Approximate Bivariate Factorization, a Geometric Viewpoint

Andre Galligo[*]
Universite de Nice (and INRIA)
Laboratoire de Mathematiques
Parc Valrose 06108 Nice cedex 02, France
galligo@unice.fr

Mark van Hoeij[†]
Department of Mathematics
Florida State University
Tallahassee, FL 32306-4510, USA
hoeij@math.fsu.edu

ABSTRACT

We briefly present and analyze, from a geometric viewpoint, strategies for designing algorithms to factor bivariate approximate polynomials in $\mathbb{C}[x, y]$.

Given a composite polynomial, stably square-free, satisfying a genericity hypothesis, we describe the effect of a perturbation on the roots of its discriminant with respect to one variable, and the perturbation of the corresponding monodromy action on a smooth fiber.

A novel geometric approach is presented, based on guided projection in the parameter space and continuation method above randomly chosen loops, to reconstruct from a perturbed polynomial a nearby composite polynomial and its irreducible factors. An algorithm and its ingredients are described.

Categories and Subject Descriptors

I.1.2 [**Computing methodologies**]: Symbolic and Algebraic Manipulation—*Algebraic Algorithms*

General Terms

Algorithms,Theory

Keywords

Approximate Factorization, Algebraic Geometry, Algorithms, Maple Code

1. INTRODUCTION

1.1 Approximate algebra

Over the past ten years symbolic-numeric algorithms for approximate polynomials (computation of greatest common divisors, functional decompositions, find zeros of multivariate systems, test primality, factorization) have been studied by many authors (see e.g. the proceedings of the SNC conferences and the references inside). The main common feature of the produced algorithms is to propose a strategy to find a nearby object (some authors claim and even prove that they found the nearest one in some sense) with required properties. This can be often re-interpreted as a reverse engineering task: recognize a perturbed situation by a "small" distortion of a representation. However the size of this small distortion is usually not easy to formalize and many authors present the efficiency of their algorithms by their effectiveness on some benchmarks.

Ideally the benchmarks should consist of polynomials coming from computations related to problems in engineering or other sciences; but for now there are too few such concrete applications. So, examples in the benchmarks are constructed algebraically first by an exact computation on exact random data and then corrupted by a random noise; this is useful to detect weaknesses and to check robustness of the proposed methods. A better control of these constructed examples would profit from geometric insights on the possible singular situations.

1.2 Approximate factorization

An important problem of this family which meets a renewed interest is the factoring of approximate multivariate polynomials and the bivariate case captures its essential issues. There exist several methods, algorithms and implementations for factoring multivariate polynomials in an exact setting. See e.g. [2] or [16] or [4] and their bibliography. A corresponding approximate problem (for two factors) can be stated as follows (see e.g. [24]):

Given $f \in \mathbb{C}[x, y]$ of total degree n, and $\varepsilon \in \mathbb{R}$, find a pair g and $h \in \mathbb{C}[x, y]$ such that $g \times h = f + \Delta f$, where $\deg g, \deg h > 0$, $||\Delta f|| < \varepsilon$, degree$(\Delta f) \leq n$ and $||\cdot||$ denotes a polynomial norm.

To simplify the discussion, let us assume, in the sequel, that f is "approximately square-free", that is, all polynomials in a neighborhood of f are square-free. This can be determined in practice by taking a random complex floating-point value for x and using a fast algorithm to certify that the approximate GCD of $f(x, y)$ and $f_y(x, y)$ is 1. If this is not the case, then an algorithm to remove the multivariate approximate GCD such as described e.g. in [7, 29, 37] could be used.

This approximate factorization problem has been already addressed by several authors as indicated below. More pre-

[*] Partially supported by the French Agence National de la Recherche, contract GECKO

[†] Supported by NSF grant 0511544. Both authors acknowledge support of the American Institute of Mathematics; this research was initiated at AIM workshop May 15–19, 2006.

SNC'07, July 25–27, 2007, London, Ontario, Canada.
Copyright 2007 ACM 978-1-59593-744-5/07/0007 ...$5.00.

cisely, the methods find factors of an approximate polynomial if the given polynomial is “sufficiently close” to being factorable.

The previous (pseudo-) definition of approximate bivariate factorization admits obvious variants such as fixing a support for the exponents of the monomials allowed for Δf or for $f + \Delta f$, or fixing the convex hull of that support. An important case is to consider polynomials of fixed bidegree in (x, y).

In the next subsection the approximate factorization problem will be translated into the task of projecting a point on a subvariety, up to small perturbations, and this clearly does not have a unique solution in general.

1.3 Projection in a parameter space

Let $\mathcal{E}$ be the allowed support for the exponents of the monomials of f and of $f+\Delta f$, as a normalization we impose that one coefficient is equal to 1; let $N = \sharp(\mathcal{E}) - 1$. So f can be considered as a point in the parameter space $\mathbb{C}^N$. Then, there is a closed algebraic subvariety $\mathcal{V}$ of $\mathbb{C}^N$ which contains all the composite polynomials. Equations of $\mathcal{V}$ could be computed via generalized resultants, they are sometimes called Noether forms (Emmy Noether introduced and studied them). The subvariety $\mathcal{V}$ has various components and singularities. Endowing $\mathbb{C}^N$ with any metric, there is no chance that the projection of a point f on $\mathcal{V}$ could be unique in general. The subvariety $\mathcal{V}$ can be stratified by the number of components of the corresponding polynomials and each such stratum is also stratified by the degrees (or the partial degrees) of these factors. The associated combinatorics is rather involved, therefore $\mathcal{V}$ is a complicated object.

Of course the algebraic question carries the same hardness shown by its geometric interpretation. So we can expect uniqueness of the projection (hence a well posed approximate factorization problem) only if f is near a smooth part of one of the previous strata (defined by some discrete invariants) and if we can guide the projection to reach that strata. This is a difficult task, much more complicated that the corresponding problem for univariate GCD that we now briefly recall.

1.4 Comparison with univariate GCD

The degree of the univariate GCD of two polynomials f and g in $\mathbb{C}[x]$ is given by the co-rank of the Sylvester matrix, but the GCD can be computed by the sequence of subresultants extracted from the Sylvester matrix.

