A Subdivision Method for Computing Nearest Gcd with Certification

Guillaume Chèze[a], André Galligo[b,c], Bernard Mourrain[c], Jean-Claude Yakoubsohn[a]

[a]*Institut Mathématique de Toulouse, équipe MIP*
Université Paul Sabatier
118 route de Narbonne
31062 Toulouse, France
`guillaume.cheze@math.univ-toulouse.fr`
`jean-claude.yakoubsohn@math.univ-toulouse.fr`

[b]*Laboratoire J.A. Dieudonné,*
UMR CNRS 6621, Université de Nice Sophia-Antipolis,
Parc Valrose, 06108 Nice Cedex 02
`galligo@math.unice.fr`

[c] *GALAAD, INRIA Méditerranée*
2004 route des Lucioles
BP 93, 06902 Sophia Antioplis
France
`Bernard.Mourrain@inria.fr`

Abstract

A new subdivision method for computing the nearest univariate gcd is described and analyzed. It is based on an exclusion test and an inclusion test. The exclusion test in a cell exploits Taylor expansion of the polynomial at the center of the cell. The inclusion test uses Smale's α-theorems to certify the existence and unicity of a solution in a cell.

Under the condition of simple roots for the distance minimization problem, we analyze the complexity of the algorithm in terms of a condition number, which is the inverse of the distance to the set of degenerate systems.

We report on some experimentation on representative examples to illustrate the behavior of the algorithm.

1. Introduction

Computing an approximate gcd of two polynomials is a fundamental and difficult problem in symbolic-numeric computation. It has been extensively studied in the Computer Algebra community, following different strategies: [6, 22, 19, 8, 13, 11, 2, 10, 3, 4, 20, 26, 15, 14, 18, 24].... The problem is usually formulated as finding a perturbation of a pair of polynomials (f, g) within a given ball of radius ε, such that the perturbed pair (p, q) has a non-trivial exact gcd (of highest possible degree) in this ball.

In our approach, we reformulate the problem as an optimization problem: Given the polynomial pair (f, g) of a given degree, we look at the nearest polynomial pair (p, q) of the same degree which has a non-trivial exact gcd. The ε-approximate gcd problem has a solution iff this nearest pair with a non-trivial gcd is within a distance ε.

This paper is an extended version of the conference paper [7]. In the first section, we present the optimization problem that we are going to solve, we state our complexity main result and mention the related approaches in the literature for approximate gcd computation. In section 2, we describe the main ingredients of our subdivision method. In section 3, we analyze its complexity using α-theory. Finally, in section 4, we report of some experimentation to illustrate the behavior of the algorithm.

1.1. A minimization problem

As usual, we denote by $\mathbf{R}_d[z]$ and $\mathbf{C}_d[z]$ the vector spaces of univariate polynomials of degree less or equal to d, with coefficients in $\mathbf{R}$ and $\mathbf{C}$. We denote by $f = (f_0, \ldots, f_d)^T$ the coefficients of the polynomial $f(z) = \sum_{k=0}^{d} f_k z^k \in \mathbf{R}_d[z]$. The 2-norm of f is $||f||^2 = \sum_{k=0}^{d} f_k \bar{f}_k$.

We address the nearest gcd problem, i.e. the following minimization problem:

Problem 1.1 *Given two degree d polynomials $f(z)$ and $g(z)$, find two degree d polynomials $p(z)$ and $q(z)$ with a non trivial gcd, which are solutions of the minimization problem*

$$\min_{\substack{p,q \in \mathbf{C}_d[z] \\ Resultant_d(p,q)=0}} ||f-p||^2 + ||g-q||^2, \tag{1}$$

Here $\text{Resultant}_d(p,q)$ is the resultant of p and q when they are considered as polynomials of degree d. We recall that $\text{Resultant}_d(p,q) = 0$ if and only if $\gcd(p,q)$ is non-trivial or $\deg(p) < d$ and $\deg(q) < d$. In the last case p and q have a common root at infinity, see Example 2 in Section 4.
In our formulation of the nearest gcd problem, we look for the nearest pair (p,q) of degree at most d with a non-trivial gcd. In Example 1 in Section 4 we will see that if $\deg f = 7$ and $\deg g = 8$ then we can find p and q with degree 8. Thus the degree of p can be bigger than the degree of f.

N.K. Karmarkar and Y.N. Lakshman in [15] reduced that problem to another minimization problem :

Problem 1.2

$$\min_{z \in \mathbb{P}(\mathbf{C})} \frac{f(z)\overline{f(z)} + g(z)\overline{g(z)}}{\sum_{k=0}^{d} z^k \overline{z}^k}. \tag{2}$$

Moreover, a global minimum z_0 of (1.2) is a root of a nearest gcd and determines the polynomials p and q of (1.1) by the formulas:

$$\begin{aligned} p(z) &= f(z) - \frac{f(z_0)}{\sum_{k=0}^{d} \overline{z_0}^k z_0^k} \sum_{k=0}^{d} \overline{z_0}^k z^k \\ q(z) &= g(z) - \frac{g(z_0)}{\sum_{k=0}^{d} \overline{z_0}^k z_0^k} \sum_{k=0}^{d} \overline{z_0}^k z^k. \end{aligned}$$

We observe that with the change of variables $z = X + iY$, the problem amounts to minimize a homogeneous rational bivariate function

$$\mathcal{F}(z) = F(X,Y) = \frac{N(X,Y)}{D(X,Y)}.$$

So we focus on the resolution of this task.

1.2. Notations

We denote by $G(X,Y) = (G_1(X,Y), G_2(X,Y))$ the couple of numerators of the gradient $\nabla F(X,Y)$. We denote by $P_\mu = N - \mu D$ where $\mu \in \mathbf{R}$ and $F = \frac{N}{D}$.

