatz).
- In fact it is true globally, since $H_3 = H_1 + H_2 - H_1 H_2$. (Nullstellensatz certificate).

Example

- Example: let $H = H_1 H_2 H_3 := (1-x)(1-y)(1-xy)$. In the local ring at $(1,1)$, each factor should be in the ideal generated by the other two (Nullstellensatz).
- In fact it is true globally, since $H_3 = H_1 + H_2 - H_1 H_2$. (Nullstellensatz certificate).
- Thus eventually we obtain

$$F = \frac{1}{H_1 H_2 H_3} = \cdots = \frac{2-y}{(1-y)(1-xy)^2} + \frac{1}{(1-x)(1-xy)^2}.$$

Example

- Example: let $H = H_1 H_2 H_3 := (1-x)(1-y)(1-xy)$. In the local ring at $(1,1)$, each factor should be in the ideal generated by the other two (Nullstellensatz).
- In fact it is true globally, since $H_3 = H_1 + H_2 - H_1 H_2$. (Nullstellensatz certificate).
- Thus eventually we obtain

$$F = \frac{1}{H_1 H_2 H_3} = \cdots = \frac{2-y}{(1-y)(1-xy)^2} + \frac{1}{(1-x)(1-xy)^2}.$$

- Reduction to the squarefree case is then easy and algorithmic.

Example

- Example: let $H = H_1 H_2 H_3 := (1-x)(1-y)(1-xy)$. In the local ring at $(1,1)$, each factor should be in the ideal generated by the other two (Nullstellensatz).
- In fact it is true globally, since $H_3 = H_1 + H_2 - H_1 H_2$. (Nullstellensatz certificate).
- Thus eventually we obtain

$$F = \frac{1}{H_1 H_2 H_3} = \cdots = \frac{2-y}{(1-y)(1-xy)^2} + \frac{1}{(1-x)(1-xy)^2}.$$

- Reduction to the squarefree case is then easy and algorithmic.
- Thus we can reduce to the case where the number of factors it at most the dimension.

Classifying singularities

- Are we in the simple case? Factor into irreducibles $H = \prod_i H_i^{n_i}$, check whether $\nabla H_i = 0, H_i = 0$ has a solution.

Classifying singularities

- Are we in the simple case? Factor into irreducibles $H = \prod_i H_i^{n_i}$, check whether $\nabla H_i = 0, H_i = 0$ has a solution.
- Multiple points of order k come from simultaneous vanishing of k irreducibles (for transversality, check the log gradients span a space of dimension k).

Classifying singularities

- Are we in the simple case? Factor into irreducibles $H = \prod_i H_i^{n_i}$, check whether $\nabla H_i = 0, H_i = 0$ has a solution.
- Multiple points of order k come from simultaneous vanishing of k irreducibles (for transversality, check the log gradients span a space of dimension k).
- I do not know how to do this in the general case. But the local factors are branches of algebraic functions, so someone must know something ....

Classifying singularities

- Are we in the simple case? Factor into irreducibles $H = \prod_i H_i^{n_i}$, check whether $\nabla H_i = 0, H_i = 0$ has a solution.
- Multiple points of order k come from simultaneous vanishing of k irreducibles (for transversality, check the log gradients span a space of dimension k).
- I do not know how to do this in the general case. But the local factors are branches of algebraic functions, so someone must know something ....
- Eventually, want to understand the worse singularities. How to compute a normal form for a singularity?

Example (effect of numerator, I)

- Given $F = G/H$ where $G = 1$, $H_1 = 3 - 2x - y$, $H_2 = 3 - x - 2y$, $H = H_1 H_2$.

Example (effect of numerator, I)

- Given $F = G/H$ where $G = 1$, $H_1 = 3 - 2x - y$, $H_2 = 3 - x - 2y$, $H = H_1 H_2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$. We are in the simple case.

Example (effect of numerator, I)

- Given $F = G/H$ where $G = 1$, $H_1 = 3 - 2x - y$, $H_2 = 3 - x - 2y$, $H = H_1 H_2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$. We are in the simple case.
- We obtain $a_{rr} \sim 3$ by multiple point formula below.

Example (effect of numerator, I)

- Given $F = G/H$ where $G = 1$, $H_1 = 3 - 2x - y$, $H_2 = 3 - x - 2y$, $H = H_1 H_2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$. We are in the simple case.
- We obtain $a_{rr} \sim 3$ by multiple point formula below.
- If we change to $G = x - y$, our implementation gives $a_{rr} \sim 0$, which is correct, but only by luck.