The seminal paper [7] introduced the use of SVD of the Sylvester matrix to estimated the degree of the univariate GCD, this allowed a first approximate GCD algorithm. But there was an important weakness in that paper, the authors confused the distance to the set of all matrices of given rank k with the distance to the strict subset of Sylvester matrices of rank k. This was revealed and corrected in a series of papers see e.g. [11, 12] which produced (under mild conditions) a certified approximate GCD algorithm relying on SVD of subresultants; this algorithm was very efficiently implemented by D. Rupprecht. Later the approach was continued by Zeng and his coworkers.

Let's compare these developments with those in approximate factorization. Recently, important progress in the understanding of the algebraic conditions fulfilled by the coefficient of a composite multivariate polynomial (Noether forms) was achieved: They can be interpreted as minors of the Ruppert-Gao matrix (see below) which will play a role similar to the Sylvester matrix as its co-rank indicates the number of factors. However, we do not have so far a tool similar to the subresultants which could give rise to a certified approximate factorization algorithm. So, for the moment it is worthwhile to accumulate knowledge of various sorts about approximate bivariate factorization: algebraic approaches successful in many cases, examples of difficult cases, geometric approaches, mixed strategies, examples related to concrete applications. Preliminary experiments reported in [14] show the feasibility of our approach on some examples.

2. PREVIOUS WORKS

2.1 Some references

For a history of early algorithms on multivariate polynomial factorization see [22] and [23]. The paper [1] is the first algorithmic paper using monodromy group action as developed below in section 2. The paper [15] considers point combinations, an exponential search, while the paper [20] uses an optimization method exponential in the degree of the factor recovered. The paper [21], is also of complexity exponential in the degree of the input. The papers [33, 31, 32, 30] discuss another interesting algorithm based on zero-sum identities of power-series solutions of $f(x, y) = 0$. This algorithm is numerically stable and of polynomial complexity. The paper [5] adopted a backward error analysis point of view. Similarly for the papers [25], [17] which relied on the use of SVD on Ruppert-Gao matrix. The paper [36] takes the general method of [34, 35] by the same authors and specializes it to the case of a single polynomial. It makes use of an analysis of how root clusters perturb under differentiation.

2.2 Factorization and topology

A bivariate polynomial equation $f(x, y) = 0$ defines a curve X in $\mathbb{C}^2$. We point out two main topological geometric properties of X closely related to factorization of f. We assumed that f is squarefree so X is reduced.

1. The number s of factors of f, i.e. the number of irreducible components of X, is equal to the dimension of the homology group $H_1(\mathbb{C}^2 - X)$.

   The intuition behind this claim is that the loops in $\mathbb{C}^2$ around each component X_i of X generate the homotopy group of the loops around X. See e.g. [10] for details.

   By Poincare duality the dimension of the co-homology group $H^1(\mathbb{C}^2 - X)$ is also s. The definition of that group in terms of differential forms provides Rupert's condition, then its expression in monomial bases gives rise to Ruppert-Gao matrix.

2. The closure of each connected component of $X - Sing(X)$ corresponds to an algebraic curve whose equation is an irreducible factor of f; here $Sing()$ denotes the singular locus which consists of a finite number of points of X; solutions of the system of 3 equations and 2 unknowns: $f = f_x = f_y = 0$.

The second condition can be analyzed further using a projection on a line. To simplify the discussion, assume that no irreducible component of $\mathcal{C}$ is a line, (this case can be treated separately), let d be the degree of f in y and call π the projection of X on the x-axis. Then, except for a finite set of values A, π is d to 1. More precisely $X - \pi^{-1}(A)$ is a d-covering of the line minus A, moreover it is the union of s connected such coverings $X_i - \pi^{-1}(A)$.

For x_0 not in A, the fiber $E = \pi^{-1}(x_0)$ consists of d distinct points, partitioned in s subsets E_i, E_i lying on $X_i - \pi^{-1}(A)$ for $1 \leq i \leq s$. Note that this partition of E characterizes the aimed factorization of f, as it defines it modulo $(x - x_0)$, and the factorization can be recovered via x-adic Hensel liftings.

These two properties were the basements of two kinds of factorization algorithms in the exact setting and are crucial for the understanding of the geometric situation in the approximate setting.

2.3 Ruppert-Gao matrix and SVD

In a series of important papers [17, 26] a group of authors extended Gao's factorization algorithm into an effective method for finding an approximate factorization of a multivariate polynomial.

The general idea is to closely follow Gao's strategy replacing the computation of the co-rank of Ruppert-Gao matrix by the computation of the singular values list of that matrix, then choosing the largest gap to infer both a level of approximation ϵ and a candidate integer s for the number of factors. With that choice, the generators of an approximate kernel are computed and candidate approximate factors are obtained via approximate multivariate GCD computations. Finally a postprocessing is used to diminish the distance between f and its proposed factorable deformation.

The authors have implemented their algorithm in Maple and report very good results on a large benchmark. So the approach is successful, nevertheless it leaves room for improvements both in efficiency and in analysis of the process.

2.4 Continuation or homotopy methods

A continuation method was proposed in [5], it consists essentially in following a path in X accumulating sufficiently many points on the same connected component say X_1. An approximate interpolation provides a candidate factor f_1 of f, then an approximate division is performed.

Other authors tried to proceed directly to the (parallel) interpolation of all s factors, with the expectation that the process will be more stable, but this requires to estimate first the correct partition of a fiber E.

One or several infinitesimal trace(s) (or zero sums) methods are used for this purpose, [30, 31, 32, 33]. A variant is to consider an interesting analog of traces obtain by discretizations on a grid in the x-axis [6] then lifted by an homotopy on X.

In the paper [34] was made the following important experimental observation (in the case of exact inputs, approximations with a great precision and with a slightly different monodromy action than the one considered here) which inspired our algorithmic probabilist strategy: The partition of the fiber E can be recovered from a small number of permutations of E corresponding to the monodromy action of random loops. Whereas, in theory as in [1], one needs to consider a set of representative of generators of the fundamental group which consists of a huge number of transpositions or other permutations.

3. A NOVEL APPROACH

3.1 Another distance

In this paper we will also consider a geometric variation of the cited problem: control the size of the distance between f and $f+\Delta f$ not only via their coefficients but also via discrete geometric data, namely discriminant and critical loci and monodromy group attached to the polynomial. This point of view can be related to the computation of the approximate GCD of two univariate polynomials f and g by pairing nearby (controlled by a distance) roots of f and g which are then merged. Here the situation is more involved as the zero-sets are curves. We chose to rely on a finite set of characteristic points: the critical locus of the projection of the associated curve on the x-axis. Indeed this characterizes an algebraic curve for a given degree $n > 2$.