For $G(X,Y) = (G_1(X,Y), G_2(X,Y))$ with

$$G_k(X,Y) = \sum_{l=0}^{\deg G_k} \sum_{i+j=l} G_{k;i,j} X^i Y^j \in \mathbb{C}[X,Y], k = 1,2,$$

we define the (Bombieri) norm $\|G\|_B$ by:

$$\|G\|_B^2 = \sum_{k=1,2} \sum_{l=0}^{\deg G_k} \sum_{i+j=l} G_{k;i,j} \overline{G}_{k;i,j} \frac{i!j!}{\deg G_k!}.$$

$A_F := \{(x_i, y_i) \,|\, i = 1,..,n\}$ is the set of global minima of $F(X,Y)$. We assume that this set is finite.
Z_H is the set of zeroes of a polynomial function $H(X,Y)$ and $d(x,y,Z_H)$ is the Euclidean distance from $(x,y) \in \mathbf{R}^2$ to Z_H. The function $H(X,Y)$ will be specialized to $P_\mu(X,Y)$ or to $G_i(X,Y)$, $i = 1,2$.
Let S or $S(x,y,r)$ be a square centered at (x,y) and of radius r.

1.3. Our approach

We use a bisection algorithm, based on an exclusion test and an inclusion test (see below), applied simultaneously to:

1- the polynomial $P_\mu(X,Y) = N(X,Y) - \mu D(X,Y)$

2- the system $G(X,Y) = (G_1(X,Y), G_2(X,Y))$.

Our bisection algorithm will iteratively update a list of retained squares and a list of approximate global minima of F. We prove a complexity result in Section 3.
We use G_1 and G_2 to compute local extrema. The polynomial P_μ is used to exclude "bad" squares. Indeed, during our algorithm μ is a candidate for the minimum of F and we set $\mu = N(x_0,y_0)/D(x_0,y_0)$ where (x_0,y_0) is a center of a square. Thus, if $P_\mu(x,y) > 0$ in a square S then $F(x,y) > \mu$, and we cannot find a global minimum in S.

We follow the approach initiated by S. Smale and his co-worker in a series of papers (see e.g. [5] and the references therein) relying on their celebrated α and γ-theorems.

In section 2 we will briefly review some of these notions and recall the definitions of α, β, γ, we also set $\gamma(G, A_F) := \max_{(x,y)\in A_F} \gamma(G; x, y)$. After that, we provide a precise quantitative definition for a point (x,y) to be, in our setting, an approximate global minimum of $F(x,y)$. Then thanks to the $\gamma-$theorem, the approximation is sufficiently good to imply that a Newton iteration converges quadratically towards a global minimum.

To state our result on the complexity analysis of the subdivision method, we need the following notations:

- $\sigma_F := \frac{1}{\min_{(x,y)\in A_F} d_F(DG(x,y), \Sigma)}$ where d_F is the Frobenius distance and Σ is the set of singular matrices.

- $V_\epsilon(G)$ is the tubular neighborhood of the zero set $G(x,y) = 0$;

- $\mathcal{N}$ is the maximal number of connected components of $U_\epsilon := V_\epsilon(G_1) \cap V_\epsilon(G_2)$, for all $\epsilon > 0$;
- for all $\epsilon > 0$, ϵK is bounding the radius of a connected component of U_ϵ.
- $\mathcal{J}(G, F, r_0) := \left\lceil \log_2(\frac{K\, r_0 \gamma(G, A_F)}{\delta_0}) \right\rceil$ where $r_0 > 0$ and δ_0 is the smallest positive root of $(13 - 3\sqrt{17})\ (2u^2 - 4u + 1)^2 - 4u = 0$.

Theorem 1.3 (Global minimum) *Assume that all the elements of A_F are regular points of the system $G(x, y) = 0$ and are contained in some square $S_0 := S(x_0, y_0, r_0)$.*
Denote by R_j the set of retained squares at step j. Then, for $j \geq J = \mathcal{J}(G, F, r_0)$, all the points of R_j are approximate global minima of $F(x, y)$. That is to say the Newton iteration applied to G from a point in R_j converges quadratically to a global minimum of F.

Theorem 1.4 (Complexity analysis) *For the same notations and assumptions as in above theorem, we have the following properties:*

1- The number of exclusion tests is bounded by $1 + J\, \mathcal{N}\, K^2$.

2- When d tends to infinity, J belongs to $O\Big(d\, \log(r_0) + \log(\sigma_F) + \log(||G||_B) + \log(K) + \log(d)\Big)$. The number of exclusions steps belongs to $O\Big(\mathcal{N}\, K^2\, (d\, \log(r_0) + \log(\sigma_F) + \log K + \log(||G||_B))\Big)$.

Remark:

1. $\sigma_F = \frac{1}{\min_{(x,y)\in A_F} d_F(DG(x,y), \Sigma)}$ is related to the condition number of the system. That is why we keep this constant in the big O notation.

Unfortunately, we do not know how to bound this constant in terms of the degree of f and g. The computation of this kind of bound is a difficult problem.

1. $\sharp A_F \leq \sum_{(p,q) \text{ solution of Problem 1.1}} \deg\, gcd(p, q)$.
   Thus if Problem 1.1 has a unique solution then $\sharp A_F \leq \deg\, gcd(p, q) \leq d$.
2. In this paper, in order to apply α and $\gamma-$theorems, we suppose that all the elements of A_F are regular points of the system $G(x, y) = 0$. We could avoid this hypothesis, instead of using α and $\gamma-$theorems, by relying on the tools developed in [12].

1.4. Other approaches in the literature

The nearest GCD problem has been studied with different approaches and with other formulations by many authors.

1.4.1. Algebraic approach via Euclid's algorithm, resultant and subresultant

The first papers, see for example Brown [6], on the complexity of Euclid's algorithm only works with exact coefficients. Then the numerical case has been successively considered by Schönhage [22], Noda and Sasaki [19], Corless, Gianni, Trager and Watt [8], Hribernig and Stetter [13], Emiris, Galligo and Lombardi [11], Beckermann, Labahn [2]. These authors consider the so-called near GCD or the ϵ-GCD problem. The singular value decomposition was first applied to the Sylvester matrix, in [8], and later applied to the subresultant matrices, in [10, 11] in order to get a certification when a condition (depending on the level of accuracy) is satisfied. An efficient implementation is done in [21].

