The Non-Monic Case in the Sparse Modular GCD Algorithm

Jennifer de Kleine* dekleine@cecm.sfu.ca
Michael Monagan* mmonagan@cecm.sfu.ca
Allan Wittkopf awittkop@maplesoft.com

Centre for Experimental and Constructive Mathematics
Simon Fraser University,
Burnaby, British Columbia, V5A 1S6 Canada

Abstract

The sparse modular GCD algorithm was presented by Zippel [5, 6] for computing the greatest common divisor of two multivariate polynomials over the integers. It improves the efficiency of Brown's algorithm when the gcd is sparse. We extend this algorithm to work for non-monic GCDs.

The non-monic case occurs when the GCD has a leading coefficient involving one or more variables. For example, the GCD $G = (4y + 2z)x^2 + 7 \in \mathbb{Z}[x, y, z]$ is non-monic in the main variable x. The problem is that at the bottom level of our algorithm we call the Euclidean algorithm, which returns a monic GCD. We describe two different approaches for handling the non-monic case. One involves solving a sequence of large structured linear systems. The other method uses a sparse rational function interpolation algorithm.

1 Introduction

In [1], Brown presented a first efficient solution to the polynomial greatest common divisor (GCD) problem: given two non-zero polynomials $A, B \in \mathbb{Z}[x_1, ..., x_n]$ compute their GCD G. Brown's algorithm is a dense modular algorithm. By reducing the inputs modulo primes and evaluating out all variables except one, x_1 say, it reduces the multivariate GCD problem to a possibly large number of univariate GCD problems over a finite field where Euclid's algorithm works well. If d bounds the degree of A and B in all variables, then Brown's algorithm uses $O(d^{n-1})$ evaluation points and does $O(d^{n-1})$ univariate GCD computations.

In practice, most polynomial GCD problems in many variables are sparse. For example, $G = x_1^d + x_2^d + ... + x_n^d - 1$ is sparse. It has $n + 1$ non-zero terms. If all terms of total degree $\leq d$ were present it would have $\binom{n+d}{d}$ non-zero terms. On this problem Brown's algorithm will do $O(d^{n-1})$ univariate GCD computations even though G has only $n + 1$ terms.

In [5, 6], Zippel presented a modification of Brown's modular GCD algorithm [1], providing an effective algorithm for when G is sparse. If t is the maximum number of terms of a coefficient of G in x_1 (in the example $t = n$), Zippel's algorithm requires $O((n-1)dt)$ evaluations and does $O((n-1)dt)$ univariate GCD computations. This is clearly much better than Brown's algorithm for $n = 10$ or more variables.

*This work was supported by the MITACS NCE of Canada and NSERC of Canada

A discussion of both algorithms together with examples can be found in Chapter 7 of the text "Algorithms for Computer Algebra" by Geddes, Czapor and Labahn [2]. Zippel's algorithm is limited in that it is only designed to handle monic GCDs. Although many sparse inputs will be monic in one of the variables we need an effective algorithm for when this is not the case. In this paper we investigate two different approaches for extending Zippel's algorithm to work for the non-monic case.

The organization of this paper is as follows. In section 2. we give a brief overview of Zippel's sparse modular GCD algorithm together with a simple example. In section 3. we discuss the non-monic problem and two approaches to dealing with it. In particular, in section 3.1 we discuss a normalization technique and in section 3.2 we discuss an alternative approach which uses a sparse rational function interpolation method. In section 4 we conclude with some final remarks about the current state of the project and future plans.

2 Zippel's Sparse Modular GCD Algorithm

Zippel's sparse modular GCD algorithm differs from Brown's modular GCD algorithm in the interpolation step. Zippel's algorithm is based on the idea that the number of non-zero terms in a multivariate polynomial is often much smaller than the number of possible terms.

There are two stages to the algorithm. In the first stage the input polynomials $A, B \in \mathbb{Z}[x_1, ..., x_n]$ are reduced modulo primes p such that the leading coefficients do not vanish. The GCDs $G_p = GCD(A_p, B_p) \bmod p$ are computed (using the second stage of the algorithm) and the $GCD(A, B)$ over $\mathbb{Z}$ is then reconstructed by applying the Chinese Remainder Theorem.