As above, denote by X the curve $f^{-1}(0) \in \mathbb{C}^2$, by π its projection on the x-axis and choose a generic (i.e. random) fiber $E = \pi^{-1}(x_0)$ in X which has n points. To simplify the notations, we let $x_0 = 0$. We denote by $\Delta \in \mathbb{C}$ the discriminant locus of π. The action of the fundamental group $\pi_1(\mathbb{C} - \Delta)$ on E, defines the monodromy group G, which can be explicitly calculated. The package algcurves in the computer algebra system Maple contains a program computing the monodromy.

When f is irreducible, the orbit of G is the whole fiber E, while when f is composite: $f = f_1 \cdots f_s$, the orbits of G provide the s-partition of E by the subsets formed by the roots of the factors f_i. This is the key combinatorial information which allows us to recover the factorization of f by a continuation method. See e.g. [8, 2]. In the exact setting, an early polynomial time algorithm for computing the absolute factorization of a bivariate polynomial using the monodromy was presented by [1], monodromy also plays an important role in the factorization algorithms presented in [15, 13, 29, 34, 35, 2, 3].

Once the partition is determined, we proceed by an interpolation at the intersection points or on a grid to compute the approximate factors, using the points identified in the earlier steps. Alternatively we can adapt from [5] and [18] the needed routines in Maple for computing continuations and interpolation.

3.2 A generic situation

Here, we restrict our study to the following situation (which is the one encountered in many application and most benchmark examples): The polynomial F to be factored is a small perturbation of a product $f = f_1 \cdots f_s$ such that the curves $X_i = f_i^{-1}(0)$ are all smooth and intersect transversally in double points (nodes). The general idea is that the input equation F has been corrupted by some noise creating a relative error on the data of order α, estimated by the user, that we can use in our program to estimate some bounds threshold; but we can do computations with rational numbers or bigfloats with a higher precision. As we can start with a "generic" change of coordinates, we can suppose that f and its perturbation F are monic in y of degree n and total degree n and that the projections of the critical points on the x-axis are all distinct, therefore we expect that the only singularities of the discriminant are double roots corresponding to the projections of intersections of two components. The proposed factoring method is based on three main ideas.

First, as the X_i are smooth and cut transversally, the discriminant points of f are either (simple) branching points (of one X_i) or double points (corresponding to projections of intersection points of two components X_i and X_j). By a (small) perturbation the double points give rise to a pair of nearby branching points which generate the same transposition of the fiber E at a base point, i.e. there is a path looping around only these two branching points which induces the identity on E.

Second, following the previous analysis, as long as we don't pass between any such cluster of branchpoints, that we call "forbidden zones", the monodromy induced by F is similar to the monodromy induced by f.

We observed that a main task is to locate the clusters formed by the deformation of the nodes. This led us to the third idea that an important preprocessing for the approximate factorization can be achieved if we can locate "centers" for these clusters and "project" F on the subvariety (in a parameter space) of the polynomials with the same terms as F having double points at these points. Our key idea to locate these centers is given in subsection 6.2.

3.3 A geometric strategy

We aim to adapt in the approximate setting, SVW's [34] method which amounts (for us) to determine the partition of the fiber E by following the roots in y above a small number of random loops in the x-axis.

So, instead of looking for all the centers of the clusters mentioned in the previous subsection, we concentrate on the ones in a neighborhood of these randomly chosen loops. Then we drag them out of that neighborhood by an adapted deformation of f, in order to obtain cleared paths, which resembles more the geometric situation encountered in the exact setting.

3.4 An illustrative example

We will use the following illustrative example: f_1 and f_2 are two random dense bivariate polynomial of degree 3 with 2-digit coefficients, f is their product divided by 100, so that the coefficient are about 50 and F is a perturbation of f with a relative error of 10^{-2}.

```
f1 := 92*y^3+44*x*y^2+40*x^2*y-67*x^3
      +8*y^2+66*x*y+68*x^2-95*y-62*x-18:
f2 := 68*y^3+39*x*y^2+20*x^2*y+45*x^3
      -65*y^2-67*x*y+93*x^2+43*y+8*x+6:
f:=f1*f2/100:  F:=evalf(f,3):
```

The relative perturbation on the coefficients of the discriminant of f is about $5 * 10^{-2}$. The relative perturbation on its roots is about 10^{-1} for the double points and about 10^{-2} for the simple points. This example will continue in sections 5 and 6.

4. MONODROMY GROUP

In this section, we describe algorithmically our main tool, the monodromy group: its representation and its calculation is implemented in the package algcurves of Maple that we will use. More details can be found in [8].

Let $f(x, y)$ be a polynomial of degree n in y, that we suppose monic in y of degree n, this hypothesis simplifies the presentation and is always satisfied after a generic (random) change of coordinates. Let $X = f^{-1}(0)$ be the defined curve in $\mathbb{C}^2$ and $\pi : X \to \mathbb{C}$ the projection on the x-axis. The discriminant locus Δ of f is the zero-set of $D := \mathrm{Res}_y(f, f'_y)$, it contains the simple branching points which are the projections of the points with a vertical tangent and the projections of the singularities of X (which are the solutions of $f = f'_y = f'_x = 0$).

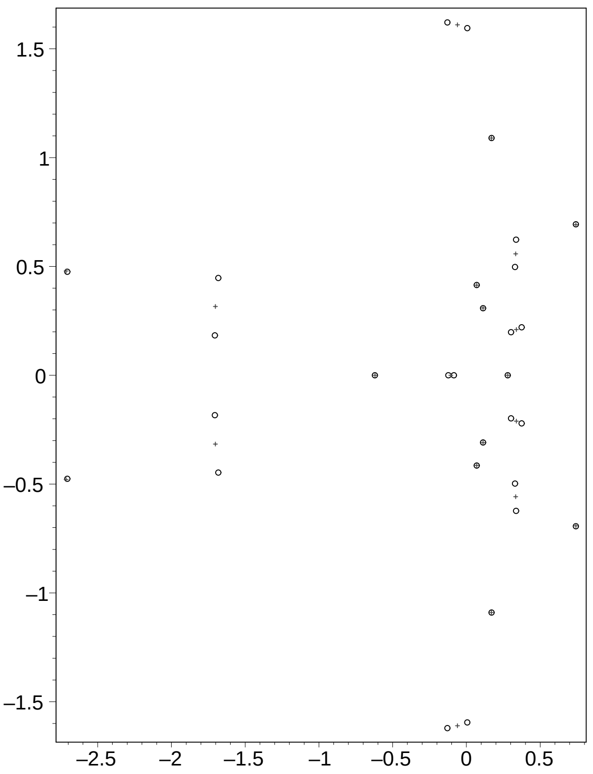

Figure 1: Discriminant of F and double points of f (indicated with $\circ$ resp. $+$ symbols)

To define the monodromy, first select a base point $x = a$ in the complex x-plane minus the discriminant locus. Let E be the fiber of p above a (i.e., the n distinct y-values for which $f(a, y) = 0$). These y-values are now assigned an order, $(y_1, y_2, \ldots, y_n)$. This ordering of the n y-values labels the sheets of the covering $X - \pi^{-1}(\Delta)$ of $\mathbb{C} - \Delta$.