1.4.2. Padé approximation and structured matrices approach

See the book of Bini and Pan [3] and the bibliography therein, and more recently Boito, Bini [4].

1.4.3. Rootfinding and cluster root approach

Pan [20] and, more recently, Zeng [26] use root finding and least squares methods.

1.4.4. Optimization approach

The resolution of problem 1.2 is the main and the most "time-consuming" step of the method propose in [15] that we aim to improve. The authors rely on techniques from Arnon-McCallum [1] and Manocha-Demmel [16]. The second paper is based on resultant for expressing the intersection of two curves and on numerical computation of eigenvalues and eigenvectors by QR iterations. Therefore the expected running time is in $O(p^3)$ where p is the product of degrees of two curves, hence the complexity of the algorithm is at least in $O(d^6)$.

Kaltofen and his co-workers [14] determine approximate GCDs from methods based on structured total least square (STLN). The STLN is an iterative method of the family of Gauss-Newton methods. The authors describe its application to the case of the Sylvester matrix associated to the input polynomials f and g, then show its interest and efficiency by producing the results of experiments. However, since the starting point of their Gauss-Newton like method is not precised, the method can diverge. This is an important drawback.

Nie-Demmel-Gu [18] uses a sum of squares (SOS) technique, the resolution relies on semi definite programming (SDP). For the nearest GCD problem, this yields linear matrix inequalities (LMI) whose size s is $O((d+2)(d+1))$ whose complexity of resolution by projective algorithm is in $O(s^3)$ [17]. So, one ends up with a complexity in $O(d^6)$.

More recently, Terui [24] uses a generalization of the gradient-projection method proposed by Tanabe [25]. Then he obtains a fast algorithm for solving a constrained minimization problem related to the approximate gcd problem. Unfortunately, there exists no complexity results for this approach. Furthermore, in this method the degree of the approximate gcd is given in input.

2. The proposed bisection method

2.1. Principles

We propose a bisection method, described below, to approximate the global minima of the function $F(X, Y)$ defined in the introduction. *In the next section, a procedure for computing an initial square is given* ; so we suppose here that a square $S(x_0, y_0, r_0)$ containing all the global minima is known.

The bisection method is based on an exclusion test and an inclusion test. An exclusion test, denoted hereafter by $E(F, S)$ or $E^+(F, S)$, is defined on the set of squares and returns true if the function F has no zero in the square S or false if it might have a zero. Examples of exclusion tests are provided in section 3.

An inclusion test $I(G, S)$ is needed to numerically prove the existence of a local minimum: it takes $G = (G_1, G_2)$ and a square S then returns true if there exists a ball $B(x^*, y^*, r^*)$, containing S, which contains one zero of G, in that case it also returns (x^*, y^*, r^*), otherwise it returns false.

Our definition of approximate minima is based on γ-theorem and α-theorem of Smale [23], [5], [12], applied to the system $G(x, y)$. We first introduce some quantities and the corresponding classical notations:

- $\beta := \beta(G; x, y) = ||DG(x, y)^{-1}G(x, y)||$
- $\gamma := \gamma(G; x, y) = sup_{k\geq 2}\left(\frac{1}{k!}||DG(x, y)^{-1}D^kG(x, y)||\right)^{1/(k-1)}$
- $\alpha := \alpha(G; x, y) = \beta\gamma$.
- $\gamma(G; A_F) = max_{(x,y)\in A_F}\gamma(G; x, y)$.

Smale's $\gamma-$theorem [5] states:

Theorem 2.1 *Let (x^*, y^*) be a zero of G and suppose that $DG(x^*, y^*)$ is invertible. If*

$$\|(x, y) - (x^*, y^*)\|\gamma(G, x^*, y^*) \leq \frac{3-\sqrt{7}}{2}$$

then the Newton iteration from (x, y) converges quadratically to (x^*, y^*).

This leads to the following definition and quantitative results on the convergence of Newton scheme.

Definition 2.2 *The point (x, y) is an approximate global minimum of $F(x, y)$ if*

$$d((x, y), A_F) \leq \frac{3-\sqrt{7}}{2\gamma(G, A_F)}.$$

Under this condition, the $\gamma-$theorem asserts that the Newton iteration from any point in the ball $B(x^*, y^*, \frac{3-\sqrt{7}}{2\gamma(G,A_F)})$ where $(x^*, y^*) \in A_F$, converges quadratically towards (x^*, y^*).
The following result gives a sufficient condition for a point to be a good starting point for the Newton iteration. This result is called α-theorem.

Theorem 2.3 *If $\alpha < (13 - 3\sqrt{17})/4$ then G has one and only one zero in the open ball $B(x, y, \sigma(x, y))$ with*

$$\sigma(x, y) = \frac{1 + \alpha - \sqrt{1 - 6\alpha + \alpha^2}}{4\gamma} \leq \frac{2-\sqrt{2}}{2\gamma}.$$

and the Newton iteration from (x, y) converges to (x^*, y^*). Furthermore, we have

$$\|(x^*, y^*) - N_G^k(x, y)\| \leq \frac{5-\sqrt{17}}{4\gamma}\left(\frac{1}{2}\right)^{2^k-1},$$

where $N_G^k(x, y)$ is the k-th iterate of the Newton iteration applied to G with starting point (x, y).

The following result gives a radius r^* of a ball centered in (x^*, y^*) such that every point in $B(x^*, y^*, r^*)$ satisfies the hypothesis of the α-theorem.

Proposition 2.4 *Let (x^*, y^*) be a zero of G and suppose that $DG(x^*, y^*)$ is invertible. Let δ_0 be the smallest positive root of $u - \frac{13-3\sqrt{17}}{4}\Psi(u)^2$, where $\Psi(u) = 2u^2 - 4u + 1$. If*

$$\|(x, y) - (x^*, y^*)\|\gamma(G, x^*, y^*) \leq \delta_0$$

then

$$\alpha(G, x, y) \leq \frac{13-3\sqrt{17}}{4}.$$

Remark: $0.07 \leq \delta_0 \leq 0.08$.