Example (effect of numerator, I)

- Given $F = G/H$ where $G = 1$, $H_1 = 3 - 2x - y$, $H_2 = 3 - x - 2y$, $H = H_1 H_2$.
- Here $\mathcal{V}$ is smooth at every point except $(1,1)$. We are in the simple case.
- We obtain $a_{rr} \sim 3$ by multiple point formula below.
- If we change to $G = x - y$, our implementation gives $a_{rr} \sim 0$, which is correct, but only by luck.
- Here G is in the ideal $\langle H_1, H_2 \rangle$, $G = H_2 - H_1$ so

$$\frac{G}{H} = \frac{1}{3 - 2x - y} - \frac{1}{3 - x - 2y}.$$

Example (effect of numerator, I)

- Given $F = G/H$ where $G = 1$, $H_1 = 3 - 2x - y$, $H_2 = 3 - x - 2y$, $H = H_1 H_2$.
- Here $\mathcal{V}$ is smooth at every point except $(1,1)$. We are in the simple case.
- We obtain $a_{rr} \sim 3$ by multiple point formula below.
- If we change to $G = x - y$, our implementation gives $a_{rr} \sim 0$, which is correct, but only by luck.
- Here G is in the ideal $\langle H_1, H_2 \rangle$, $G = H_2 - H_1$ so

$$\frac{G}{H} = \frac{1}{3 - 2x - y} - \frac{1}{3 - x - 2y}.$$

- Thus it is not always even obvious whether a point is smooth, and vanishing numerator affects exponential rate.

Example (effect of numerator, II)

- Here $F = G/H$ where $G = x - y$, $H_1 = 3 - 2x - y^2$, $H_2 = 3 - x^2 - 2y^2$, $H = H_1 H_2$.

Example (effect of numerator, II)

- Here $F = G/H$ where $G = x - y$, $H_1 = 3 - 2x - y^2$, $H_2 = 3 - x^2 - 2y^2$, $H = H_1 H_2$.
- Again $\mathcal{V}$ is clearly smooth at every point except $(1, 1)$.

Example (effect of numerator, II)

- Here $F = G/H$ where $G = x - y$, $H_1 = 3 - 2x - y^2$, $H_2 = 3 - x^2 - 2y^2$, $H = H_1 H_2$.
- Again $\mathcal{V}$ is clearly smooth at every point except $(1, 1)$.
- Our current implementation gives $a_{rr} \sim 0$, but this is wrong.

Example (effect of numerator, II)

- Here $F = G/H$ where $G = x - y$, $H_1 = 3 - 2x - y^2$, $H_2 = 3 - x^2 - 2y^2$, $H = H_1 H_2$.
- Again $\mathcal{V}$ is clearly smooth at every point except $(1, 1)$.
- Our current implementation gives $a_{rr} \sim 0$, but this is wrong.
- Here G is not in the ideal $\langle H_1, H_2 \rangle$ of the polynomial ring.

Example (effect of numerator, II)

- Here $F = G/H$ where $G = x - y$, $H_1 = 3 - 2x - y^2$, $H_2 = 3 - x^2 - 2y^2$, $H = H_1 H_2$.
- Again $\mathcal{V}$ is clearly smooth at every point except $(1, 1)$.
- Our current implementation gives $a_{rr} \sim 0$, but this is wrong.
- Here G is not in the ideal $\langle H_1, H_2 \rangle$ of the polynomial ring.
- We need to go to the local analytic ring. Ring theoretic arguments (Nullstellensatz, Noetherianity) show that G must lie in the ideal generated by H_1, H_2 and a simplification again occurs. Again we will have smooth point behaviour.

Example (effect of numerator, II)

- Here $F = G/H$ where $G = x - y$, $H_1 = 3 - 2x - y^2$, $H_2 = 3 - x^2 - 2y^2$, $H = H_1 H_2$.
- Again $\mathcal{V}$ is clearly smooth at every point except $(1, 1)$.
- Our current implementation gives $a_{rr} \sim 0$, but this is wrong.
- Here G is not in the ideal $\langle H_1, H_2 \rangle$ of the polynomial ring.
- We need to go to the local analytic ring. Ring theoretic arguments (Nullstellensatz, Noetherianity) show that G must lie in the ideal generated by H_1, H_2 and a simplification again occurs. Again we will have smooth point behaviour.
- How to do this algorithmically?

Example (effect of numerator, III)

- If the multiple point is not isolated (includes smooth points), vanishing numerator does not affect exponential rate, *except* if all coefficients are zero.

