Resultant-based methods for plane curves intersection problems

Laurent Busé, Houssam Khalil, Bernard Mourrain

Resultant-based methods for plane curves intersection problems

Laurent Busé[1], Houssam Khalil[2], and Bernard Mourrain[1]

[1] INRIA, Galaad, 2004 Route des Lucioles,
BP 93, 06902 Sophia Antipolis Cedex, France.
{lbuse,mourrain}@sophia.inria.fr
[2] Laboratoire de Mathématiques Appliquées de Lyon, UCBL and CNRS,
22 Avenue Claude Bernard, 69622 Villeurbanne Cedex, France,
khalil@maply.univ-lyon1.fr

Abstract. We present an algorithm for solving polynomial equations, which uses generalized eigenvalues and eigenvectors of resultant matrices. We give special attention to the case of two bivariate polynomials and the Sylvester or Bezout resultant constructions. We propose a new method to treat multiple roots, detail its numerical aspects and describe experiments on tangential problems, which show the efficiency of the approach. An industrial application of the method is presented at the end of the paper. It consists in recovering cylinders from a large cloud of points and requires intensive resolution of polynomial equations.

1 Introduction

We present an algorithm, which uses generalized eigenvalues and eigenvectors, for solving systems of two bivariate polynomials $p(x, y) = q(x, y) = 0$ in $\mathbb{R}$, with $p, q \in \mathbb{R}[x, y]$. Such a problem can be viewed as computing the intersection points of two implicitly defined plane algebraic curves, which is a key operation in Computer Aided Geometric Design. Several methods already exist to solve this problem, and we refer the interested reader to [7] for a general overview. The one that we present is based on a matrix formulation and the use of generalized eigenvalues and eigenvectors of two companion matrices built from the Sylvester or Bezout matrix of p and q, following methods initiated by Stetter (see [6, chapters 2] for a nice overview), which allows us to apply a preprocessing step and to develop efficient solvers for specific applications. A new improvment that we propose is the treatment of multiple roots, although these roots are usually considered as obstacles for numerical linear algebra techniques; the numerical difficulties are handled with the help of singular value decomposition (SVD) of the matrix of eigenvectors associated to each eigenvalue.

Similar methods, based on eigen-computations, have already been addressed, in particular in the papers [13,4,3]. Both methods project the roots of the system $p(x, y) = q(x, y) = 0$, say, on the x-coordinates, using a matrix formulation, and then lift them up, as well as their multiplicity. We follow the same path, but improve these results. Indeed, in [3] two multiplication maps have to be computed

to project the roots on the x-coordinates, whereas we only need one such map; in [13], the lifting of the x-coordinates of the roots is done by solving simultaneously two univariate polynomials, with approximate coefficients, obtained by substitutting the x-coordinnates by an approximation of an eigenvalue. This procedure is numerically very delicate. In our approach, we gather both the projection and the lifting step into a single eigen-problem and propose a stable way to treat multiple roots.

The paper is organized as follows. In section 2, we recall the needed tools. In section 3, we briefly give an overview of Bezout's theorem and explain how to use generalized eigenvalues and eigenvectors for solving two bivariate polynomial systems. Then, in section 4 we describe the numerical problems, how we remedy to them by using SVD, and give the algorithm. Finally, in section 5, we give some examples and show an industrial application.

2 Preliminaries

In this section, we first recall how a univariate polynomial can be solved via eigenvalue computations. The algorithm we provide in this paper can be seen as a generalization of this simple but important result. Then we briefly survey the very basic properties of the well-known Sylvester resultant. Finally we recall some elementary definitions of generalized eigenvalues and eigenvectors.