Proof. We use the following inequality, see [23]

$$\alpha(G,x,y) \leq \frac{u}{\Psi(u)^2},$$

where $u = \|(x,y)-(x^*,y^*)\|\gamma(G,x^*,y^*)$. Furthermore, we have $u/\Psi(u)^2 \leq (13-3\sqrt{17})/4$ for $u \in [0;\delta_0]$ and this gives the desired result. □

In our complexity study we will need a bound on $\gamma(G,A_F)$. The following proposition will be useful.

Proposition 2.5 *Let* $\deg G = \max(\deg G_1, \deg G_2)$, $(x,y) \in S(0,0,r_0)$, *Then*

$$\gamma(G;x,y) \leq \|G\|_B \, \|DG(x,y)^{-1}\| (\deg G)^2 (1+2r_0^2)^{(\deg G-2)/2}.$$

Proof. We set some notations: $\|(x,y)\|_1 = (1+x^2+y^2)^{1/2}$, $\Delta(a_i)$ is the diagonal matrix with coefficients a_i.

By [23, Proposition 3], we have the following bound:

$$\gamma(G;x,y) \leq \frac{\mathcal{C}(G;x,y)\deg G^{3/2}}{2\|(x,y)\|_1},$$

where $\mathcal{C}(G;x,y) = \|G\|_B \|DG(x,y)^{-1}\Delta(\deg(G_i)^{1/2}\|(x,y)\|_1^{\deg G_i - 1})\|$.
We use the following classical inequality $\|AB\| \leq \|A\|\|B\|$ to conclude. □

Let a threshold $\epsilon > 0$ be given, the output of this bisection method will be a set (eventually empty) $Z = \{(x_i^*, y_i^*, r_i^*, \mu_i^*)\}$, such that :

1- the (x_i^*, y_i^*)'s are approximate global minima of $F(x,y)$ and $\mu_i^* = F(x_i^*, y_i^*)$.

2- the ball $B(x_i^*, y_i^*, r_i^*)$ contains one and only one zero of $G(x,y)$ where $r_i^* = \sigma(x_i^*, y_i^*)$.

3- If $i \neq j$ then $B(x_j^*, y_j^*, r_j^*) \cap B(x_i^*, y_i^*, r_i^*) = \emptyset$ and $|\mu_i^* - \mu_j^*| < \epsilon$.

2.2. Computation of an initial square

Lemma 1 in [15] gives us a bound for the initial square:

Lemma 2.6 *Let* $f = \sum_{k=0}^{d_1} f_k z^k$, $g = \sum_{k=0}^{d_2} g_k z^k$, *where* $f_{d_1} \neq 0$ *and* $g_{d_2} \neq 0$. *Let F be the rational bivariate function corresponding to the approximate gcd problem. Let* $(x,y) \in A_F$ *then* $\|(x,y)\| \leq 5 \max\left(\frac{\|f\|^2}{f_{d_1}^2}, \frac{\|g\|^2}{g_{d_2}^2}\right)$.

2.3. Sketch of algorithm

The algorithm consists of an initialization followed by a while loop with an internal for loop. We call step k, the k-th step of the while loop.

Proposition 2.7 *Assume there exists $\epsilon > 0$ such that for all square $S(x,y,\epsilon) \subset S_0$ we have :*

1- the inclusion test is true if $S(x,y,\epsilon)$ contains a zero of G.

2- the exclusion test is true if $S(x,y,\epsilon)$ does not contain a zero of G.

Then we have :

Algorithm 2.1: Approximate gcd

```
Input: $F = N/D$, $G = (G_1, G_2)$, an initial square $S_0$ and a threshold $\epsilon > 0$, as described
       above.
◇ Create a set of squares $L := \{S_0\}$, a set of solutions $Z := \infty$ and a value (the minimum
to be updated) $\mu := \mathcal{F}(\infty)$.
◇ While $L$ is not empty do
    • Compute $\delta = \min F(x_i, y_i)$ where the $(x_i, y_i)$'s are the centers of squares of $L$.
    • If $\delta < \mu$ then $\mu := \delta$.
    • For each square $S$ of $L$ perform the exclusion tests $E^+(P_\mu, S)$, $E(G_1, S)$ and $E(G_2, S)$.
      If at least one of these exclusion tests is true then remove $S$ from $L$ ; else, perform an
      inclusion test $I(G, S)$.
    • If it returns false then divide $S$ in 4 equal squares ;
      else an approximate local minimum $(x^*, y^*)$ is provided, it is the unique zero of $G(x, y)$
      in the ball $B(x^*, y^*, r^*)$.
        – If $\mu^* = F(x^*, y^*) > \mu + \epsilon$ then remove $S$ ;
          else update the set $Z$ as follows:
            ∗ If $\mu^* = F(x^*, y^*) < \mu$ then put $\mu := \mu^*$. For each element $(x_i^*, y_i^*, r_i^*, \mu_i^*)$ of $Z$
              do
                · If $\mu < \mu_i^*$ and $|\mu - \mu_i^*| > \epsilon$ then remove the solution $(x_i^*, y_i^*, r_i^*, \mu_i^*)$ from $Z$ ;
                · If for any element $(x_i^*, y_i^*, r_i^*, \mu_i^*)$ of $Z$, $|\mu - \mu_i^*| < \epsilon$ and $(x^*, y^*)$ is not in
                  the ball $B(x_i^*, y_i^*, r_i^*)$ then add $(x^*, y^*, r^*, \mu^*)$ to $Z$.
Output: The set $Z$ of approximate global minima of $F$.
```

1- The algorithm stops.

2- Let $\mu(\epsilon)$ be the value of μ at the end of the algorithm with input ϵ. Then $\lim_{\epsilon\to 0}\mu(\epsilon) = \min_{(x,y)\in S_0} F(x,y)$.