In the second stage of the algorithm the GCD in $\mathbb{Z}_p[x_1, ..., x_n]$ is computed as follows. The n-variate problem is reduced to a series of $(n-1)$-variate problems by evaluating at random points, $\alpha \in \mathbb{Z}_p$, for x_n. This is done for sufficiently many α until the $n - variate$ gcd can be recovered by interpolation. This step is applied recursively. At the bottom level we apply the Euclidean Algorithm to compute the univariate GCD modulo p.

In both stages of the algorithm, the first image at any given level is computed recursively, call this G_1. Any subsequent images computed at that level are then reconstructed using a sparse interpolation based on the form of G_1. The main assumption is that G_1 is of the correct form, that is, all non-zero terms are present. The remaining images may then be computed by solving a sequence of linear systems for the unknown coefficients.

Example 1. Let $G, A, B \in \mathbb{Z}[x, y]$ be defined as follows

$$G = x^2 + 3y^3x + 35$$
$$A = (y+1) \times G = (y+1)\,x^2 + (3y^4 + 3y^3)\,x + 35y + 35$$
$$B = (x+1) \times G = x^3 + (3y^3 + 1)\,x^2 + (3y^3 + 35)\,x + 35$$

Working modulo $p_1 = 11$ we compute our first GCD image in $\mathbb{Z}_{11}[x, y]$ using a dense interpolation:

1. Compute a degree bound on y, $d_y = \mathsf{degree}(\mathsf{Gcd}(A(1,\, y), B(1,\, y))\ mod\ 11) = 3$.
2. Evaluate at four random points for y and compute our univariate GCD images in $\mathbb{Z}_{11}[x]$ using the Euclidean algorithm:

$$\begin{aligned} g_1 \leftarrow \mathsf{Gcd}(A(x,2),\ B(x,2))\ \mathsf{mod}\ 11 &= x^2 + 2x + 2 \\ g_2 \leftarrow \mathsf{Gcd}(A(x,4),\ B(x,4))\ \mathsf{mod}\ 11 &= x^2 + 5x + 2 \\ g_3 \leftarrow \mathsf{Gcd}(A(x,5),\ B(x,5))\ \mathsf{mod}\ 11 &= x^2 + x + 2 \\ g_4 \leftarrow \mathsf{Gcd}(A(x,6),\ B(x,6))\ \mathsf{mod}\ 11 &= x^2 + 10x + 2. \end{aligned}$$

3. Interpolate in y to construct our first bivariate image modulo 11,

$$G_1 \leftarrow \mathsf{Interp}([\,2,\,4,\,5,\,6\,],[\,g_1,\,g_2,\,g_3,\,g_4\,],y) \bmod 11 \;=\; x^2 + 3y^3x + 2.$$

Working modulo $p_2 = 13$ we compute a second GCD image in $\mathbb{Z}_{13}[x,y]$ using a sparse interpolation:

1. Assume the form of the GCD, substituting unknowns for the coefficients of G_1,

$$H \;\leftarrow\; x^2 \;+\; \alpha y^3 x \;+\; \beta.$$

2. We have at most one unknown per coefficient in our main variable x so we need only one evaluation point. We evaluate at $y = 8$, and compute our univariate GCD image in $\mathbb{Z}_{13}[x]$:

$$\begin{array}{lcl} H(x,8) \bmod 13 & = & x^2 + 5\alpha x + \beta \\ \mathsf{Gcd}(A(x,8),\,B(x,8)) \bmod 13 & = & x^2 + 2x + 9 \end{array}$$

3. Equate by coefficient and solve for the unknowns in $\mathbb{Z}_{13}$:

$$\left.\begin{array}{rcl} 5\alpha & = & 2 \\ \beta & = & 9 \end{array}\right\} \Rightarrow \alpha = 3,\, \beta = 9.$$

4. Substitute these values back into H to get our second bivariate image modulo 13,

$$G_2 \leftarrow \mathsf{subs}(\{\,\alpha = 3, \beta = 9\,\},\, H) \;=\; x^2 + 3y^3x + 9.$$