2.1 A univariate polynomial solver

Let $\mathbb{K}$ be any field and $f(x) := f_d x^d + f_{d-1}x^{d-1} + \cdots + f_1 x + f_0 \in \mathbb{K}[x]$. Using the standard Euclidean polynomial division it is easy to see that the quotient algebra $\mathcal{A} = \mathbb{K}[x]/I$, where I denotes the principal ideal of $\mathbb{K}[x]$ generated by the polynomial $f(x)$, is a vector space over $\mathbb{K}$ of dimension d with canonical basis $\{1, x, \ldots, x^{d-1}\}$.

Consider $M_x : \mathcal{A} \to \mathcal{A}$, the multiplication by x in $\mathcal{A}$. It is straightforward to check that the matrix of M_x in the basis $\{1, x, \ldots, x^{d-1}\}$ is given by

$$M_x = \begin{bmatrix} 0 & \cdots & 0 & -f_0/f_d \\ 1 & \ddots & & \vdots \\ \vdots & & 0 & \vdots \\ 0 & & 1 & -f_{d-1}/f_d \end{bmatrix},$$

(the column on the far right corresponds to the Euclidean division of x^d by f). The characteristic polynomial of M_x equals $\frac{(-1)^d}{f_d} f(x)$ and the roots of the polynomial f can be recovered, with their corresponding multiplicities, by computing the eigenvalues of the multiplication map M_x. See e.g. [12].

2.2 The Sylvester resultant

Let A be a commutative ring, which is assumed to be a domain, and suppose we are given two polynomials in $A[x]$ of respective degree d_0 and d_1

$$f_0(x) := c_{0,0} + c_{0,1}x + \cdots + c_{0,d_0}x^{d_0},$$
$$f_1(x) := c_{1,0} + c_{1,1}x + \cdots + c_{1,d_1}x^{d_1}.$$

The Sylvester matrix of f_0 and f_1 (in degree (d_0, d_1)) is the matrix whose columns contain successively the coefficients of the polynomials $f_0, xf_0, \ldots, x^{d_1-1}f_0, f_1$, $xf_1, \ldots, x^{d_0-1}f_1$ expanded in the monomial basis $\{1, x, \ldots, x^{d_0+d_1-1}\}$. More precisely, it is the matrix of the A-linear map

$$\begin{aligned} \sigma : A[x]_{<d_1} \oplus A[x]_{<d_0} &\to A[x]_{<d_0+d_1} \\ (q_0, q_1) &\mapsto f_0q_0 + f_1q_1 \end{aligned} \tag{1}$$

in the monomial bases $\{1, \ldots, x^{d_1-1}\}, \{1, \ldots, x^{d_0-1}\}$ and $\{1, \ldots, x^{d_0+d_1-1}\}$, where $A[x]_{<k}$ denotes the set of polynomials in $A[x]$ of degree lower or equal to $k-1$. It is thus a square matrix of size $d_0 + d_1$ which has the following form:

$$S := \begin{array}{c} \overbrace{\phantom{f_0 \cdots x^{d_1-1}f_0 \quad f_1 \cdots x^{d_0-1}f_1}}^{d_0+d_1} \\ f_0 \cdots x^{d_1-1}f_0 \quad f_1 \cdots x^{d_0-1}f_1 \\ \left(\begin{array}{ccc|ccc} c_{0,0} & & 0 & c_{1,0} & & 0 \\ \vdots & \ddots & & \vdots & \ddots & \\ & & c_{0,0} & & & c_{1,0} \\ c_{0,d_0} & & \vdots & c_{1,d_1} & & \vdots \\ & \ddots & \vdots & & \ddots & \vdots \\ 0 & & c_{0,d_0} & 0 & & c_{1,d_1} \end{array}\right) \begin{array}{c} 1 \\ x \\ \vdots \\ \vdots \\ \vdots \\ x^{d_0+d_1-1} \end{array} \end{array}$$

Proposition 1. *If A is a field, then we have* $\dim\ker(S) = \deg(\gcd(f_0, f_1))$.