For each point $b \in \Delta$, one chooses a path γ_b in the complex x-plane which starts and ends at $x = a$, encircles only $x = b$ counterclockwise and avoids all points of Δ. The n-tuple $(y_1, y_2, \ldots, y_n)$ is then analytically continued around this path γ_b. When one returns to $x = a$, a new n-tuple is found, which has the same entries as $(y_1, y_2, \ldots, y_n)$, but ordered differently: $(y_{\sigma_b(1)}, y_{\sigma_b(2)}, \ldots, y_{\sigma_b(n)})$, where σ_b is a permutation acting on the set of labels $\{1, 2, \ldots, n\}$. We will say that the permutation σ_b is attached to the path γ_b. Note that for different choices of γ_b, we obtain different permutations.

Here are some typical situations. If $x = b$ is a simple branching point, then σ_b is a transposition. If $x = b$ is the projection of a simple double point (a node), then σ_b is the identity. If $x = b$ is the projection of a cusp singularity like the one with local equation $x^2 - y^3 = 0$, then σ_b is the cyclic permutation of order 3.

For the effective calculation of the monodromy some labeling and ordering should be done. The Maple implementation (algcurves) that we will use (see [8]) and complete with new procedures, made the following choices.

1. **Base point and circles:** With every discriminant point b_i, a radius $r(b_i)$ is associated: (ρ denotes the distance)

   $$r(b_i) = \frac{2}{5}\rho(b_i, \{b_1, b_2, \ldots, b_m\} - \{b_i\}),$$

   Let $C(b_i, r(b_i))$ denote the circle with center b_i and radius $r(b_i)$. Then the circles $C(b_i, r(b_i))$ do not intersect each other.

   Now a base point a is chosen, such that the real part of a is smaller than the real parts of any of the b_i. By this choice, the arguments of $b_i - a$ are between $-\pi/2$ and $\pi/2$.

2. **Labeling of the sheets:** At the base point $x = a$ there are n distinct finite y-values, they form the fiber E of p. These are determined numerically as the solutions of $f(a, y) = 0$. Let these n y-values be assigned an order $(y_1, y_2, \ldots, y_n)$, this labels the corresponding sheet of the covering $y(x)$ which contains y_i.

3. **Ordering of the discriminant points:** We order these points according to their argument with respect to the base point: if $\arg(b_i - a) < \arg(b_j - a)$, then b_i precedes b_j in the ordering, where $\arg(\cdot)$ denotes the argument function. If $\arg(b_i - a) = \arg(b_j - a)$, then b_i precedes b_j if $|b_i - a| < |b_j - a|$.

4. **Choice of the paths:** The simplest path $L(b_i)$ around b_i consists of one line segment from a to $b_i - r(b_i)$. This is followed by $C(b_i, r(b_i))$, starting at $b_i - r(b_i)$. Successively, a line segment is followed from $b_i - r(b_i)$, back to a. The algorithm chooses a path that is composed of line segments and semi-circles, and that is equivalent to the simplest path.

5. **Analytic continuation:** The path γ_b is discretized in small segments $x_i x_{i+1}$ so that when a path is followed in the complex x-plane from x_1 to x_2, the n entries of $f^{-1}(x_1)$ i.e. the roots of $f(x_1, y) = 0$, follow paths on the Riemann surface to the roots of $f(x_2, y) = 0$, by a numerical integration of a vector field defined by derivatives of f (see [8] or [5] for more details) . This gives rise to an n-tuple, whose ordering is induced by the ordering of $f^{-1}(x_1)$. The accepted size of $|x_{i+1} - x_i|$ depends on the separation of the entries of $f^{-1}(x_i)$.

6. **Monodromy group:** Consider a closed path starting from $x = a$ and returning there after encircling one branch point $x = b$. After analytic continuation of E along this path, the entries of E are recovered, but they are shuffled by the permutation σ_b.

   The collection of all σ_b generates the monodromy group, which is represented here as a subgroup of S_n, the group of permutations of $\{1, 2, \ldots, n\}$. Note that this representation depends on the choice of the labeling of the y-values at $x = a$, so it is only unique up to conjugation.

   The point $x = \infty$ might also be a branch point. The corresponding permutation σ_∞ does not need to be computed by analytic continuation, since it can be determined from the other σ_b.

7. **Example:** We apply the procedure monodromy to the perturbation F of f in our illustrative example.

   ```
   with(algcurves):
   m := monodromy(convert(F,rational),x,y);
   ```

   The output is a list with three entries. The first entry m[1] is the basepoint $x = a$ that the algorithm chose. The second entry m[2] is the fiber above a (i.e. our list E), it consists of n complex numbers. The third entry m[3] is a list, each entry contains a branchpoint b_i with its permutation σ_{b_i} (given as a product of disjoint cycles).

5. EFFECT OF A PERMUTATION

Given a polynomial F supposed a small perturbation of a composite polynomial f, and a "generic" base point a in $\mathbb{C}$, we aim to recover the partition of the fiber of f above a. We denote by X the curve defined by f and by Y the "perturbed" curve defined by F. We analyze the situation.

5.1 Deformation of the discriminant and forbidden zones

With the hypothesis on f described in section 3.2, the discriminant Δ of f admits only a subset of simple roots s_1, and a subset of double roots s_2. By a small deformation F of f, each element of s_2 is deformed into a "cluster" of 2 roots of the discriminant Δ_p of F, while each element of s_1 deforms to a root of Δ_p. Our task is thus to separate in these two categories the elements of Δ_p as they all appear as simple roots of the discriminant of F. The ones in the first category come in pairs and form a set of clusters. So we will look for nearby points of Δ_p.