Apply the Chinese Remainder Theorem to the integer coefficients of G_1 and G_2 to reconstruct our GCD in $\mathbb{Z}[x,y]$,

$$\mathsf{chrem}([\,G_1,\,G_2\,],\,[\,p_1,\,p_2\,]) \;=\; x^2 + 3y^3x + 35.$$

3 The Non-Monic Case

In Example 1, the leading coefficient of our GCD is 1 and so Zippel's algorithm works fine. Consider the problem in $\mathbb{Z}[x,y]$:

$$G = (y+50)x^3 + 100$$
$$A = (y+1) \times G = (y^2 + 51y + 50)x^3 + 100y + 100$$
$$B = (y+2) \times G = (y^2 + 52y + 100)x^3 + 100y + 200$$

Here our GCD is non-monic, G has leading coefficient $y+50$ in the main variable x. Suppose we have computed our first bivariate image modulo 13 using a dense interpolation, $G_1 = (y+11)x^3 + 9$. We proceed to compute a second image using a sparse interpolation working modulo 17. First, assume the GCD has the form $H = (y+\alpha)x^3 + \beta$ for some $\alpha, \beta \in \mathbb{Z}_{17}$. We have at most one unknown per coefficient in x so we evaluate at one random point, $y = 5$ and compute one univariate GCD in $\mathbb{Z}_{17}[x]$:

$$\begin{array}{lcl} H(x,5) \bmod 17 & = & (5+\alpha)x^3 + \beta \\ \mathsf{Gcd}(A(x,5),\,B(x,5)) \bmod 17 & = & x^3 + 8 \end{array}$$

Equating by coefficient, we get the equations, $5 + \alpha = 1$ and $\beta = 8$. Solving for α and β in $\mathbb{Z}_{17}$, gives us the bivariate image, $G_2 = (y+13)x^3 + 8$, which is incorrect. The correct image is $G(x,5) \bmod 17 \equiv (y+16)x^3 + 15$. The problem is that at the bottom level we are computing a univariate GCD modulo p which will always be monic. We are thus always equating the leading coefficient to be 1, giving us incorrect results.

We now present two approaches to dealing with the non-monic case. First, we will look at the problem of normalization and present a new idea for making it work with the sparse interpolation step. Second we will look at the sparse rational function interpolation method.

3.1 Normalization

One way of dealing with the non-monic problem is to normalize the univariate GCD images by multiplying through by some scalar, γ, such that solving the linear systems gives us correct results. Consider again, the previous problem in $\mathbb{Z}[x,y]$. If we take $\gamma(y)$ to be the GCD of the leading coefficients of A and B in our main variable x, we get

$$\gamma = \mathsf{gcd}((y+1)(y+50),(y+2)(y+50)) = y+50 \in \mathbb{Z}[y].$$

We can then normalize our monic univariate GCD image by multiplying through by $\gamma(5) \bmod 17 \equiv 4$ to get $4x^3+15$. Equating by coefficient with $H(x,5) \bmod 17$ we get the equations, $5+\alpha = 4$ and $\beta = 15$. Solving for α and β in $\mathbb{Z}_{17}$, gives us the bivariate image, $G_2 = (y+16)x^3+15$, which is the correct image. Now in this example the GCD of the leading coefficients of our input polynomials happened to be exactly the leading coefficient of our actual GCD, but in general this is not the case. It may happen that

$$\gamma = \mathsf{gcd}(\mathsf{lc}_x(A), \mathsf{lc}_x(B)) = \Delta \times \mathsf{lc}_x(\mathsf{gcd}(A,B))$$

for some factor $\Delta \in \mathbb{Z}[y]$. For example, the first main step in a polynomial factorization algorithm for factoring $f(x,y) \in \mathbb{Z}[x,y]$ is to compute the square-free factorization of $f(x,y)$. This is accomplished by a sequence of GCD computations, the first of which is to compute

$$g = \mathsf{gcd}(f(x,y), \frac{\partial f(x,y)}{\partial x}).$$

In this case $\gamma(y) = \mathsf{lc}_x(f)$ even though $\mathsf{lc}_x(g)$ may be 1. Normalizing our univariate GCDs by γ with this extra factor Δ causes two problems. Firstly, it causes problems in the sparse interpolation step. Secondly, to make the sparse interpolation work, we have to reconstruct a possibly much bigger polynomial where we may lose sparsity.