Proof. Let $d(x) := \gcd(f_0, f_1)$. We easily check that the kernel of (1) is the set of pairs $\left(r(x)\frac{f_1(x)}{d(x)}, -r(x)\frac{f_0(x)}{d(x)}\right)$ where $r(x)$ is an arbitrary polynomial in $A[x]_{<\deg(d)}$. See also [11, corollary 5.3] for another description of this kernel.

Definition 1. *The resultant of the polynomials f_0 and f_1, denoted $Res(f_0, f_1)$, is the determinant of the Sylvester matrix of f_0 and f_1.*

The resultant $\mathrm{Res}(f_0, f_1)$ is thus an element in A. It has been widely studied in the literature, as it provides a way to eliminate the variable x from the polynomial system $f_0(x) = f_1(x) = 0$. Indeed,

Proposition 2. $\mathrm{Res}(f_0, f_1) = 0$ *if and only if either $c_{0,d_0} = c_{1,d_1} = 0$, or either $f_0(x)$ and $f_1(x)$ have a common root in the algebraic closure of the quotient field* $\mathrm{Frac}(A)$ *of A (or equivalently f_0 and f_1 are not coprime in* $\mathrm{Frac}(A)[x]$*).*

Proof. See e.g. [12, IV §8].

Another matrix, called the Bezout matrix, can be used to compute $\mathrm{Res}(f_0, f_1)$. We now suppose, without loss of generality, that $d_1 \geq d_0$.

Definition 2. *The Bezoutian of the polynomials f_0 and f_1 is the element in $A[x, y]$ defined by*

$$\Theta_{f_0,f_1}(x, y) := \frac{f_0(x)f_1(y) - f_1(x)f_0(y)}{x - y} = \sum_{i,j=0}^{d_1-1} \theta_{i,j} x^i y^j,$$

and the Bezout matrix is $B_{f_0,f_1} := (\theta_{i,j})_{0\leq i,j\leq d_1-1}$.

The matrix B_{f_0,f_1} is a $d_1 \times d_1$-matrix which is symmetric. It is thus a smaller matrix than the Sylvester matrix, but has more complicated entries, and its determinant still equals the resultant of f_0 and f_1 (up to a power of c_{1,d_1}):

Proposition 3 ([11] §5.4). *We have*

$$\det(B_{f_0,f_1}) = (-1)^{\frac{1}{2}d_1(d_1-1)} (c_{1,d_1})^{d_1-d_0} \mathrm{Res}(f_0, f_1).$$

Moreover, if A is a field then $\dim \ker(B_{f_0,f_1}) = \deg(\gcd(f_0, f_1))$.

2.3 Generalized eigenvalues and eigenvectors

Let A and B be two matrices of size $n \times n$. A *generalized eigenvalue* of A and B is a value in the set

$$\lambda(A, B) := \{\lambda \in \mathbb{C} : \det(A - \lambda B) = 0\}.$$

A vector $x \neq 0$ is called a *generalized eigenvector* associated to the eigenvalue $\lambda \in \lambda(A, B)$ if $Ax = \lambda Bx$. The matrices A and B have n generalized eigenvalues if and only if $\mathrm{rank}(B) = n$. If $\mathrm{rank}(B) < n$, then $\lambda(A, B)$ can be finite, empty, or infinite. Note that if $0 \neq \mu \in \lambda(A, B)$ then $1/\mu \in \lambda(B, A)$. Moreover, if B is invertible then $\lambda(A, B) = \lambda(B^{-1}A, I) = \lambda(B^{-1}A)$, which is the ordinary spectrum of $B^{-1}A$.

Recall that an $n \times n$ matrix $T(x)$ with polynomial entries can be equivalently written as a polynomial with $n \times n$ matrix coefficients. If $d = \max_{i,j}\{\deg(T_{ij}(x))\}$, we obtain $T(x) = T_d x^d + T_{d-1}x^{d-1} + \cdots + T_0$, where T_i are $n \times n$ matrices.