This is done as follows. The user specifies a level of distortion of the coefficients of f, that we use to estimate probabilistically a level of deformation for each type of roots, as follows. Let the level of relative approximation for the coefficients of f be α. Then the expectation for the level of relative error for D, β, is about $2n\alpha$. The expectation for the level for the simple roots is also about β, while the one for the double roots is about $\sqrt{\beta}$. In our illustrative example $\alpha = 5 * 10^{-2}$, so we roughly expect perturbations of order 10^{-2} for the points in s_1 and perturbation of order 10^{-1} for the points in s_2.

We note that each point in s_2 is the projection of a double point of X which is also deformed in a cluster of two critical points of the projection of Y. So we can improve the previous estimation of vicinity by considering not only the distance between two points of Δ_p but also the distance between the corresponding critical points in Y. They should be in the same level of magnitude.

A next observation is the following "conservation law". The monodromy (on any generic fiber of f) defined by a small circle looping only around a double point of Δ is the identity; because each path in the Riemann surface above the circles remains in the same layer (as for two crossing lines in $\mathbb{C}^2$). This feature is conserved by deformation: the monodromy defined by a path γ looping only around a pair of two roots of Δ_p, obtained by deformation of a double point of D, is the identity (the simplest example is a couple of crossing lines deformed in an hyperbola).

If we are able to recognize all the pairs of roots of Δ_p, obtained by deformation of an element of s_2, we can merge the two points of each such pair by creating cuts or "forbidden zones" in the complex plane. Then we require that the paths γ used to calculate the monodromy do not cross these cuts, i.e. avoid these forbidden zones. Once this is done, the remaining roots of Δ_p define (a large number of) permutations of the fiber E whose s orbits give the target partition of E.

Here are some easy enumerative data. If f admits two factors of degree $n/2$, then there are $n^2/4$ double points in the discriminant. The extreme cases are when f admits a factor of degree 1 and one of degree $n-1$, then there are $n-1$ double points; while when f admits n factors of degree 1, there are $n(n-1)/2$ double points.

5.2 A geometric property

By a (small) deformation $\tilde{f}_i$ of each factor f_i of $f = f_1 \cdots f_s$, satisfying the hypothesis of section 3.2, the number of intersection points between the irreducible components X_j and X_k, $j \neq k$, remains unchanged; hence the number of singular points of X during the deformation remains unchanged.

The opposite implication is also true but less obvious.

Proposition: Let $f = f_1 \cdots f_s$, satisfying the hypothesis of section 3.2, and $\tilde{f}$ obtained by a (small) deformation of f. If the number of singular points of f remains unchanged during the deformation, then $\tilde{f}$ is also composite with s factors.

This can be proved using the monodromy action attached to perturbed loops following the small deformation of the simple and double roots of the discriminant of f.

This result provides an interesting geometric insight on the variety of composite polynomials evoked in subsection 1.3 and will be useful for the our probabilistic algorithm.

5.3 Recognition process

In order to detect the pairs of roots (p, q) to be merged, we could perform analytic continuation over the shortest path around p and q for every pair of roots of the discriminant. However, that would be $O(n^4)$ paths. To reduce the number of paths over which we perform analytic continuation, we order the points as described in the previous section 2, and for each point, compute a path around it and the basepoint. This requires $O(n^2)$ paths. Then we use two procedures given below to detect which pairs may be merged.

Take the triangle: basepoint - p - q. Consider all the points inside that triangle, in the order in which they appear in the output M of algcurves[monodromy] (see in section 2 the subsection on the sorting of discriminant points). The only points that could be inside that triangle have ordering between p and q, and must appear on the same side of L as the basepoint, where L is the line through p and q. Finding these points is done with a simple Maple procedure, see the website [14].

Then we multiply the permutations corresponding to the points inside that triangle, to get a permutation g. Now, in order to compare the permutation of point p with that of point q, we first have to conjugate with g, and then multiply and check for identity.

Instead of deleting all pairs with distance below a certain threshold, we can also work without a threshold and simply start deleting grouped branchpoints (starting with the relatively closest pair, then the next closest, and so on) and stop as soon as the group becomes reducible (i.e. not transitive). One can now try to compute a number δ such that no δ-perturbation of F can bring a pair of distance d together. Such δ would then be a lower bound for the distance from F to the nearest reducible polynomial. In a future work, we hope to compute a lower bound in this way and compare it to alternatives like [25].

In our illustrative example, we order the 30 roots of Δ_p as said before, and take the set S of the pairs under the threshold 0.3 that satisfy the previous matching test. Then we get 9 elements:

$$S := [[15, 16, .037], [18, 19, .075], [12, 13, .075], [23, 25, .12],$$

$$[6, 8, .12], [28, 29, .13], [2, 3, .13], [21, 26, .26], [5, 10, .26]]$$

So $30 - 9 \cdot 2 = 12$ permutations remain. The generated subgroup is not transitive and provides the partition in two subsets (with 3 elements each) $\{2, 3, 4\}$ and $\{1, 5, 6\}$ of the fiber. The corresponding factors are computed by interpolation.

Remark If a pair p,q of discriminant points of F comes from a double point of f, then we assume that the path around p,q corresponds to a path around that double point (and hence has trivial monodromy). However, it is possible that this assumption fails; during the deformation, the points p,q move away from each other, starting at the double point. However, if during this process a third branchpoint r moves between p,q, then the effect of that will be that σ_p (or σ_q) will be conjugated by σ_r, after which $\sigma_p\sigma_q$ no longer needs to be trivial. A fix could be to check σ_p,σ_q *and* σ_r whenever a pair p,q is so close to another branchpoint r that this situation may have occurred (for additional robustness this could be combined with the computation of the pseudo singular points (see below) because such a point should be found near the center between p,q if p,q originated from a double point).

6. TOWARD AN EFFICIENT ALGORITHM

We now list several ingredients needed for our probabilistic algorithm.

6.1 Early detection

In this subsection, we sketch a probabilistic method to reach by an early detection the targeted partition of the fiber E.

In the previous section, we sketched a method that explores all points of the discriminant set Δ_p and determines iteratively a set of pairs of points of Δ_p which should be discarded together with their associated permutations of E. Thus the set of generators of the monodromy group decreases. The process stops when the orbit of the monodromy group splits, hence provides a partition of E.