We are left with the following, we know that there is some scalar for each univariate GCD image that will give us the correct normalization, we just don't know what it is. The solution is to treat these scalars as unknowns and incorporate them into the problem. We illustrate this idea with an example.

Example 2. Let $G, A, B \in \mathbb{Z}[x,y]$ be defined as follows

$$G = (y+50)x^3 + 100$$

$$A = (yx+1) \times G = (y^2+50y)x^4 + (y+50)x^3 + 100yx + 100$$

$$B = (yx+2) \times G = (y^2+50y)x^4 + (2y+100)x^3 + 100yx + 200$$

Using a dense interpolation we get our first GCD image in $\mathbb{Z}_{11}[x,y]$,

$$G_1 = (y+6)x^3 + 1.$$

Working modulo 13 we compute a second GCD image in $\mathbb{Z}_{13}[x,y]$ using sparse interpolation:

1. Assume the form of the GCD, substituting unknowns for the coefficients of G_1,

$$H \leftarrow (\alpha y + \beta)x^3 + \delta.$$

2. We have at most two unknowns per coefficient in our main variable x so we need two evaluation points. We evaluate at $y = 3, 12$, and compute our univariate GCD images in $\mathbb{Z}_{13}[x]$:

$$\begin{aligned} h_1 \leftarrow H(x,3) \bmod 13 &= (3\alpha+\beta)x^3 + \delta \\ h_2 \leftarrow H(x,12) \bmod 13 &= (12\alpha+\beta)x^3 + \delta \end{aligned}$$

$$\begin{aligned} g_1 \leftarrow \mathsf{Gcd}(A(x,3),\, B(x,3)) \bmod 13 &= x^3 + 9 \\ g_2 \leftarrow \mathsf{Gcd}(A(x,12),\, B(x,12)) \bmod 13 &= x^3 + 10 \end{aligned}$$

3. Normalize g_1 and g_2 by some unknown scalars $\gamma_1, \gamma_2 \in \mathbb{Z}_{13}$:

$$\begin{aligned} \gamma_1 \times g_1 &= \gamma_1 x^3 + 9\gamma_1 \\ \gamma_2 \times g_2 &= \gamma_2 x^3 + 10\gamma_2 \end{aligned}$$

4. Equate by coefficient and solve for the unknowns in $\mathbb{Z}_{13}$:

$$\left.\begin{array}{rclcrcl} 3\alpha+\beta &=& \gamma_1, & & 9\gamma_1 &=& \delta \\ 12\alpha+\beta &=& \gamma_2, & & 10\gamma_2 &=& \delta \end{array}\right\} \Rightarrow \alpha = 3\delta, \beta = 7\delta, \gamma_1 = 3\delta, \gamma_2 = 4\delta.$$

5. We get a solution in terms of δ, allowing us to set it arbitrarily. We choose δ such that our GCD image is normalized to have leading coefficient 1, that is, $\delta \leftarrow 3^{-1} \bmod 13 = 9$, which gives us $\alpha = 1$, $\beta = 11$.

6. Substitute these values back into H to get our second bivariate image modulo 13,

$$G_2 \leftarrow \mathsf{subs}(\{\alpha = 1, \beta = 11, \delta = 9\}, H) = (y+11)x^3 + 9.$$

This is precisely our GCD, G mod 13. As you can see, this method results in a larger system to solve in more unknowns. However, it turns out that the system is highly structured, as illustrated in the following example.

Example 3. Consider a GCD of the form

$$H = x^4\left(a1\; yz^3 + a2\; z^2\right) + x\left(b1\; yz + b2\right) + c1\; y^2z - c2\; yz^2.$$

And say the GCD we are looking at is given by

$$G = x^4\left(z^3y - 5\,z^2\right) + x\left(9\,yz + 11\right) + 3\,zy^2 - 7\,yz^2.$$