Definition 3. *The companion matrices of $T(x)$ are the two matrices A, B given by*

$$A = \begin{pmatrix} 0 & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I \\ T_0^t & T_1^t & \cdots & T_{d-1}^t \end{pmatrix}, \; B = \begin{pmatrix} I & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & I & 0 \\ 0 & \cdots & 0 & -T_d^t \end{pmatrix}.$$

And we have the following interesting property:

Proposition 4. *With the above notation, the following equivalence holds:*

$$T^t(x)v = 0 \Leftrightarrow (A - xB) \begin{pmatrix} v \\ xv \\ \vdots \\ x^{d-1}v \end{pmatrix} = 0.$$

Proof. A straightforward computation.

3 The intersection of two plane algebraic curves

From now on we assume that $\mathbb{K}$ is an algebraically closed field and that $p(x, y)$ and $q(x, y)$ are two polynomials in $\mathbb{K}[x, y]$. Our aim will be to study and compute their common roots. This problem can be interpreted geometrically. Polynomials p and q define two algebraic curves in the affine plane $\mathbb{A}^2$ (having coordinates (x, y)), and we would like to know their intersection points.

Hereafter we will consider systems $p(x, y) = q(x, y) = 0$ having only a *finite* number of roots. This condition is not quite restrictive since it is sufficient (and necessary) to require that polynomials p and q are coprime in $\mathbb{K}[x, y]$ (otherwise we divide them by their gcd).

In the case where one of the curve is a line, i.e. one of the polynomial has degree 1, the problem is reduced to solving a univariate polynomial. Indeed, one may assume e.g. that $q(x, y) = y$ and thus we are looking for the roots of $p(x, 0) = 0$. Let us denote them by $z_1, \ldots, z_s$. Then we know that

$$p(x, 0) = c(x - z_1)^{\mu_1}(x - z_2)^{\mu_2} \cdots (x - z_s)^{\mu_s},$$

where the z_i's are assumed to be distinct and c is a non-zero constant in $\mathbb{K}$. The integer μ_i, for all $i = 1, \ldots, s$, is called the *multiplicity* of the root z_i, and it turns out that $\sum_{i=1}^{s} \mu_i = \deg(p)$ if $p(x, y)$ does not vanish at infinity in the direction of the y-axis (in other words, if the homogeneous part of p of highest degree does not vanish when $y = 0$). This later condition can be avoid in the projective setting: let $p^h(x, y, t)$ be the homogeneous polynomial obtained by homogenizing $p(x, y)$ with the new variable t, then we have

$$p(x, 0, t) = c(x - z_1 t)^{\mu_1}(x - z_2 t)^{\mu_2} \cdots (x - z_s t)^{\mu_s} t^{\mu_\infty},$$

where μ_∞ is an integer corresponding to the multiplicity of the root at infinity, and $\mu_\infty + \sum_{i=1}^{s} \mu_i = \deg(p)$. Moreover, it turns out that the roots $z_1, \ldots, z_s$ and their corresponding multiplicities can be computed by eigenvalues and eigenvectors computation (see section 2.1).

In the following we generalize this approach to the case in which p and q are bivariate polynomials of arbitrary degree. For this we first need to recall the notion of multiplicity in this context. Then we will show how to recover the roots from multiplication maps.

3.1 Intersection multiplicity

Let $p(x,y), q(x,y)$ be two coprime polynomials in $\mathbb{K}[x,y]$, $\mathcal{C}_p$ and $\mathcal{C}_q$ the corresponding algebraic plane curves, $I := (p,q)$ the ideal they generate in $\mathbb{K}[x,y]$ and $\mathcal{A} := \mathbb{K}[x,y]/I$ the associated quotient ring. We denote by $z_1 = (x_1, y_1), \ldots, z_s = (x_s, y_s)$ the distinct intersection points in $\mathbb{A}^2$ of $\mathcal{C}_p$ and $\mathcal{C}_q$ (i.e. the distinct roots of the system $p(x,y) = q(x,y) = 0$).