Here we will proceed in the opposite direction: we start with an empty set of generators of the monodromy group and iteratively enlarge it, so we will coagulate the orbits to get the partition. More precisely, we consider few permutation $\sigma_1, \ldots, \sigma_l$ (with say e.g. $l = 5$), of the fiber E and decompose each of them into a product of cycles. These permutations should respect the partition of E, so the support (i.e. the large orbit) of each cycle should be contained in some subset E_i of the partition. This provides strong indications which allows to recover the partition with a good probability of success.

Indeed, each of these permutations are a product of a large number of transpositions, consequently they act on each subset E_i like a random permutation, so it is unlikely that the supports of two cycles are equal (or one contains another one) unless the support is equal to some E_i. Therefore in the exact setting, one can reconstruct the partition with high probability, already with $l = 2$ or $l = 3$. This analysis will be developed in a future work [9].

In presence of noise, this process might produce unlucky (i.e. false) cycles joining two distinct subsets of the partition, (if one of the loops defining a permutation crosses a "forbidden zone") that should be discarded. This problem can be overcome by taking a larger number of loops (e.g. $l = 5$) and performing the following preprocessing on f.

6.2 Pseudo singular points

We consider the clusters of points on $Y = F^{-1}(0)$ created by the deformation of the singular points of $X = f^{-1}(0)$. We can call them pseudo singular points. They are the solutions of a system of three "approximate equations" (F, F'_x, F'_y) in (x, y). We expect the same order of magnitude ϵ for the derivatives of f as for f. By a deformation of f, a common solution A to the three equations (f, f'_x, f'_y) produces a cluster of three points (A_{12}, A_{13}, A_{23}). A_{12} (respectively A_{13}, and A_{23}) is the common solution of (F, F'_x), respectively of (F, F'_y) and (F'_x, F'_y). We expect that the order of magnitude of the distance between A and A_{12}, or A_{13} is about $\sqrt{\epsilon}$ while the order of magnitude of the distance between A and A_{23} is only about ϵ. A practical way for locating such clusters is as follows.

1. Compute approximately a chosen subset $\mathcal{A}$ of the $(n-1)^2$ common solutions of (F'_x, F'_y) in $\mathbb{C}^2$. This can be done via resultants of a rational approximation of F.

2. Take the subset $\mathcal{B} \subseteq \mathcal{A}$ consisting of those $P \in \mathcal{A}$ for which $F(P)$ is close to 0 (tolerance $O(\epsilon)$).

We applied the procedure of computation of the pseudo singular points to our illustrative example. With a threshold equal to 0.1 or to 0.2, we obtain 11 pseudo singular points, (X has only 9 singular points). However, one of the pseudo singular points of Y very near to a singular point of X is obtained with distance larger than the two "extra" ones. This shows that we may have to accept more pseudo singular points of Y than singular points of X in order not to miss one.

6.3 A "Red Sea" preprocessing

We plan to apply a probabilistic algorithm, computing the permutations σ_j of the fiber E attached to few loops γ_j with $1 \leq j \leq l$. So we do not need to control all pseudo singular points but only those in the neighborhood of these loops.

We first choose "randomly" the loops γ_j, e.g. taking the border path OM_jN_j of quarter of a discs centered at the origin of the x-axis with radius 2 making random angles θ_j with the real axis of that plane. To avoid that the segments OM_j and ON_j (and even the arcs M_jN_j) cross the forbidden zone defined by presence of the projections of the pseudo singular points, we propose to deform the polynomial f as follows.

First, the set $\mathcal{S}$ of pseudo singular points of F in a (small) tubular neighborhood $\mathcal{N}$ of the l loops γ_j is computed.

Second, we drag the points of $\mathcal{S}$ out of the neighborhood $\mathcal{N}$ on a new set of points $\mathcal{T}$. Simultaneously we perform a (small) deformation $F + \Delta F$ of F imposing that $F + \Delta F$ admits singular points at $\mathcal{T}$. This last step can be achieved by a Newton like approximation algorithm.

Then we check that $F + \Delta F$ has no more pseudo singular points in the tubular neighborhood $\mathcal{N}$ of the l loops γ_j, we also adjust the induced small deformation on the fiber E. In the following steps, the deformed polynomial $F + \Delta F$ replaces F and this is considered as a preprocessing for the approximate factorization.

6.4 Interpolation and postprocessing

The previous procedures "recognize" the partition of the fiber E, into s disjoint subsets I_j with $1 \leq j \leq s$ of $\{1, \ldots, n\}$ and identify several points $P_i = (x_i, y_i)$ on the corresponding layer. Then it remains to compute, by interpolation at these points, candidate factors F_j. Then this factorization can be improved further by a postprocessing consisting of one or more Newton steps.

The general method is to find the coefficients of each F_j, e.g. in the monomial basis of dense polynomials of degree n_j, by imposing a family of linear conditions on these coefficients. These conditions express the vanishing of F_j at the P_i corresponding to I_j. Then one solves approximately the (overdetermined) linear system using QR or SVD techniques.

There are several ways to choose the points P_i.

1. Take m points x_i in the complement of the discriminant locus and all n points $P_{i,l}$ above each x_i organized in a grid.

2. Take double points or critical points of F corresponding to the branching points producing I_j, as well as the conditions coming from $F'_y(M_i) = 0$.

The first possibility with $m = 2n$ points widely spread in the x-axis gives satisfactory results.

Also, it seems that the distance (defined by the difference of the coefficients) between each candidate factor F_i to the corresponding f_i is smaller than the distance from their product to F. So Newton step(s) can improve the result. If $s = 2$ a Newton step amounts to solving approximately (SVD) in the coefficients of $(\delta F_1, \delta F_2)$ the overdetermined linear system

$$F - F_1 * F_2 - F_1 * \delta F_2 - F_2 * \delta F_1 = 0.$$

The second possibility has the advantage of focusing on the discriminant and critical loci. So the candidate F_i will have discriminant and a critical loci near a subset of the loci of F. This is more in the spirit of a geometric approach to the factoring problem.

Remark When the polynomial F is monic in y, the first possibility has the advantage that, for each triple of x-values (x_1, x_2, x_3), we can check the zero-sum condition for each set of points corresponding to a given j, then return to the main loop if it fails, or simply if needed deform the points in a fiber (as done e.g. in [35]) before the interpolation.

