Journal of Symbolic Computation 75 (2016) 171–192

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Bounded-degree factors of lacunary multivariate polynomials ☆

Bruno Grenet

LIRMM, Université de Montpellier, UMR 5506 CNRS, France

ARTICLE INFO

Article history:
Received 8 December 2014
Accepted 10 July 2015
Available online 4 November 2015

Keywords:
Multivariate lacunary polynomials
Polynomial factorization
Newton polygon
Puiseux series
Wronskian determinant

ABSTRACT

In this paper, we present a new method for computing bounded-degree factors of lacunary multivariate polynomials. In particular for polynomials over number fields, we give a new algorithm that takes as input a multivariate polynomial f in lacunary representation and a degree bound d and computes the irreducible factors of degree at most d of f in time polynomial in the lacunary size of f and in d. Our algorithm, which is valid for any field of zero characteristic, is based on a new gap theorem that enables reducing the problem to several instances of (a) the univariate case and (b) low-degree multivariate factorization.

The reduction algorithms we propose are elementary in that they only manipulate the exponent vectors of the input polynomial. The proof of correctness and the complexity bounds rely on the Newton polytope of the polynomial, where the underlying valued field consists of Puiseux series in a single variable.

1. Introduction

The factorization of polynomials is a well-studied subject in symbolic computation. Although there exist effective fields in which testing irreducibility of polynomials is undecidable (Fröhlich and Shepherdson, 1955), the irreducible factorization of univariate or multivariate polynomials can be computed in time polynomial in the degree of the input polynomial for many base fields. Without claim of exhaustiveness, one can cite the cases of polynomials over rational numbers (Lenstra et al., 1982;

☆ Partially supported by a LIX–Qualcomm®–Carnot postdoctoral fellowship, and by the French Agence Nationale de la Recherche under grant CATREL #ANR-12-BS02-001.

E-mail address: Bruno.Grenet@lirmm.fr.

http://dx.doi.org/10.1016/j.jsc.2015.11.013
0747-7171/

Kaltofen, 1989) and algebraic number fields (Lenstra, 1983; Landau, 1985; Lenstra, 1987), or over finite fields (Berlekamp, 1967). From a somewhat different perspective, one can also compute the factorization in an extension of the base field, such as (approximate) factorization in the real or complex numbers (Pan, 2002; Kaltofen et al., 2008) or absolute factorization, that is factorization over an algebraic closure of the base field (Chèze and Galligo, 2005).

The purpose of this paper is to propose polynomial-time algorithms when the input polynomial is given in *lacunary representation*, that is as a list of nonzero monomials. These algorithms have complexity *logarithmic* in the degree.[1] Note that in lacunary representation, even evaluating a polynomial over an input is intractable: For instance, the monomial X^d has lacunary size $O(\log d)$ while its evaluation on the input 2 is an integer of size d. More generally, testing the irreducibility of lacunary polynomials or computing the greatest common divisor of two lacunary polynomials are NP-hard problems (Plaisted, 1977; Karpinski and Shparlinski, 1999; Kaltofen and Koiran, 2005). This motivates refining our ambitions and computing only a partial factorization of the input polynomial, namely the irreducible factors of bounded degree.

1.1. Previous work

Cucker et al. (1999) gave an algorithm to compute the integer roots of univariate integer polynomials in time polynomial in the lacunary representation. This result was generalized by Lenstra (1999) who described an algorithm to compute the bounded-degree factors of polynomials over number fields. His algorithm takes as input a description of the number field by means of an irreducible polynomial with integer coefficients in dense representation, the polynomial to factor in lacunary representation, and a bound on the degree of the factors it computes. The complexity is polynomial in the size of the input and in the degree bound (rather than in its bit-size). Then, Kaltofen and Koiran (2005) generalized this result to the computation of linear factors of bivariate polynomials over the rational numbers, and then to the computation of bounded-degree factors of multivariate polynomials over number fields (Kaltofen and Koiran, 2006). Seemingly independently of this latest result, Avendaño et al. (2007) generalized the first result of Kaltofen and Koiran (2005) and gave an algorithm to compute the bounded-degree factors of bivariate polynomials over number fields. They also explained how to compute the bounded-degree factors with at least three monomials over an algebraic closure of the rational numbers. Note that the binomial factors include univariate linear factors and the number of such factors cannot be polynomially bounded in the logarithm of the degree. We proposed another algorithm for the computation of the multilinear factors in the bivariate and multivariate cases (Chattopadhyay et al., 2013; Chattopadhyay et al., submitted for publication). Since it relies on Lenstra's algorithm for univariate factors, it is valid in full generality over number fields only, though our approach works in more general settings and allow for partial results over any fields of characteristic zero and to some extent in positive characteristic. All these results are based on a technique, due to Cucker et al. (1999), that consists in finding *gaps* in the input polynomial (*cf.* next section).

Avendaño (2009) proposed a different technique to test whether a given linear factor divides a lacunary bivariate polynomial, again over number fields. To our knowledge, his approach does not allow to compute the factors. It is based on a bound on the number of real roots of the intersection of a lacunary polynomial with a line. This latter result has been extended to the intersection of a lacunary polynomial with a low-degree polynomial by Koiran et al. (2015). It appears that Avendaño's method could be combined with this more recent result to obtain an algorithm that tests whether a given low-degree polynomial divides a lacunary bivariate polynomial. Nevertheless this algorithm would only work with some low-degree polynomials, since it requires in particular the polynomial to have real roots.

Let us finally mention two other results. Sagraloff (2014) gave an algorithm to compute the real roots of an integer polynomial with arithmetic complexity polynomial in the size of the lacunary rep-

[1] The lacunary representation is also known as sparse representation in the literature. Yet is customary to use the term *lacunary* for algorithms of complexity logarithmic in the degree, and *sparse* for algorithms of complexity polynomial in the degree.