A modern definition of the intersection multiplicity of $\mathcal{C}_p$ and $\mathcal{C}_q$ at a point z_i is (see [8, §1.6])

$$i(z_i, \mathcal{C}_p \cap \mathcal{C}_q) := \dim_{\mathbb{K}} \mathcal{A}_{z_i} < \infty,$$

where the point $z_i \in \mathbb{A}^2$ is here abusively (but usually) identified with its corresponding prime ideal $(x - x_i, y - y_i)$ in $\mathbb{K}[x,y]$, $\mathcal{A}_{z_i}$ denoting the ordinary localization of the ring $\mathcal{A}$ by this prime ideal. As a result, the finite $\mathbb{K}$-algebra $\mathcal{A}$ (which is actually finite if and only if $p(x,y)$ and $q(x,y)$ are coprime in $\mathbb{K}[x,y]$) can be decomposed as the direct sum

$$\mathcal{A} = \mathcal{A}_{z_1} \oplus \mathcal{A}_{z_2} \oplus \cdots \oplus \mathcal{A}_{z_s}$$

and consequently $\dim_{\mathbb{K}} \mathcal{A} = \sum_{i=1}^{s} i(z_i, \mathcal{C}_p \cap \mathcal{C}_q)$.

The intersection multiplicities can be computed using a resultant. The main idea is to project "algebraically" the intersection points on the x-axis. To do this let us see both polynomials p and q in $A[y]$ where $A := \mathbb{K}[x]$, that is to say as univariate polynomials in y with coefficients in the ring A which is a domain; we can rewrite

$$p(x,y) = \sum_{i=0}^{d_1} a_i(x) y^i, \quad q(x,y) = \sum_{i=0}^{d_2} b_i(x) y^i. \tag{2}$$

Their resultant (with respect to y) is an element in A which is non-zero, since p and q are assumed to be coprime in $A[y]$, and which can be factorized, assuming that $a_{d_1}(x)$ and $b_{d_2}(x)$ do not vanish simultaneously, as (see proposition 2)

$$\mathrm{Res}(p,q) = c \prod_{i=1}^{r} (x - \alpha_i)^{\beta_i} \tag{3}$$

where the α_i's are distinct elements in $\mathbb{K}$ and $\{\alpha_1, \ldots, \alpha_r\} = \{x_1, \ldots, x_s\}$ (as sets). For instance, if all the $x_i's$ are distinct then we have $r = s$ and $\alpha_i = x_i$ for all $i = 1, \ldots, s$. Moreover, we have (see e.g. [8, §1.6]):

Proposition 5. *For all $i \in \{1, \ldots, r\}$, the integer β_i equals the sum of all the intersection multiplicities of the points $z_j = (x_j, y_j) \in \mathcal{C}_p \cap \mathcal{C}_q$ such that $x_j = \alpha_i$:*

$$\beta_i = \sum_{z_j = (x_j, y_j) \mid x_j = \alpha_i} i(z_j, \mathcal{C}_p \cap \mathcal{C}_q).$$

As a corollary, if all the x_i's are distinct (this can be easily obtained by a linear change of coordinates (x,y)) then $i(z_i, \mathcal{C}_p \cap \mathcal{C}_q)$ is nothing but the valuation of $\mathrm{Res}(p,q)$ at $x = x_i$ (i.e. the exponent of the largest power of $(x - x_i)$ which divides

$\mathrm{Res}(p,q)$). Another corollary of this result is the well-known Bezout theorem for algebraic plane curves, which is better stated in the projective context. Let us denote by $p^h(x,y,t)$ and $q^h(x,y,t)$ the homogeneous polynomials in $\mathbb{K}[x,y,t]$ obtained from p and q by homogenization with the new variable t.

