1D-5 Solving Approximate GCD of Multivariate Polynomials By Maple/Matlab/C Combination

Kai Li Lihong Zhi Matu-Tarow Noda
Department of Computer Science, Ehime University, Japan

1 Introduction

The problem of computing approximate GCD of multivariate polynomials with inexact coefficients has been studied in [1] and [2]. In [3], an approach that modifies the modular algorithm based on Hensel construction was proposed and discussed briefly. In this paper, we mention the implementation of the algorithm using Maple, Matlab and C in detail.

2 Hensel Algorithm for Multivariate GCD

The problem that we consider here is: Given two multivariate polynomials $F(x_1, \cdots, x_n)$ and $G(x_1, \cdots, x_n)$ with coefficients of limited accuracy, are there nearby polynomials with a non-trivial GCD for a given tolerance ϵ?

The EZGCD algorithm reduces two input polynomials to two univariate polynomials whose GCD is then lifted back to the multivariate domain using a generalized Newton's iteration. It is essential to choose a good evaluation point so that the homomorphic GCD is relatively prime to one of its cofactor. Now let us concentrate on Hensel construction. For simplicity, suppose we are given polynomials $F(x,y)$, $G^{(0)}(x)$, and $H^{(0)}(x)$ such that:

$$F \equiv G^{(0)} H^{(0)} \mod y \tag{1}$$

with $\deg_y(F) \leq n$, where $\deg_y(F)$ is the total degree of F w.r.t. y, and n is an integer. Assume we want to lift the two univariate factors to degree n in y. At step k of Hensel lifting, we want to compute $G^{(k)}(x,y)$ and $H^{(k)}(x,y)$ such that:

$$F \equiv G^{(k)} H^{(k)} \mod y^{k+1}, \quad \text{and} \tag{2}$$

$$G^{(k)} \equiv G^{(k-1)} \mod y^k, \quad H^{(k)} \equiv H^{(k-1)} \mod y^k \tag{3}$$

We can derive that lifting from y^k to y^{k+1} amounts to solving for $\Delta_i' s$ in the following univariate Diophantine equation:

$$\frac{F - G^{(k-1)} H^{(k-1)}}{y^k} \equiv \Delta_1^{(k)} H^{(0)} + \Delta_2^{(k)} G^{(0)} \mod y, \tag{4}$$

where $\Delta_i^{(k)} (i = 1, 2)$ are univariate polynomials in $\mathbb{C}[x]$. If $G^{(0)}(x)$ and $H^{(0)}(x)$ are relatively prime, then the problem is reduced to solve the linear algebra equations:

$$M \quad \vec{z} = \vec{b}, \tag{5}$$

where $\vec{b}$ is the coefficient vector of polynomial in the left side of (4), $\vec{z}$ is the coefficient vector of polynomial $\Delta_i^{(k)}(i = 1,2)$; M is Sylvester matrix of polynomial $G^{(0)}$ and $H^{(0)}$.

Here we prefer QR decomposition to solve (5). Because it is easy to exploit the structure of the Sylvester matrix M for a efficient and stable algorithm, by adopting both Given rotations and Householder transformation methods.

Another important issue related to Hensel construction is when the Hensel lifting is terminated. In the symbolic computation, process will be terminated after $k \geq \deg_y(F)$. However, it shows difference in the floating-point computation case. Both factors $G^{(k)}$ and $H^{(k)}$ continue to include terms with high degree w.r.t. y. The computation becomes more unreliable when the power of y increase to a relatively high. So rather than lifting to the full degree w.r.t. y, we prefer to estimate the degrees of the y in the factors and stop the lifting as soon as one factor arrives it. Try to get the full expression of another factor by solving an overdetermined linear equation system, and to improve them by solving a linearized minimization problem as in [3].

$$\min_{\Delta G, \Delta H} \|F - GH - G\Delta H - \Delta GH\|. \tag{6}$$

3 Comprehensive Environment Construction

It is clear that we need powerful tools to handle several numerical linear algebraic problems such as solving linear systems, least squares and minimization. A large amount of experiments have been done to show much more efficient if we make use of Matlab or C programs. Our comprehensive system for approximate GCD computation of multivariate polynomials can be described as:

- Use Maple to perform symbolic computation;
- Invoke Matlab function from within Maple system to perform numerical computation such as linear programming solving and least squares solving.
- Merge C programming application routines `qr_sylvester.c` acting as the Sylvester matrix QR solver into Matlab being as its plug-in function, so as to be available from Maple.

4 Conclusion

In this paper, we briefly discuss using Hensel lifting algorithm to compute approximate GCD of multivariate polynomials in $\mathbb{C}$. A comprehensive environment has been constructed by Maple/Matlab/C combination for a more efficient algorithm.

References

[1] Noda, M.-T. and Sasaki, T.: Approximate GCD and its application to illconditioned algebraic equations, *J. Comput. Appl. Math., 38*(1991), 335-351.

[2] Corless, R. M., Gianni, P. M., Trager, G. M. and Watt, S. M.: The singular value decomposition for polynomial systems, *Proc. ISSAC '95*, ACM Press, New York, 1995, 195-207.

[3] L.H.Zhi and M.-T. Noda: Approximate GCD of Multivariate Polynomials, submited to ASCM '2000.