GOOD TRUNCATIONS FOR COMPUTING INTEGRAL BASES

D. BASSON, J. BOEHM, S. LAPLAGNE, M. MARAIS

Abstract. We provide good truncation bound that speed up the computation of integral basis.

1. Introduction

Integral basis are very useful in real life.

Given a polynomial $f \in K[x, y]$ monic in Y, we wish to compute the local contribution to the integral basis at the origin or the local integral basis at the origin.

2. The determinacy

In this section we show:

(1) An upper bound for the determinacy can be found fast reducing the polynomial modulo any prime number p.
(2) If we truncate the polynoial f by standard degree at the determinacy, the characteristic exponents of f will not change.
(3) The integrality exponent of f can be computed from the characteristic exponents, so we can compute very fast the integrality exponent.

Remark: the coefficients of the Puiseux expansions might change, but in all examples we tried, the coefficients don't change after truncation.

3. Local contribution to the integral basis at the origin

In this section we focus on the computation of the local contribution at the origin.

In this case, by [1, Proposition 23] we can truncate the powers of X at order two times the integrality exponent and we will get the correct integral basis.

4. Local integral basis at the origin

When we are interested in the local integral basis at the origin we can also truncate powers of Y.

Applying [1, Proposition 23] in terms of Y we obtain that the factors at the origin are uniquely determined and they are the same factors as the factors obtained when we develope the factorization in terms of X, so we are able to recover the correct information.

The factor g_0 however corresponds to the branches of f at $Y = 0$ and the factor f_0 corresponds to the branches of f at $X = 0$, so we cannot recover the Puiseux expansions of f_0 from the Puiseux expansions of g_0.

Hence when we apply a truncation in Y we can expect to compute correctly the local integral basis but not the local contribution.

5. Timings

We get very good timings.

References

[1] Adrien Poteaux and Martin Weimann. Computing puiseux series: a fast divide and conquer algorithm. *Annales Henri Lebesgue*, 4:1061–1102, 2021.