Proposition 6 (Bezout theorem). *If p and q are coprime then the algebraic projective plane curves associated to $p^h(x,y,t)$ and $q^h(x,y,t)$ intersect in $\deg(p)\deg(q)$ points in $\mathbb{P}^2$, counted with multiplicities*

Proof. It follows directly from the previous proposition and the well-known fact that $\mathrm{Res}(p^h,q^h)$ is a homogeneous polynomial in $\mathbb{K}[x,t]$ of degree $\deg(p)\deg(q)$ (see e.g. [12], [17]).

3.2 Intersection points

We take again the notation of (2) and (3). We denote by $S(x)$ the Sylvester matrix of $p(x,y)$ and $q(x,y)$ seen in $A[y]$ (assuming that both has degree at least one in y); thus $\det(S(x)) = \mathrm{Res}(p,q)$. From (3), we deduce immediately that $\det(S(x))$ vanishes at a point $x_0 \in \mathbb{K}$ if and only if either there exists y_0 such that $p(x_0,y_0) = q(x_0,y_0) = 0$ or $a_{d_1}(x_0) = b_{d_2}(x_0) = 0$ (in which case the intersection point is at infinity). Therefore one may ask the following question: *given a point x_0 such that $\det(S(x_0)) = 0$, how may we recover all the possible points y_0 such that $p(x_0,y_0) = q(x_0,y_0) = 0$?*

Suppose we are given such a point x_0, and assume that $a_{d_1}(x_0)$ and $b_{d_2}(x_0)$ are not both zero. If $\ker(S(x_0)^t)$ is of dimension one, then it is easy to check that there is only one point y_0 such that $p(x_0,y_0) = q(x_0,y_0) = 0$, and moreover any element in $\ker(S(x_0)^t)$ is a multiple of the vector $[1, y_0, \cdots, y_0^{d_1+d_2-1}]$. Therefore to compute y_0, we only need to compute a basis of $\ker(S(x_0)^t)$ and to compute the quotient of its second coordinate by its first one. In case of multiple roots, this construction can be generalized as follows:

Proposition 7. *With the above notation, let $\Lambda_1, \cdots, \Lambda_d$ be any basis of the kernel $\ker(S(x_0)^t)$, $\mathbf{\Lambda}$ be the matrix whose i^{th} row is the vector Λ_i, Δ_0 be the $d\times d$-submatrix of $\mathbf{\Lambda}$ corresponding to the first d columns and Δ_1 be the $d\times d$-submatrix of $\mathbf{\Lambda}$ corresponding to the columns of index $2,3,\ldots,d+1$. Then $\lambda(\Delta_1,\Delta_0)$ is the set of roots of the equations $p(x_0,y) = q(x_0,y) = 0$ (i.e. the set of y-coordinates of the intersection points of $\mathcal{C}_p$ and $\mathcal{C}_q$ above $x = x_0$).*

Proof. Let $g(y) := \gcd(p(x_0,y), q(x_0,y)) \in \mathbb{K}[y]$ and $k_1 = \deg(p(x_0,y)), k_2 = \deg(q(x_0,y))$. By proposition 1, we know that $d = \deg(g(y))$. Consider the map

$$\begin{aligned} \mathrm{rem}_{x_0} : \mathbb{K}[y]_{<k_1+k_2} &\to \mathbb{K}[y]_{<d} \\ t(y) &\mapsto \mathrm{remainder}(t(y), g(y)), \end{aligned}$$

which sends $t(y)$ on the remainder of its Euclidean division by $g(y)$. Observe that, for all $i = 0,\ldots,k_1+k_2-2$, $\mathrm{rem}_{x_0}(y^{i+1}) = \mathrm{rem}_{x_0}(y\,\mathrm{rem}_{x_0}(y^i))$. Consequently, $\mathrm{rem}_{x_0}(y^{i+1}) = \mathtt{M}_y \mathrm{rem}_{x_0}(y^i)$, where $\mathtt{M}_y$ is the operator of multiplication by y in