6.5 Illustrative example

With our illustrative example, we took 12 points in the rectangle $[-2..1, -1.5..1.5]$ included in the x-axis and the points above them (in the corresponding fibers) divided in two sets U and V. Therefore, U defines 36 homogeneous linear conditions on the ten coefficients of F_1; and respectively for F_2. We normalized by letting F_1 and F_2 to be monic in y. We have that the (normalized) maximal difference between the coefficients of f and its perturbation F is 0.01. We obtain two candidate factors F_1 and F_2 with a maximal difference with the coefficients of f_1 and f_2 of 0.02 and 0.03 while the maximal difference between the coefficients of F and $F_1 * F_2$ is 0.03. This can be improved with only one Newton step and we get new candidate factors $\tilde{F}_1$ and $\tilde{F}_2$ such that the maximal difference between the coefficients of F and $\tilde{F}_1\tilde{F}_2$ is 0.003 which seems satisfactory.

7. WHAT HAPPENS WITH HIGHER SINGULARITIES

In section 3.2 we made the important hypothesis that F was a small deformation of a composite polynomial $f = f_1 \cdots f_s$ such that $X_i = f_i^{-1}(0)$ are all smooth. So the only singularities of f were a fixed number of nodes (double points).

In this section, we discuss some issues and sketch some possibilities of solutions when f has more complicated singularities.

First of all let us recall the classical results on perturbation of multiple roots of an univariate polynomial: the typical behavior is the one of $x^d - \epsilon$ with ϵ small, so the root 0 is perturbed at a distance of d^{th}-root of ϵ.

This extends to systems of two bivariate polynomials: the more complicated the singularity is, the further the roots move away. Indeed, it suffices to analyze the projections on two lines, and clusters of several roots are not easy to locate.

One strategy used in one variable, and which can be generalized in two (or more) variables to locate the clusters of d-roots of a polynomial P is the following. First compute the zeros of P, P', P", ..., $P^{(d)}$. Then for each zero x_1 of $P^{(d)}$, such that $P^{(d+1)}(x_1)$ is big enough, there are two zeros of $P^{(d-1)}$ at a distance about $\sqrt{\epsilon}$, then three zeros of $P^{(d-2)}$ at a distance about the third root of ϵ, and so on till d zeros of P at a distance about the d^{th}-root of ϵ. See also section 3 in [36]. The situation is more complicated in two variables because one has to consider not the totally ordered iterations of derivations of P but the the only partially ordered iteration of the partial derivations of $F(x, y)$. This was studied by several authors including T. Ojika [28], and more recently by Giusti, Lecerf, Salvy, and Yakoubsohn [19], see also references in [19, 27].

In our setting such singularities may show up in three different ways: For one or more i, $X_i = f_i^{-1}(0)$ admits singularities, but

- We still have the property that for any two i and j, X_i and X_j intersect transversally (and their small deformations also intersect transversally) and form only nodes at these intersections. Then the strategy presented in the previous sections can be adapted and re-used with small changes, because we focus on the intersections.

- We still have the property that for any two i and j, the intersection point of X_i and X_j is a smooth point of X_i and a smooth point of X_j, but they are tangent. Then we can try to adapt the previous strategy but the analysis and the algorithm will be a bit more intricate.

- For two i and j, the intersection point of X_i and X_j is not a smooth point of X_i. Then the situation is even more complicated.

In all cases, a general strategy for obtaining an approximate factorization of F is to first approximate F by the nearby more singular polynomial.

Let us illustrate this paradigm on a case that we encountered when we experimented our algorithms. We created a benchmark of small degree polynomial with two factors, choosing two random polynomials f_1 and f_2 of degree 4 and 5 via the Maple command randpoly, and then deforming f into F. However, one such pair represented two curves X_1 and X_2 almost tangent at a point. Instead of finding the expected $4 \times 5 = 20$ common solutions to F'_x and F'_y, on which F almost vanishes, we found 21 even for quite small perturbations.

This can be explained as follows: to simplify the notations, let suppose that the tangent point correspond to two parabolas and is represented by the two equations $f_1 := y - x^2$ and $f_2 := y + 2x^2$, so $f = y^2 + x^2y - 2x^4$. By a small deformation, the nearby composite curve is formed by the union of two parabolas intersecting (transversally) at two points nearby 0. However, the partial derivatives F'_x and F'_y, of the deformed polynomial F have 3 common roots nearby 0, (because $f'_y = 2y + x^2$ and $f'_x = 2xy - 8x^3$ have a root in 0 of multiplicity 3) and F almost vanishes at these 3 points, so we can hardly distinguish 2 out of these 3 points.

The suggestion is to consider the second derivatives: $f"_{yy} = 2$, $f"_{xy} = 2x$ and $f'_{xx} = 2y - 24x^2$ and observe that $f"^2_{xy} - f'_{xx}f'_{yy}$ vanishes at 0 and that $f"_{xy}$ and $f"_{xx}$ have a common simple solution in 0. This last property is conserved by a small deformation and give a good estimate of the location of the cluster.

If among the pseudo singular points we correctly selected the subset $\mathcal{B}$ coming from true singularities, then we can apply a Newton iteration to bring those points closer to true singularities, as follows: Consider the space of polynomials that vanish at all $P \in \mathcal{B}$, and then replace F by the projection of F on that space. We implemented this, see NearestNpoly on [14]. If the initial errors were sufficiently small, say of size $O(\epsilon)$, then this process has quadratic convergence and should lead to an output that is no further than $O(\epsilon^2)$ away from an optimal output f (assuming transversal intersections, the distance from a $P \in \mathcal{B}$ to a singularity of f is $O(\epsilon)$, and hence $f(P) = O(\epsilon^2)$. Note that this gives another potential way to determine a lower bound for the distance from F to a nearest reducible polynomial. We hope to compute such a lower bound and compare to other approaches in the final version of this paper).

8. CONCLUDING REMARKS

We presented a new geometric approach and sketched an algorithm for factoring bivariate approximate polynomials. A draft implementation and some test files with examples are available at [14]. The algorithms work for bivariate approximate polynomials that are close enough to exactly factorable polynomials.

The presented algorithm assumes that the input polynomial approximates a product of polynomials defining smooth curves that intersect transversally. In the last section we discussed the effect of more complicated singularities. In the near future we will develop a complete implementation of our algorithm, test it on the existing benchmarks and try to extend our algorithm to this more general case. We can also incorporate in our algorithm, the zero-sums tests in order to speed up the iteration step before going to the interpolation step. Another interesting direction of research is to compare our approach with the one using the Ruppert-Gao matrix developed in [17].