Proof. The point 1 holds by construction under these assumptions. For point 2, the radius $r(\epsilon)$ of the retained squares in the last iteration decreases. Hence from the continuity of F, we obtain $\lim_{\epsilon\to 0}\mu(\epsilon) = \min_{(x,y)\in S_0} F(x,y)$. □

3. Complexity analysis

3.1. Exclusion test

Let H be a polynomial in $\mathbf{R}[X,Y]$ and denote by $D^k H(x,y)$ the homogeneous part of degree k of the Taylor expansion of H at the point (x,y).

Let $S(x_0,y_0,r_0)$ be a square. To define an exclusion function $E(H,S)$, we rely on the following expression and lemma:

$$M_H(x_0,y_0,r_0) = |H(x_0,y_0)| - \sum_{k\geq 1} \frac{||D^k H(x_0,y_0)||}{k!} r_0^k.$$

Lemma 3.1 *If $M_H(x_0,y_0,r_0) > 0$ then the closed square $\bar{S}(x_0,y_0,r_0)$ does not contain any zero of the polynomial $H(x,y)$.*

Proof. The proof follows from Taylor formula and a simple inequality. □

A key to analyze the complexity of the algorithm of section 2 is the following lemma, see [9].

Lemma 3.2 *Let $H \in \mathbf{R}[X,Y]$ be a polynomial of degree e, consider the associated algebraic variety $Z_H = \{(x,y) \in \mathbf{R}^2 : H(x,y) = 0\}$.*

Let $L_e = \dfrac{2^{1/e}-1}{\sqrt{2}}$ and $m_H(x,y)$ be the function implicitly defined by

$$M_H(x,y,m_H(x,y)) = 0.$$

Then $m_H(x,y)$ is related to the distance $d(x,y,Z_H)$ by the following inequalities:

$$L_e.d(x,y,Z_H) \leq m_H(x,y) \leq d(x,y,Z_H).$$

The exclusion tests to be used for the algorithm of section 2 are defined for a polynomial P by:

1- $E(P,S)$ is true if $M_P(x,y,r) > 0$,

2- $E^+(P,S)$ is true if $M_P(x,y,r) > 0$ and $P(x,y) > 0$.

Since the degree of P_μ is $2d$ and the degree of the G_i's is $4d-2$, we get: if $P_\mu(x,y) > 0$ then $m_{P_\mu}(x,y) \geq L_{2d}.d(x,y,Z_{P_\mu})$ and $m_{G_i}(x,y) \geq L_{4d-2}.d(x,y,Z_{G_i})$.

Remark: $\deg G_i \leq 4d-2$ because the coefficient of the term of degree $4d-1$ is $2df_d^2 + 2dg_d^2 - 2df_d^2 - 2dg_d^2 = 0$.

Putting these facts together, we obtain the following proposition:

Proposition 3.3 *Let $H(X,Y) = P_\mu(X,Y)$ or $H(X,Y) = G_i(X,Y)$, $i = 1,2$. Let $S := S(x,y,r)$. The logical relation*

$$E(H,S) \text{ is } \mathsf{true} \quad \text{means} \quad m_H(x,y) > r,$$

so S does not contain any zero and is excluded in the bisection algorithm. Otherwise

$$E(H,S) \text{ is } \mathsf{false} \quad \text{means} \quad m_H(x,y) \le r,$$

and S may contain zeros. In this case S will be divided into four squares each of them with a radius $r/2$.

3.2. Inclusion test

Our inclusion test is based on Smale's α-theory. The test is true if :

1- $\alpha(G;x,y) < \dfrac{13-3\sqrt{17}}{4}$,
and

2- $S(x,y,r) \subset B\big(x,y,\sigma(x,y)\big)$, in other words $r\sqrt{2} < \sigma(x,y)$, see Theorem 2.3.

3.3. Proof of theorem 1.3

Consider a retained square $S := S(x_l, y_l, \frac{r_0}{2^k})$ at step k. We have $\mu := \mu_k$ and $P_\mu(x_l,y_l) \ge 0$.

If $E(P_\mu, S)$ is false and $E(G_i,S)$ is false, it means, by Proposition 3.3, that $P_\mu(x_l,y_l) = 0$ or $m_{P_\mu}(x_l,y_l) \le \dfrac{r_0}{2^k}$ and $m_{G_i}(x_l,y_l) \le \dfrac{r_0}{2^k}$.

From Lemma 3.2, we deduce that

$$\text{if } P_\mu(x_l,y_l) > 0 \text{ then} \qquad L_{2d}.d(x_l,y_l,Z_{P_\mu}) \le m(x_l,y_l) \le \frac{r_0}{2^k}.$$

Hence if $P_\mu(x_l,y_l) > 0$, then for each $(x,y) \in S$ we have

$$d(x,y,Z_{P_\mu}) \le ||(x,y)-(x_l,y_l)|| + d(x_l,y_l,Z_{P_\mu}) \le \sigma_k\, r_k$$

where $\sigma_k = (1 + 1/L_{2d})$ and $r_k := \frac{r_0}{2^k}$. In the same way,

$$\begin{aligned} d(x,y,Z_{G_i}) &\le ||(x,y)-(x_l,y_l)|| + d(x_l,y_l,Z_{G_i}) \\ &\le \omega_k\, r_k, \quad i = 1,2. \end{aligned}$$

where $\omega_k = (1 + 1/L_{4d-2})$. For $H(x,y) \in \mathbf{R}[x,y]$ and $\epsilon > 0$, let

$$V_\epsilon(H) = \{(x,y) \in \mathbf{R}^2, \text{st. } \exists (u,v) \in \mathbf{C}^2, H(u,v) = 0, \quad d((x,y),(u,v)) < \epsilon\}.$$

$V_\epsilon(H)$ is called the tubular neighborhood of the solution set of $H(x,y) = 0$ at distance ε.

By the previous inequalities, if $P_\mu(x_l,y_l) > 0$ then we have

$$S \subset U_k := V_{\sigma_k}(P_\mu) \cap V_{\omega_k}(G_1) \cap V_{\omega_k}(G_2).$$

Notice that $A_F \subset \cap_{k\ge 0} U_k$. Let $\kappa_k\, r_k$ be half the maximal diameter of a connected component U_k. The area of U_k is bounded by $\pi\, \nu_k\, \kappa_k^2\, r_k^2$ where ν_k is the number of connected components of U_k. For k big enough, this number of connected components is the number of real roots of $G_1(x,y) = G_2(x,y) = 0$ with $P_\mu(x,y) > 0$. We denote by $\mathcal{N} < \infty$ the maximum of all the ν_k.