the quotient ring $\mathbb{K}[y]/(g(y))$ (see section 2.1). Let Δ be the matrix of rem_{x_0} in the monomial basis of $\mathbb{K}[y]_{<k_1+k_2}$, Δ_0 a submatrix of Δ of size $d \times k$ and Δ_1 the submatrix obtained by shifting the index of columns of Δ_0 in Δ by 1. Then by the previous remark, we have $\Delta_1 = \mathtt{M}_y\Delta_0$. In particular, if we choose the first d columns of Δ, we have $\Delta_0 = \mathrm{Id}$ so that $\lambda(\Delta_0, \Delta_1)$ are the roots of $g(y) = p(x_0, y) = q(x_0, y) = 0$.

Now, $S(x_0)$ is the matrix in the monomial bases of the map (1), denoted here σ_{x_0}. By construction $\mathrm{rem}_{x_0} \circ \sigma_{x_0} = 0$ and $\dim \ker(\sigma_{x_0}) = \deg(g) = \dim \mathrm{Im}(\mathrm{rem}_{x_0})$, which shows that the rows of the matrix of rem_{x_0} form a basis of $\ker(S(x_0)^t) \equiv \mathbb{K}[y]_{<d}$. Since by any change of basis of $\ker(S(x_0)^t)$ the equality $\mathtt{M}_y\Delta_0 = \Delta_1$ remains true, the claim is proved.

Remark 1. In this theorem, observe that the construction of Δ_0 and Δ_1 is always possible, that is to say that $\mathbf{\Lambda}$ has at least $d + 1$ columns, because we assumed that both polynomials p and q depend on the variable y which implies that the number of rows in the matrix $S(x_0)$ is always strictly greater than $\deg(\gcd(p(x_0, y), q(x_0, y)))$.

The previous theorem shows how to recover the y-coordinates of the roots of the system $p = q = 0$ from the Sylvester matrix. However, instead of the Sylvester matrix we can use the Bezout matrix to compute $\mathrm{Res}(p, q) \in A[x]$ (see section 2). This matrix has all the required properties to replace the Sylvester matrix in all the previous results, except in proposition 7. Indeed, the Bezout matrix being smaller than the Sylvester matrix, the condition given in remark 1 is not always fulfilled; using the Bezout matrix in proposition 7 requires that for all roots x_0 of $\mathrm{Res}(p, q)$,

$$\max(\deg(p(x_0, y), \deg(q(x_0, y)) > \deg(\gcd(p(x_0, y), q(x_0, y)).$$

This condition may fail: take for instance $x_0 = -1$ in the system

$$\begin{cases} p(x, y) = x^2y^2 - 2y^2 + xy - y + x + 1 \\ q(x, y) = y + xy \end{cases}$$

Note that the use of the Bezout matrix gives, in practice, a faster method because it is a smaller matrix than the Sylvester matrix, even if its computation takes more time (the savings in the eigen-computations is greater).

3.3 The algorithm

We are now ready to describe our resultant-based solver. According to proposition 4, we replace the computation of the resultant, its zeroes, and the kernel of $S^t(x_0)$ (or the Bezout matrix) for each root x_0 by the computation of generalized eigenvalues and eigenvectors of the associated companion matrices A and B. This computation can be achieved with the QZ algorithm [9].