Example (effect of numerator, III)

- If the multiple point is not isolated (includes smooth points), vanishing numerator does not affect exponential rate, *except* if all coefficients are zero.
- I am not sure how to detect this latter case in general.

Example (effect of numerator, III)

- If the multiple point is not isolated (includes smooth points), vanishing numerator does not affect exponential rate, *except* if all coefficients are zero.
- I am not sure how to detect this latter case in general.
- For example, $(x-y)/(1-x-y)$ has identically zero diagonal. The contributing point for the main diagonal is $(1/2, 1/2)$ and the smooth point formula will yield 0 for each coefficient.

Example (effect of numerator, III)

- If the multiple point is not isolated (includes smooth points), vanishing numerator does not affect exponential rate, *except* if all coefficients are zero.
- I am not sure how to detect this latter case in general.
- For example, $(x-y)/(1-x-y)$ has identically zero diagonal. The contributing point for the main diagonal is $(1/2, 1/2)$ and the smooth point formula will yield 0 for each coefficient.
- If we perturb the direction slightly, we obtain similar results to above, and the first order asymptotic varies continuously in direction.

Example (effect of numerator, III)

- If the multiple point is not isolated (includes smooth points), vanishing numerator does not affect exponential rate, *except* if all coefficients are zero.
- I am not sure how to detect this latter case in general.
- For example, $(x-y)/(1-x-y)$ has identically zero diagonal. The contributing point for the main diagonal is $(1/2, 1/2)$ and the smooth point formula will yield 0 for each coefficient.
- If we perturb the direction slightly, we obtain similar results to above, and the first order asymptotic varies continuously in direction.
- Our implementation only tells us, with increasing effort, that each coefficient in the asymptotic expansion is zero. It would be nice to be able to detect this in a preprocessing step.

Example (local factorization of lemniscate)

- Given $F = 1/H$ where H is irreducible, given by $H(x, y) = 19 - 20x - 20y + 5x^2 + 14xy + 5y^2 - 2x^2y - 2xy^2 + x^2y^2$.

Example (local factorization of lemniscate)

- Given $F = 1/H$ where H is irreducible, given by $H(x, y) = 19 - 20x - 20y + 5x^2 + 14xy + 5y^2 - 2x^2y - 2xy^2 + x^2y^2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$, which we see by solving the system $\{H = 0, \nabla H = 0\}$.

Example (local factorization of lemniscate)

- Given $F = 1/H$ where H is irreducible, given by $H(x, y) = 19 - 20x - 20y + 5x^2 + 14xy + 5y^2 - 2x^2y - 2xy^2 + x^2y^2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$, which we see by solving the system $\{H = 0, \nabla H = 0\}$.
- At $(1, 1)$, changing variables to $h(u, v) := H(1 + u, 1 + v)$, we see that $h(u, v) = 4u^2 + 10uv + 4v^2 + C(u, v)$ where C has no terms of degree less than 3.

Example (local factorization of lemniscate)

- Given $F = 1/H$ where H is irreducible, given by $H(x, y) = 19 - 20x - 20y + 5x^2 + 14xy + 5y^2 - 2x^2y - 2xy^2 + x^2y^2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$, which we see by solving the system $\{H = 0, \nabla H = 0\}$.
- At $(1, 1)$, changing variables to $h(u, v) := H(1 + u, 1 + v)$, we see that $h(u, v) = 4u^2 + 10uv + 4v^2 + C(u, v)$ where C has no terms of degree less than 3.
- The quadratic part factors into distinct factors, showing that $(1, 1)$ is a transverse multiple point.

Example (local factorization of lemniscate)

- Given $F = 1/H$ where H is irreducible, given by $H(x, y) = 19 - 20x - 20y + 5x^2 + 14xy + 5y^2 - 2x^2y - 2xy^2 + x^2y^2$.
- Here $\mathcal{V}$ is smooth at every point except $(1, 1)$, which we see by solving the system $\{H = 0, \nabla H = 0\}$.
- At $(1, 1)$, changing variables to $h(u, v) := H(1 + u, 1 + v)$, we see that $h(u, v) = 4u^2 + 10uv + 4v^2 + C(u, v)$ where C has no terms of degree less than 3.
- The quadratic part factors into distinct factors, showing that $(1, 1)$ is a transverse multiple point.
- The current implementation does not deal with this at all.