When k tends to infinity, the connected components of U_k tend to the real roots of $G_1(x,y) = G_2(x,y) = 0$ with $P_\mu(x,y) > 0$ and κ_k tends to $\sqrt{2\,(1 + cos(\alpha))}$ where α is the angle between the tangents of the curves $G_1(x,y) = 0$ and $G_2(x,y) = 0$, at the root. Let K bound all the possible κ_k.

Let J be the first J such that $K\,r_J \leq \dfrac{\delta_0}{\gamma(G, A_F)}$, then for $k \geq J$, all the points of the retained squares are approximated zeros of the set A_F, by Proposition 2.4 and Theorem 2.3. An upper bound for J is given in the theorem.

3.4. Proof of theorem 1.4

To compute an upper bound for the number of exclusion tests, we notice that at step k, U_k contains the union of the kept squares (whose exclusion test is false), their number is denoted by q_k. Since the area of this union is $q_k r_k^2$ and must be less or equal to the area of U_k, we get

$$q_k \leq \pi\,\nu_k\,\delta_k^2 \leq \pi\,\mathcal{N}\,K^2.$$

Now, let p_k be the number of excluded squares at step k. As we know the relation $4q_{k-1} = p_k + q_k$ with $p_0 = 0$, $q_0 = 1$ holds, the number of exclusion tests until step j is bounded by

$$\sum_{k=0}^{j} p_k + q_k = 1 + 4\sum_{k=0}^{j-1} q_k \leq 1 + j\,\pi\,\mathcal{N}\,K^2.$$

The last part of Theorem 1.4 comes from Proposition 2.5 which gives: for $(x,y) \in A_F$, $\log\gamma(G;x,y)$ belongs to $O\Big(\log\Big(d^2 r_0^{4(d-1)}\,||G||_B\,||DG^{-1}(x,y)||\Big)\Big)$ or to $O\Big(\log(d) + d\log(r_0) + \log(||G||_B) + \log(\sigma_F)\Big)$.

4. Examples

Our algorithm have been implemented in Matlab 7 (we use an Intel Xeon 3 Ghz processor) and we have perform some experiments to show its behavior. Currently, our algorithm is slower than other methods (e.g. [4, 14, 24]) due to the bound on the initial square given in Lemma 2.6. Nevertheless, our algorithm is certified. In the following we give examples and timings.

4.1. Increasing the degree

We consider the following example taken from the work of D. Rupprecht [21]. In his Ph.D. thesis, he developed a technique which allowed him to certify the degree of an approximate gcd but only if the required precision ε belongs to some intervals. There are small gaps between these intervals and the numerical computation of an approximate gcd is sensitive.

In our formulation of the nearest gcd problem, we look for the nearest pair (p,q) of degree d with a non-trivial gcd.

$$\begin{aligned}
f &= (z^2 - 1.001)\,(z^2 + 1.00000001)\,(z^3 + 2\,z^2 - 2.999999\,z + 1)\\
&= 1.000000000000000\,z^7 + 2.000000000000000\,z^6 - 3.000998990000000\,z^5\\
&\quad +0.998000020000000\,z^4 - 0.998000041009990\,z^3 - 2.003000010020000\,z^2\\
&\quad +3.002999029029990\,z - 1.001000010010000
\end{aligned}$$

$$\begin{aligned}
g &= (z^2 - 0.999)(z^2 + 1.00000003)(z^4 + 3z - 1.0000002)\\
&= 1.000000000000000z^8 + 0.001000030000000\,z^6 + 3.000000000000000\,z^5\\
&\quad -1.999000229970000\,z^4 + 0.003000090000000\,z^3 - 0.001000030200006\,z^2\\
&\quad -2.997000089910000\,z + 0.999000229770006
\end{aligned}$$

The nearest perturbed pair (p, q) with a non-trivial gcd that we compute is:

$$\begin{aligned}
p \quad = \quad & (0.000000001537219 + 0.000000006148875\, i)\, z^8 \\
& +(0.999999993851125 + 0.000000001537219\, i)\, z^7 \\
& +(1.999999998462781 - 0.000000006148875\, i)\, z^6 \\
& +(-3.000998983851125 - 0.000000001537219\, i)\, z^5 \\
& +(0.998000021537219 + 0.000000006148874\, i)\, z^4 \\
& +(-0.998000047158864 + 0.000000001537219\, i)\, z^3 \\
& +(-2.003000011557218 - 0.000000006148874\, i\, z^2 \\
& +(3.002999035178864 - 0.000000001537219\, i)\, z \\
& +(-1.001000008472781 + 0.000000006148874\, i) \\
q \quad = \quad & (1.000000000000000 + 0.000000008719619\, i)\, z^8 \\
& +(-0.000000008719619 - 0.000000000000000\, i)\, z^7 \\
& +(0.001000030000000 - 0.000000008719619\, i)\, z^6 \\
& +(3.000000008719619 + 0.000000000000000\, i)\, z^5 \\
& +(-1.999000229970001 + 0.000000008719619\, i)\, z^4 \\
& +(0.003000081280381 - 0.000000000000000\, i)\, z^3 \\
& +(-0.001000030200006 - 0.000000008719619\, i)\, z^2 \\
& +(-2.997000081190381 + 0.000000000000000\, i)\, z \\
& +(0.999000229770005 + 0.000000008719619\, i)
\end{aligned}$$

The order of the perturbation is 10^{-8} and the degree of the corresponding gcd is 1.

Note that in this example, even if f is of degree 7, we allow a perturbation of degree 8 (which is the degree of g).