Working modulo 23 we evaluate at the points $(y,z) = (14,12), (4,3), (16,21)$ to get the following:

$$\begin{aligned} h_1 \leftarrow H(x,14,12) \bmod 23 &= (19\;a1 + 6\;a2)\,x^4 + (7\;b1 + b2)\,x + 6\;c1 + 8\;c2 \\ h_2 \leftarrow H(x,4,3) \bmod 23 &= (16\;a1 + 9\;a2)\,x^4 + (12\;b1 + b2)\,x + 2\;c1 + 10\;c2 \\ h_3 \leftarrow H(x,16,21) \bmod 23 &= (10\;a1 + 4\;a2)\,x^4 + (14\;b1 + b2)\,x + 17\;c1 + 5\;c2 \end{aligned}$$

$$\begin{aligned} g_1 \leftarrow G(x,14,12) \bmod 23 &= x^4 + 10x + 10 \\ g_2 \leftarrow G(x,14,12) \bmod 23 &= x^4 + 7x + 18 \\ g_3 \leftarrow G(x,14,12) \bmod 23 &= x^4 + 7x + 19 \end{aligned}$$

Scaling the images g_1, g_2 and g_3 by the unknown multipliers m_1, m_2 and m_3, respectively, and equating by coefficient we get the following system to solve in $\mathbb{Z}_{23}$:

$$\begin{array}{rclcrclcrcl} 19\;a1 + 6\;a2 &=& m1, & & 7\;b1 + b2 &=& 10m1, & & 6\;c1 + 8\;c2 &=& 10m1 \\ 16\;a1 + 9\;a2 &=& m2, & & 12\;b1 + b2 &=& 7m2, & & 2\;c1 + 10\;c2 &=& 18m2 \\ 10\;a1 + 4\;a2 &=& m3, & & 14\;b1 + b2 &=& 7m3, & & 17\;c1 + 5\;c2 &=& 19m3 \end{array}$$

Looking at this system as a matrix:

$$\begin{bmatrix} 10 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 16 & 9 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 19 & 6 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 14 & 1 & 0 & 0 & 0 & 0 & -7 \\ 0 & 0 & 12 & 1 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 7 & 1 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 17 & 5 & 0 & 0 & -19 \\ 0 & 0 & 0 & 0 & 2 & 10 & 0 & -18 & 0 \\ 0 & 0 & 0 & 0 & 6 & 8 & -10 & 0 & 0 \end{bmatrix},$$

we see that we have a block-form matrix, for which a fast solver has been constructed.

It should be noted that if the GCD has a content in the main variable, this algorithm will fail. Thus the content must be computed and removed at the top level of the algorithm. Unfortunately this is not sufficient for avoiding problems. It may happen that a certain choice of prime or evaluation point may introduce a content. Consider, for example, the GCD, $G = (y^2 + y + 1)x^2 + y^2 + 12y + 1$. G has no content in the main variable x over $\mathbb{Z}$ but if we choose the prime $p = 11$ we introduce the content $y^2 + y + 1$. There is no inexpensive way to detect at what level the content was introduced. We propose to deal with this problem as follows. At each level of the computation, keep track of the number of successes, s, and the number of failures, f. If at any point $f > s$, then return FAIL to the next level up.

3.2 Sparse Rational Function Interpolation

An alternative way of handling the non-monic case is to use sparse rational function interpolation. The idea is as follows. If we are computing the GCD of two polynomials in $\mathbb{Z}[x, w, y, z]$, we take x as our main variable and compute the monic GCD in $\mathbb{Z}(w, y, z)[x]$, of the form:

$$x^n + \sum_{i=0}^{n-1} \frac{a_i(w, y, z)}{b_i(w, y, z)} x^i,$$

where $a_i, b_i \in \mathbb{Z}[w, y, z]$. The idea is to interpolate the rational function coefficients using a sparse interpolation process. For example, if our GCD is $(y + 14)yx^3 + 12y^2x + y + 14$, then we compute the monic GCD

$$x^3 + \frac{12y}{y + 14} x + \frac{1}{y}.$$

We can then recover the non-monic GCD by multiplying through by the least common multiple of the denominators. In our example, we multiply through by $\mathsf{lcm}(y + 14, y) = (y + 14)y$ to get our non-monic GCD $(y + 14)yx^3 + 12y^2x + y + 14$.