Critical point equations

- A smooth point of $\mathcal{V}$ is critical for direction $\overline{\mathbf{r}}$ iff the outward normal to $\log \mathcal{V}$ is parallel to $\mathbf{r}$. In other words, for some $\lambda \in \mathbb{C}$, $\mathbf{z}_*$ solves

$$\nabla_{\log} H(\mathbf{z}) := (z_1 \partial H/\partial z_1, \ldots, z_d \partial H/\partial H_d) = \lambda \mathbf{r}$$
$$H(\mathbf{z}) = \mathbf{0}.$$

This gives $d+1$ polynomial equations in $d+1$ unknowns.

Critical point equations

- A smooth point of $\mathcal{V}$ is critical for direction $\overline{\mathbf{r}}$ iff the outward normal to $\log \mathcal{V}$ is parallel to $\mathbf{r}$. In other words, for some $\lambda \in \mathbb{C}$, $\mathbf{z}_*$ solves

$$\nabla_{\log} H(\mathbf{z}) := (z_1 \partial H/\partial z_1, \ldots, z_d \partial H/\partial H_d) = \lambda \mathbf{r}$$
$$H(\mathbf{z}) = \mathbf{0}.$$

  This gives $d+1$ polynomial equations in $d+1$ unknowns.
- For multiple points given by k factors intersecting, there is a related polynomial system expressing the vanishing of all minors of order $k+1$ of a $k+1$ by d matrix. This is not yet implemented, but is easy provided we can deal with factorization of H (e.g. in the simple case).

Critical point equations

- A smooth point of $\mathcal{V}$ is critical for direction $\overline{\mathbf{r}}$ iff the outward normal to $\log \mathcal{V}$ is parallel to $\mathbf{r}$. In other words, for some $\lambda \in \mathbb{C}$, $\mathbf{z}_*$ solves

$$\nabla_{\log} H(\mathbf{z}) := (z_1 \partial H/\partial z_1, \ldots, z_d \partial H/\partial H_d) = \lambda \mathbf{r}$$
$$H(\mathbf{z}) = \mathbf{0}.$$

  This gives $d+1$ polynomial equations in $d+1$ unknowns.
- For multiple points given by k factors intersecting, there is a related polynomial system expressing the vanishing of all minors of order $k+1$ of a $k+1$ by d matrix. This is not yet implemented, but is easy provided we can deal with factorization of H (e.g. in the simple case).
- In fact $\lambda \in \mathbb{R}$ which helps to eliminate some noncontributing critical points.

Example (almost trivial)

- $G = 1, H = 1 - x - y.$

Example (almost trivial)

- $G = 1, H = 1 - x - y$.
- $\nabla_{\log} H = (x, y)$ so clearly all points are smooth.

Example (almost trivial)

- $G = 1, H = 1 - x - y$.
- $\nabla_{\log} H = (x, y)$ so clearly all points are smooth.
- $(x, y) = \lambda(r, s), H = 0$ so $x = r/(r + s), y = s/(r + s)$, so exponential rate $(r + s)^{r+s}/(r^r s^s)$. Note solution is unique.

Example (almost trivial)

- $G = 1, H = 1 - x - y$.
- $\nabla_{\log} H = (x, y)$ so clearly all points are smooth.
- $(x, y) = \lambda(r, s), H = 0$ so $x = r/(r+s), y = s/(r+s)$, so exponential rate $(r+s)^{r+s}/(r^r s^s)$. Note solution is unique.
- Leading term easily computed to equal

$$\sqrt{\frac{r+s}{2\pi r s}}.$$

Example (almost trivial)

- $G = 1, H = 1 - x - y$.
- $\nabla_{\log} H = (x, y)$ so clearly all points are smooth.
- $(x, y) = \lambda(r, s), H = 0$ so $x = r/(r+s), y = s/(r+s)$, so exponential rate $(r+s)^{r+s}/(r^r s^s)$. Note solution is unique.
- Leading term easily computed to equal

$$\sqrt{\frac{r+s}{2\pi r s}}.$$

- For higher order terms, even this example should be done by computer algebra. For example

$$a_{rr} \sim 4^r \left[\frac{1}{\sqrt{\pi r}} - \frac{1}{8\sqrt{\pi r^3}} + \frac{1}{128\sqrt{\pi r^5}}\right].$$

Example (easy)

- General (symbolic) direction for previous example: 1 second.

Example (easy)

- General (symbolic) direction for previous example: 1 second.
- Polyominoes: fixed direction in 1 second.