4.2. Solution at infinity

Here, we show that the global minimum can be reached at infinity. We consider:

$$\begin{aligned}
f &= z^2 - 4z + 3, \\
g &= z^2 + 4z + 3.
\end{aligned}$$

In this situation, we have:

$$F(X, Y) = \frac{N(X, Y)}{D(X, Y)} = 2\,\frac{9 + 22X^2 + 10Y^2 + (X^2 + Y^2)^2}{1 + X^2 + Y^2 + (X^2 + Y^2)^2}.$$

Figure 1 shows the graph of F. Notice that there are no global minima of F in $\mathbf{C}$. Indeed, we have

$$\lim_{\|(x,y)\| \to \infty} F(x, y) = \mathcal{F}(\infty) = 2,$$

$$N(X, Y) - 2D(X, Y) = 16 + 42X^2 + 18Y^2 > 0.$$

Then the global minimum is reached at infinity.
In this situation the initial square given by Lemma 2.6 is $[-130, 130]^2$, and our algorithm excludes this square after 10 iterations. That is to say, the size of the smallest excluded square is $130/2^{10}$. The cpu-time in this example is 0.06 seconds.

In this situation, we have $\mathcal{F}(\infty) = 2$. Furthermore, with $z_0 = \infty$ and the formula recalled in Subsection 1.1, we obtain $p = -4z+3$ and $q = 4z+3$. Notice that $\deg(p) < 2$ and $\deg(q) < 2$.

Remark: We compute $\mathcal{F}(\infty)$ in the following way: $\mathcal{F}(\infty) = \mathcal{F}^r(0)$, where $\mathcal{F}^r(z) = z^{\deg(\mathcal{F})}\mathcal{F}(1/z)$.

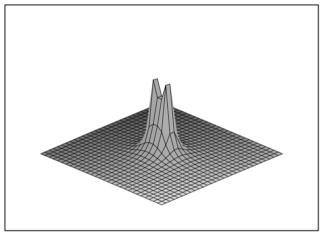

Figure 1: F has no global minima in $\mathbf{C}$.

4.3. Example 3

Here, we consider:

$$\begin{aligned} f &= 9z^3 - 18z^2 + 3z + 1, \\ g &= -37z^3 + 63z^2 - 30z + 3. \end{aligned}$$

We illustrate the algorithm step by step. The smallest perturbation that we obtain is: $p(z) = 9.059z^3 - 17.875z^2 + 3.26z + 1.544$,
$q(z) = -36.916z^3 + 63.174z^2 - 29.636z + 3.758$, and the common root is $z = 0.479$.
Step 1. We begin in Figure 2 (a) with a big square and we divide it in four squares.

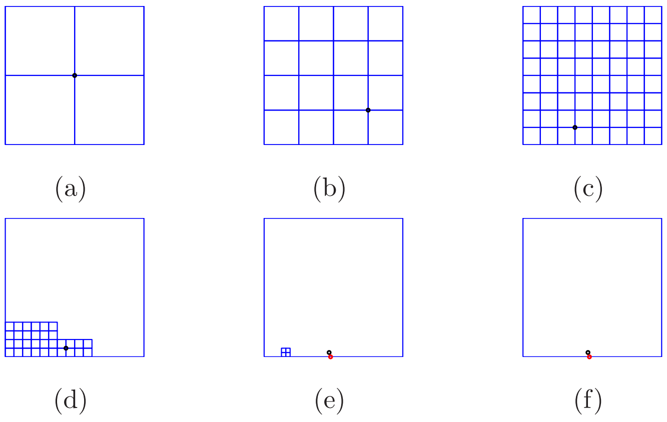

Figure 2: Subdivision steps

Step 2. As we cannot exclude any squares and all the inclusion tests are false we divide each square in four squares (Figure 2 (b)).

During the algorithm we compute F for each center of each square. The point in the figure corresponds to the center where F is minimum. Furthermore, the value of F at this point is equal to the real number μ defined in our algorithm.

Step 3. In Figure 2 (c), we cannot exclude any squares and the inclusion tests are false.

Step 4. At this step, see Figure 2 (d), we can exclude a lot of squares.

Step 5. In Figure 2 (e), we see that we can exclude a lot of squares and one inclusion test is true. So with the Newton's method applied to G we can compute the global minima. That is to say we apply Newton's method with the black point as a starting point and we get the red point (on the x-axis) as the global minimum.

Step 6. In this last step, we can exclude the last squares and we find the global minimum, see Figure 2 (f).

4.4. Random examples

We have carried out the following tests:

- f and g are polynomials of degree d with random coefficients. Our results are given in Figure 3, and *cpu* denotes the cpu-time needed in seconds to performs our algorithm.
- $f(z) = (z-1)f_1(z) + 10^{-3}\epsilon_1(z)$, $g(z) = (z-1)g_1(z) + 10^{-3}\epsilon_2(z)$, where f_1 and f_2 are random polynomials with degree $d-1$, ϵ_1 and ϵ_2 are random polynomials with degree d. In these examples our algorithm gives an approximate gcd near $z-1$. Our results are given in Figure 4, and *cpu* denotes the cpu-time needed in seconds to performs our algorithm.

Here random means that we use the Gaussian distribution with mean 0 and variance 1.

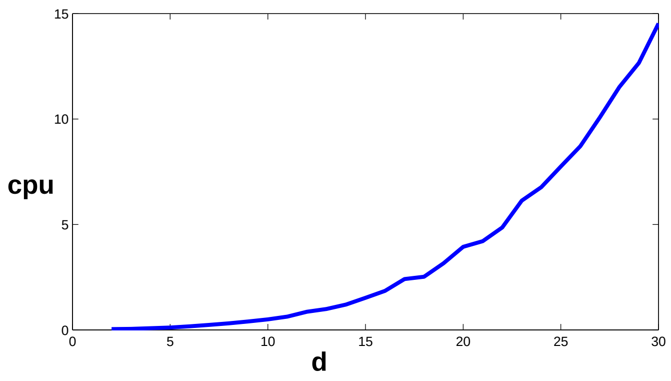

Figure 3: Random examples

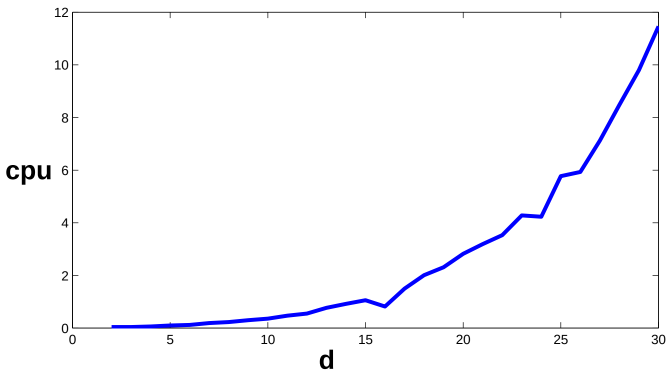

Figure 4: $f = (z-1).f_1 + 10^{-3}\epsilon_1$, $g = (z-1).g_1 + 10^{-3}\epsilon_2$