To illustrate how the sparse rational function reconstruction works, consider the following. Suppose one of the rational function coefficients is

$$C = \frac{*w^3 + *zy^2}{*z^2 + *y^2 + wy^3},$$

here $*$ indicates an integer. Suppose we have reconstructed C at $w = 5$ to get

$$C_1 = \frac{* + *zy^2}{*z^2 + *y^2 + y^3}.$$

Notice we have normalized the leading coefficient of the denominator to be 1, essentially dividing through by w. We then assume the form to be:

$$H = \frac{\alpha(w) + \beta(w)zy^2}{\delta(w)z^2 + \gamma(w)y^2 + y^3},$$

where $\alpha(w), \beta(w), \delta(w), \gamma(w)$ are rational functions in w. We have 4 unknowns so we need 4 equations to solve for the next image, C_2. We do this for as many w values as we need. We then do rational function interpolation in w to get:

$$\frac{*w^2 + \frac{*}{w}zy^2}{\frac{*}{w}z^2 + \frac{*}{w}y^2 + y^3}$$

Clearing the fractions in w gets us what we want:

$$\frac{*w^3 + *zy^2}{*z^2 + *y^2 + wy^3}.$$

Example 4. Let $G, A, B \in \mathbb{Z}[x, y]$ be defined as follows

$$G = (y + 14)yx^3 + 12y^2x + y + 14$$

$$A = (yx + 1) \times G \quad \textit{and} \quad B = (yx + 2) \times G$$

Working modulo $p_1 = 11$ we compute our first monic GCD image in $\mathbb{Z}_{11}(y)[x]$ using a dense rational function interpolation.

Note, rather than doing the rational function interpolations by solving systems of linear equations, which would have a cubic time complexity in the number of evaluation points, we use polynomial interpolation and *rational function reconstruction* which both have only quadratic time complexity. Details of the algorithm used by Maple's `Ratrecon` command below for rational function reconstruction may be found in [3].

1. We need 4 evaluation points to reconstruct the rational functions in y. Evaluate at four random points, $y = 1, 4, 9, 3$, and compute our univariate GCD images in $\mathbb{Z}_{11}[x]$:

$$\begin{array}{rcll} g_1 & \leftarrow & \mathsf{Gcd}(A(x,1),\ B(x,1)) \bmod 11 & = x^3 + 3x + 1 \\ g_2 & \leftarrow & \mathsf{Gcd}(A(x,4),\ B(x,4)) \bmod 11 & = x^3 + 10x + 3 \\ g_3 & \leftarrow & \mathsf{Gcd}(A(x,9),\ B(x,9)) \bmod 11 & = x^3 + 9x + 5 \\ g_4 & \leftarrow & \mathsf{Gcd}(A(x,3),\ B(x,3)) \bmod 11 & = x^3 + 6x + 4 \end{array}$$

2. Interpolate in y,

$$G_1 \leftarrow \mathsf{Interp}([1,4,9,3],[g_1,g_2,g_3,g_4],y) \bmod 11 = x^3 + (y^3+2y^2+8y+3)\,x + 6y^3+8y^2+7y+2.$$

3. Apply rational reconstruction to the coefficients in x:

$$m \leftarrow \mathsf{expand}((y-1)\times(y-4)\times(y-9)\times(y-3)) \bmod 11,$$

$$G_1 \leftarrow \mathsf{Ratrecon}(G_1, m, y) \bmod 11 = x^3 + \frac{y}{y+3}x + \frac{1}{y}.$$

Working modulo $p_2 = 13$ we compute a second monic GCD image in $\mathbb{Z}_{13}(y)[x]$ using a sparse rational function interpolation:

1. Assume the form of the GCD, substituting unknowns for the integer coefficients of the rational functions of G_1,

$$H \leftarrow x^3 + \frac{\alpha y}{y+\beta}x + \frac{\delta}{y}.$$

2. We have at most two unknowns per coefficient in our main variable x so we need two evaluation points. We evaluate at $y = 1, 6$, and compute our univariate GCD images in $\mathbb{Z}_{13}[x]$:

$$\begin{array}{rcl} h_1 \leftarrow H(x,1) \bmod 13 & = & x^3 + \frac{\alpha}{1+\beta}\,x + \frac{\delta}{1} \\ h_2 \leftarrow H(x,6) \bmod 13 & = & x^3 + \frac{6\,\alpha}{6+\beta}\,x + \frac{\delta}{6} \end{array}$$

$$\begin{array}{rcl} g_1 \leftarrow \mathsf{Gcd}(A(x,1),\, B(x,1)) \bmod p_2 & = & x^3 + 6x + 1 \\ g_2 \leftarrow \mathsf{Gcd}(A(x,6),\, B(x,6)) \bmod p_2 & = & x^3 + x + 11 \end{array}$$