9. REFERENCES

[1] C. Bajaj, J. Canny, T. Garrity, and J. Warren. Factoring rational polynomials over the complex numbers. *SIAM J. Comput.*, 22(2):318–331, 1993.

[2] G. Chèze and A. Galligo. Four lectures on polynomial absolute factorization. In *Solving polynomial equations*, volume 14 of *Algorithms Comput. Math.*, pages 339–392. Springer, Berlin, 2005.

[3] G. Chèze and A. Galligo. From an approximate to an exact absolute polynomial factorization. *J. Symbolic Comput.*, 41(6):682–696, 2006.

[4] G. Chèze and G. Lecerf. Lifting and recombination techniques for absolute factorization. Accepted for publication in Journal of Complexity, 2006.

[5] R. Corless, M. Giesbrecht, I. Kotsireas, M. van Hoeij, and S. Watt. Towards factoring bivariate approximate polynomials. In *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation (ISSAC 2001)*, pages 85–92. ACM, 2001.

[6] R. M. Corless, A. Galligo, I. S. Kotsireas, and S. M. Watt. A geometric-numeric algorithm for absolute factorization of multivariate polynomials. In *ISSAC '02: Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 37–45, New York, NY, USA, 2002. ACM Press.

[7] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The Singular Value Decomposition for polynomial systems. pages 195–207, Montréal, Canada, 1995. ACM.

[8] B. Deconinck and M. van Hoeij. Computing riemann matrices of algebraic curves. *PhysicaD*, 152:28–46, 2001.

[9] P. Diaconis and A. Galligo. A probabilistic approach to compute the orbits of a monodromy group of a plane curve. In preparation, 2007.

[10] A. Dimca. *Singularities and topology of hypersurfaces.* Universitext. Springer-Verlag, New York, 1992.

[11] I. Z. Emiris, A. Galligo, and H. Lombardi. Numerical univariate polynomial GCD. volume 32, pages 323–344, 1996.

[12] I. Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate gcds. *J. Pure Applied Algebra*, 117-118:229–251, 1997.

[13] A. Galligo. Real factorization of multivariate polynomials with integer coefficients. *Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI)*, 258(Teor. Predst. Din. Sist. Komb. i Algoritm. Metody. 4):60–70, 355, 1999.

[14] A. Galligo and M. van Hoeij. Implementation: http://www.math.fsu.edu/~hoeij/files/approxfact.

[15] A. Galligo and S. M. Watt. A numerical absolute primality test for bivariate polynomials. pages 217–224, Maui, USA, 1997. ACM.

[16] S. Gao. Factoring multivariate polynomials via partial differential equations. *Math. Comp.*, 72(242):801–822 (electronic), 2003.

[17] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In *ISSAC '04: Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 167–174, New York, NY, USA, 2004. ACM Press.

[18] M. Giesbrecht, G. Labahn, and W.-s. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. In *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation (ISSAC 2006)*, pages 116–123. ACM, 2006.

[19] M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn. On location and approximation of clusters of zeros: Case of embedding dimension one. *Foundations of Computational Mathematics*, 7(1):1–58, 2007.

[20] M. Hitz, E. Kaltofen, and Y. N. Lakshman. Efficient algorithms for computing the nearest polynomial with a real root and related problems. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, pages 205–212, New York, 1999. ACM.

[21] W. Huang, Y.and Wu, H. J. Stetter, and L. Zhi. Pseudofactors of multivariate polynomials. In *Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation*, pages 161–168, New York, 2000. ACM.

[22] E. Kaltofen. Polynomial factorization 1982–1986. In *Computers in mathematics (Stanford, CA, 1986)*, volume 125 of *Lecture Notes in Pure and Appl. Math.*, pages 285–309. Dekker, New York, 1990.

[23] E. Kaltofen. Polynomial factorization 1987–1991. In *Proc. LATIN '92*, volume 583 of *Lect. Notes Comput. Sci*, pages 294–313, New York, 1992. Springer Verlag.

[24] E. Kaltofen. Challenges of symbolic computation: My favorite open problems. *JSC*, 29(6):891–919, 2000.

[25] E. Kaltofen and J. May. On approximate irreducibility of polynomials in several variables. In *Proc. ISSAC03*, pages 161–168, 2003.

[26] E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials using singular value decomposition. 2006.

[27] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theor. Comput. Sci.*, 359(1):111–122, 2006.

[28] T. Ojika. Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations. *J. Math. Anal. Appl.*, 123:199–221, 1987.

[29] D. Rupprecht. *Elements de géométrie algébrique approchée: Etude du pgcd et de la factorisation.* PhD thesis, Univ. Nice Sophia Antipolis, 2000.

[30] T. Sasaki. Approximate multivariate polynomial factorization based on zero-sum relations. In *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation (ISSAC 2001)*, pages 284–291. ACM, 2001.

[31] T. Sasaki, T. Saito, and T. Hilano. Analysis of approximate factorization algorithm. I. *Japan J. Indust. Appl. Math.*, 9(3):351–368, 1992.

[32] T. Sasaki and M. Sasaki. A unified method for multivariate polynomial factorizations. *Japan J. Indust. Appl. Math.*, 10(1):21–39, 1993.

[33] T. Sasaki, M. Suzuki, M. Kolář, and M. Sasaki. Approximate factorization of multivariate polynomials and absolute irreducibility testing. *Japan J. Indust. Appl. Math.*, 8(3):357–375, 1991.

[34] A. Sommese, J. Verschelde, and C. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In *Application of Algebraic Geometry to Coding Theory, Physics and Computation*, pages 297–315. Kluwer Academic Publishers, 2001. Proceedings of a NATO Conference, February 25 - March 1, 2001, Eilat, Israel.

[35] A. Sommese, J. Verschelde, and C. Wampler. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM J. Numer. Anal.*, 40(6):2026–2046, 2002.

[36] A. Sommese, J. Verschelde, and C. Wampler. Numerical factorization of multivariate complex polynomials. *Theoretical Computer Science*, 315(2–3):651–669, 2004.

[37] Z. Zeng and B. H. Dayton. The approximate gcd of inexact polynomials. In *ISSAC '04: Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 320–327, New York, NY, USA, 2004. ACM Press.