Example (easy)

- General (symbolic) direction for previous example: 1 second.
- Polyominoes: fixed direction in 1 second.
- Most smooth problems in 2 variables can be done within a few seconds for up to order 3 and many to higher order.

Example (easy)

- General (symbolic) direction for previous example: 1 second.
- Polyominoes: fixed direction in 1 second.
- Most smooth problems in 2 variables can be done within a few seconds for up to order 3 and many to higher order.
- For 3 or more variables, even order 3 can be slow.

Example (easy)

- General (symbolic) direction for previous example: 1 second.
- Polyominoes: fixed direction in 1 second.
- Most smooth problems in 2 variables can be done within a few seconds for up to order 3 and many to higher order.
- For 3 or more variables, even order 3 can be slow.
- Double point examples in 2 variables are very easy, even with vanishing numerator.

Example (harder)

- An interesting lattice path problem yields

$$
\begin{aligned}
G &= (1+x)(1-2t(1+x^2)) \\
H &= (1-y)(1-t(1+x^2+xy^2))(1-t(1+x^2))
\end{aligned}
$$

Example (harder)

- An interesting lattice path problem yields

$$
\begin{aligned}
G &= (1+x)(1-2t(1+x^2)) \\
H &= (1-y)(1-t(1+x^2+xy^2))(1-t(1+x^2))
\end{aligned}
$$

- Critical points: we have $(1,1,1/3), (1,\sqrt{2},\frac{1}{4}), (1,-\sqrt{2},\frac{1}{4})$ $(-1,i\sqrt{2},\frac{1}{4}), (-1,i\sqrt{2},\frac{1}{4})$.

Example (harder)

- An interesting lattice path problem yields

$$G = (1+x)(1-2t(1+x^2))$$
$$H = (1-y)(1-t(1+x^2+xy^2))(1-t(1+x^2))$$

- Critical points: we have $(1,1,1/3), (1,\sqrt{2},\frac{1}{4}), (1,-\sqrt{2},\frac{1}{4})$ $(-1,i\sqrt{2},\frac{1}{4}), (-1,i\sqrt{2},\frac{1}{4})$.
- Automatic detection of contributing points is not implemented. In this case the highest point $(1,1,1/3)$ does not contribute but the others do.

Example (harder)

- An interesting lattice path problem yields

$$G = (1+x)(1-2t(1+x^2))$$
$$H = (1-y)(1-t(1+x^2+xy^2))(1-t(1+x^2))$$

- Critical points: we have $(1,1,1/3), (1,\sqrt{2},\frac{1}{4}), (1,-\sqrt{2},\frac{1}{4})$ $(-1,i\sqrt{2},\frac{1}{4}), (-1,i\sqrt{2},\frac{1}{4})$.
- Automatic detection of contributing points is not implemented. In this case the highest point $(1,1,1/3)$ does not contribute but the others do.
- First order asymptotic is zero at smooth point $(1,\sqrt{2},\frac{1}{4})$. Second order computation fails to halt in reasonable time (hours).

Why so slow?

- The problem in the previous example seems to be the multiple factors in H. In this case the positive contributing point is a zero of only one factor H_2 and is smooth. If we rewrite $G/H = (G/H_1 H_3)/H_2$, everything works fine, giving answer at that point

$$\frac{(2\sqrt{2})^r}{\pi r^2} \frac{\left(8 - 2\sqrt{2}\right)}{\left(3 - 2\sqrt{2}\right)}.$$

Why so slow?

- The problem in the previous example seems to be the multiple factors in H. In this case the positive contributing point is a zero of only one factor H_2 and is smooth. If we rewrite $G/H = (G/H_1H_3)/H_2$, everything works fine, giving answer at that point
$$\frac{(2\sqrt{2})^r}{\pi r^2}\frac{(8-2\sqrt{2})}{(3-2\sqrt{2})}.$$
- Similarly the current method for computing critical points gives completely spurious points such as $(4, 1, 1/17)$ when run on G/H.

Why so slow?

- The problem in the previous example seems to be the multiple factors in H. In this case the positive contributing point is a zero of only one factor H_2 and is smooth. If we rewrite $G/H = (G/H_1H_3)/H_2$, everything works fine, giving answer at that point
$$\frac{(2\sqrt{2})^r}{\pi r^2}\frac{(8-2\sqrt{2})}{(3-2\sqrt{2})}.$$
- Similarly the current method for computing critical points gives completely spurious points such as $(4,1,1/17)$ when run on G/H.
- Thus factorization is very important, which brings us back to the issues discussed earlier.