3. Equate by coefficient and solve for the unknowns in $\mathbb{Z}_{13}$:

$$\left.\begin{array}{rclrcl} 6 & = & \frac{\alpha}{1+\beta}, & 1 & = & \frac{\delta}{1} \\ 1 & = & \frac{6\,\alpha}{6+\beta}, & 11 & = & \frac{\delta}{6} \end{array}\right\} \Rightarrow \alpha = 12,\ \beta = 1,\ \delta = 1$$

4. Substitute these values back into H to get our second monic image in $\mathbb{Z}_{13}(y)[x]$,

$$G_2 \leftarrow \mathsf{subs}(\{\alpha = 12, \beta = 1, \delta = 1\}, H) \;=\; x^3 + \frac{12\,y}{y+1}\,x + \frac{1}{y}.$$

Apply the Chinese Remainder Theorem to the integer coefficients of the rational functions of G_1 and G_2 to reconstruct our monic GCD in $\mathbb{Z}(y)[x]$,

$$\mathsf{chrem}([G_1, G_2], [11, 13]) \;= x^3 + \frac{12\,y}{y+14}\,x + \frac{1}{y}.$$

Finally, we clear the fractions to get our non-monic GCD in $\mathbb{Z}[x, y]$, $(y+14)yx^3 + 12y^2x + y + 14$.

Since the first approach requires that the inputs don't have contents in the main variable, and the second approach reconstructs the monic gcd, we need to compute and remove the content in the main variable at the top level of both algorithms. Computing the content can be very expensive as it may involve doing many multivariate GCD computations in one less variable than the actual GCD problem, $\mathsf{content}(G, x) = \mathsf{gcd}(\mathsf{content}(A, x), \mathsf{content}(B, x))$, where $\mathsf{content}(A, x) = \mathsf{gcd}(\mathsf{coeffs}(A, x))$. A more efficient way of computing the content is as follows.

1. Let $c = [\mathsf{coeffs}(A, x), \mathsf{coeffs}(B, x)]$.

2. Let c_{min} be the coefficient of minimum total degree.

3. Take a random linear combination of the coefficients:

$$f = \sum_{i=1}^{nops(c)} \alpha_i \times c_i, \ \alpha_i \in \mathbb{Z}$$

4. $cont\_g \leftarrow \mathsf{gcd}(f, c_{min})$

5. If $cont\_g \mid A$ and $cont\_g \mid B$ then we have our content.

6. Otherwise go back to 3. and try again.

4 Conclusion

We have done an implementation in Maple using the "recden" data structure of Zippel's algorithm extended to work for monic input over finite fields and algebraic number fields, as well as for non-monic input over the integers using the sparse rational function interpolation method. We are currently working on an implementation of the modified normalization technique in the "recden" data structure. The aim is to get a running time comparison of the two methods in Maple. We plan to extend these methods to work for non-monic problems over finite fields and algebraic number fields.

References

[1] W. S. Brown. On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. *J. ACM* **18** (1971), 478-504.

[2] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Publ., Boston, Massachusetts, USA, 1992.

[3] M. van Hoeij, M. Monagan. Algorithms for Polynomial GCD Computation over Algebraic Function Fields. To appear in *Proceedings of ISSAC '04*, ACM Press, 2004.
Preprint available from http://www.cecm.sfu.ca/CAG/products.html .

[4] M. Monagan, J. Ales, J. de Kleine, C. Pastro, A. Wittkopf. Data Structures and Algorithms for Polynomials. MITACS Research Report, November, 2000.
Available from http://www.cecm.sfu.ca/CAG/products.html .

[5] R. Zippel. *Probabilistic Algorithms for Sparse Polynomials*. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1979.

[6] R. Zippel. Interpolating Polynomials from their Values. *J. Symbolic Comput.* **9**, 3 (1990), 375-403.