5. Acknowledgments

The authors thank Prof. Lihong Zhi for her comments during the presentation at SNC'09 about cases with roots at infinity (Example 2). The authors also thank the referees for their comments, which helped to increase the quality of the paper. B. Mourrain is partially supported by Marie-Curie Initial Training Network SAGA, [FP7/2007-2013], grant [PITN-GA-2008-214584].

[1] Dennis S. Arnon and Scott McCallum. A polynomial-time algorithm for the topological type of a real algebraic curve. *J. Symbolic Comput.*, 5(1-2):213–236, 1988.

[2] Bernhard Beckermann and George Labahn. When are two numerical polynomials relatively prime? *J. Symbolic Comput.*, 26(6):677–689, 1998. Symbolic numeric algebra for polynomials.

[3] D. Bini and V. Y. Pan. *Polynomial and matrix computations, Vol 1 : Fundamental Algorithms*. Birkhäuser, Boston, 1994.

[4] Dario A. Bini and Paola Boito. Structured matrix-based methods for polynomial ϵ-gcd: analysis and comparisons. In *ISSAC 2007*, pages 9–16. ACM, New York, 2007.

[5] Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale. Complexity and real computation: a manifesto. *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, 6(1):3–26, 1996.

[6] W. S. Brown. On Euclid's algorithm and the computation of polynomial greatest common divisors. *J. Assoc. Comput. Mach.*, 18:478–504, 1971.

[7] G. Chèze, J.-C. Yakoubsohn, A. Galligo, and B. Mourrain. Computing Nearest Gcd with Certification. In *Symbolic-Numeric Computation (SNC'09)*, pages 29–34, Kyoto Japon, 08 2009. ACM New York, NY, USA.

[8] R.M. Corless, P.M. Gianni, B.M. Trager, and S.M. Watt. The singular value decomposition for polynomial systems. In A.H.M Levelt, editor, *Proc. Internat. Symp. Symbolic Algebraic Comput. (ISSAC '95)*, pages 195–207, 1995.

[9] J. P. Dedieu, X. Gourdon, and J. C. Yakoubsohn. Computing the distance from a point to an algebraic hypersurface. In *The mathematics of numerical analysis (Park City, UT, 1995)*, volume 32 of *Lectures in Appl. Math.*, pages 285–293. Amer. Math. Soc., Providence, RI, 1996.

[10] I.Z. Emiris, A. Galligo, and H. Lombardi. *Numerical Univariate Polynomial GCD*, volume 32 of *Lectures in Applied Math.*, pages 323–343. AMS, 1996.

[11] I.Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. *J. Pure & Applied Algebra, Special Issue on Algorithms for Algebra*, 117 & 118:229–251, May 1997.

[12] M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn. On location and approximation of clusters of zeros of analytic functions. *Found. Comput. Math.*, 5(3):257–311, 2005.

[13] V. Hribernig and H. J. Stetter. Detection and validation of clusters of polynomial zeros. *J. Symbolic Comput.*, 24(6):667–681, 1997.

[14] Erich Kaltofen, Zhengfeng Yang, and Lihong Zhi. Structured low rank approximation of a Sylvester matrix. In *Symbolic-numeric computation*, Trends Math., pages 69–83. Birkhäuser, Basel, 2007.

[15] N. K. Karmarkar and Y. N. Lakshman. On approximate GCDs of univariate polynomials. *J. Symbolic Comput.*, 26(6):653–666, 1998. Symbolic numeric algebra for polynomials.

[16] Dinesh Manocha and James Demmel. Algorithms for intersecting parametric and algebraic curves: I simple intersections. *ACM Trans. Graph.*, 13(1):73–100, 1994.

[17] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[18] Jiawang Nie, James Demmel, and Ming Gu. Global minimization of rational functions and the nearest GCDs. *J. Global Optim.*, 40(4):697–718, 2008.

[19] Matu-Tarow Noda and Tateaki Sasaki. Approximate GCD and its application to ill-conditioned algebraic equations. In *Proceedings of the International Symposium on Computational Mathematics (Matsuyama, 1990)*, volume 38, pages 335–351, 1991.

[20] Victor Y. Pan. Computation of approximate polynomial GCDs and an extension. *Inform. and Comput.*, 167(2):71–85, 2001.

[21] D. Rupprecht. *Eléments de Géométrie Algébrique Approchée: étude du PGCD et de la factorisation.* PhD thesis, Université de Nice - Sophia Antipolis, Jan. 2000.

[22] Arnold Schönhage. Quasi-gcd computations. *J. Complexity*, 1(1):118–137, 1985.

[23] Michael Shub and Steve Smale. Complexity of Bézout's theorem. I. Geometric aspects. *J. Amer. Math. Soc.*, 6(2):459–501, 1993.

[24] Akira Terui. An Iterative Method for Calculating Approximate GCD of Univariate Polynomials. In *ISSAC 2009*, pages 351–358. ACM, New York, 2009.

[25] K. Tanabe. A geometric method in nonlinear programming. *J. Optim. Theory Appl.*, 30(2):181–210, 1980.

[26] Zhonggang Zeng The approximate GCD of inexact polynomials. Part I: a univariate algorithm. Preprint 2004.