Asymptotic formulae — higher terms

- We change variable by $\mathbf{z} = \mathbf{z}_* \exp(i\theta)$ and derive asymptotics of a Fourier-Laplace integral $I(\lambda)$.

Asymptotic formulae — higher terms

- We change variable by $\mathbf{z} = \mathbf{z}_* \exp(i\theta)$ and derive asymptotics of a Fourier-Laplace integral $I(\lambda)$.
- The first term is usually easy to compute but others can be difficult, and they are necessary in many applications.

Asymptotic formulae — higher terms

- We change variable by $\mathbf{z} = \mathbf{z}_* \exp(i\theta)$ and derive asymptotics of a Fourier-Laplace integral $I(\lambda)$.
- The first term is usually easy to compute but others can be difficult, and they are necessary in many applications.
- This appears to be the main performance bottleneck in our current implementation.

Asymptotic formulae — higher terms

- We change variable by $\mathbf{z} = \mathbf{z}_* \exp(i\theta)$ and derive asymptotics of a Fourier-Laplace integral $I(\lambda)$.
- The first term is usually easy to compute but others can be difficult, and they are necessary in many applications.
- This appears to be the main performance bottleneck in our current implementation.
- For smooth and multiple points we have used an explicit formula of Hörmander.

Asymptotic formulae — higher terms

- We change variable by $\mathbf{z} = \mathbf{z}_* \exp(i\theta)$ and derive asymptotics of a Fourier-Laplace integral $I(\lambda)$.
- The first term is usually easy to compute but others can be difficult, and they are necessary in many applications.
- This appears to be the main performance bottleneck in our current implementation.
- For smooth and multiple points we have used an explicit formula of Hörmander.
- An alternative approach involving solving a system of equations may also be practical. We have not yet explored it.

Hörmander's explicit formula

For an isolated nondegenerate stationary point $\mathbf{0}$ in dimension d,

$$I(\lambda) \sim \left(\det\left(\frac{\lambda f''(\mathbf{0})}{2\pi}\right)\right)^{-1/2} \sum_{k \geq 0} \lambda^{-k} L_k(A, f)$$

where L_k is a differential operator of order $2k$ evaluated at $\mathbf{0}$:

$$\underline{f}(t) = f(t) - (1/2) t f''(\mathbf{0}) t^T$$

$$\mathcal{D} = \sum_{a,b} (f''(\mathbf{0})^{-1})_{a,b} (-\mathrm{i}\partial_a)(-\mathrm{i}\partial_b)$$

$$L_k(A, f) = \sum_{l \leq 2k} \frac{\mathcal{D}^{l+k}(A \underline{f}^l)(\mathbf{0})}{(-1)^k 2^{l+k} l!(l+k)!}.$$

For example $L_0(A, f) = A$,
$L_1(A, f) = -\mathcal{D}(A)/2 - \mathcal{D}^2(A\underline{f})/8 - \mathcal{D}^3(A\underline{f}^2)/48.$

Computing better with Hörmander's formula

- The current Sage code struggles when $d = 3, k = 3$, and sometimes even for smaller parameters. My guess is that we should be able to reorganize the computation to be more efficient.

Computing better with Hörmander's formula

- The current Sage code struggles when $d = 3, k = 3$, and sometimes even for smaller parameters. My guess is that we should be able to reorganize the computation to be more efficient.
- Note that $\underline{f}$ vanishes to order 3 at 0, so $A\underline{f}^l$ vanishes to order $3l$, and $\mathcal{D}$ is a 2nd order linear operator. When $\mathcal{D}^{l+k}$ is applied to $A\underline{f}^l$ and evaluated at 0, many terms are automatically zero.

Computing better with Hörmander's formula

- The current Sage code struggles when $d = 3, k = 3$, and sometimes even for smaller parameters. My guess is that we should be able to reorganize the computation to be more efficient.
- Note that $\underline{f}$ vanishes to order 3 at 0, so $A\underline{f}^l$ vanishes to order $3l$, and $\mathcal{D}$ is a 2nd order linear operator. When $\mathcal{D}^{l+k}$ is applied to $A\underline{f}^l$ and evaluated at 0, many terms are automatically zero.
- Maybe we can rewrite

$$\sum_k \lambda^{-k} L_k(A, f) = \sum_l \sum_{2k \geq l} \lambda^{-k} \frac{\mathcal{D}^{l+k}(A\underline{f}^l)(\mathbf{0})}{(-1)^k 2^{l+k} l!(l+k)!}.$$