Algorithm 1 EXACT ARITHMETIC VERSION

1. *Compute the Bezout matrix* $B(x)$ *of* p *and* q.
2. *Compute the associated companion matrices* A *and* B *(see proposition 4).*
3. *Compute the generalized eigenvalues and eigenvectors of* (A, B). *The eigenvalues give the* x*-coordinates of the intersection points, and their multiplicities give the sum of the intersection multiplicities of the points above them (see proposition 5). The eigenvector spaces give bases of* $\ker(B(x_0))$ *in proposition 7; their dimensions give the degree of the gcd of* p *and* q *at these points.*
4. *For each value* x_0,
   (a) *If the number of associated eigenvectors is at least* $\max(\deg(p(x_0, y)), \deg(q(x_0, y)))$, *which is the size of* $B(x_0)$, *compute* Δ_0 *and* Δ_1 *by using a basis of* $\ker(S(x_0)^t)$,
   (b) *if not, then compute* Δ_0 *and* Δ_1 *by using the eigenvectors associated to* x_0.
5. *Compute the eigenvalues of* (Δ_1, Δ_0) *which give the* y*-coordinates of the intersection points above* x_0 *(see proposition 7).*

4 Numerical difficulties

The following difficulties are found in numerical computations:

a. In order to compute the x-coordinates of intersection points, it is necessary to compute the real generalized eigenvalues of (A, B). Numerically, some of these values can contain a nonzero imaginary part. Generally such a problem is encountered for multiple eigenvalues.
b. What do we mean by a numerical multiple x-coordinate?
c. How do we choose the linearly independent vectors among the computed eigenvectors?

In order to solve these problems, we need the singular value decomposition (SVD for short).

4.1 SVD as remedy

Let us recall known results on the singular value decomposition.

Theorem 1 ([9]). *For a real matrix* A *of size* $m \times n$, *there exist two orthogonal matrices*

$$U = [u_1, \cdots, u_m] \in \mathbb{R}^{m \times m}, \text{ and } V = [v_1, \cdots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$U^t A V = diag(\sigma_1, \cdots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min\{m, n\}, \text{ and}$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0.$$

σ_i are called the ith singular value of A, u_i and v_i are respectively the ith left and right singular vectors.

Definition 4. *For a matrix A, the numerical rank of A is defined by:*

$$\operatorname{rank}(A,\epsilon) = \min\{\operatorname{rank}(B) : \left\| \frac{A}{\|A\|} - B \right\| \leq \epsilon\},$$

where $\|\cdot\|$ denotes either $\|\cdot\|_2$, the spectral norm, or $\|\cdot\|_F$, the Frobenius norm.

Theorem 2. *([9]) If $A \in \mathbb{R}^{m\times n}$ is of rank r and $k < r$ then*

$$\sigma_{k+1} = \min_{\operatorname{rank}(B)=k} \|A - B\|.$$

Theorem 2 shows that the smallest singular value is the distance between A and all the matrices of rank $< p = \min\{m, n\}$. Thus if $r_\epsilon = \operatorname{rank}(A, \epsilon)$ then

$$\sigma_1 \geq \cdots \geq \sigma_{r_\epsilon} > \epsilon\sigma_1 \geq \sigma_{r_{\epsilon+1}} \geq \cdots \geq \sigma_p.$$

As a result, in order to determine the rank of a matrix A, we find the singular values $(\sigma_i)_i$ so that

$$\operatorname{rank}(A,\epsilon) = \max\{i : \frac{\sigma_i}{\sigma_1} > \epsilon\}.$$

For difficulty "a" we choose a small $\epsilon \in \mathbb{R}$ and consider as real any eigenvalue whose imaginary part is of absolute value less than ϵ.

For difficulty "b", we choose possibly a different small $\epsilon \in \mathbb{R}$ and gather the points which lay in an interval of size ϵ. According to the following proposition, the proposed algorithm remains effective:

Proposition 8. *Assume that $\{\xi_1 \cdots \xi_n\} \subseteq \lambda(A, B)$ with corresponding generalized eigenvectors $v_1^{(1)}, \cdots, v_{k_1}^{(1)}, \cdots, v_1^{(n)}, \cdots, v_{k_n}^{(n)}$ such that $\sum_{i=1}^n k_i < d_1 + d_2$. We define $\mathbf{\Lambda}$ to be the matrix whose rows are $v_1^{(1)}, \cdots, v_{k_1}^{(1)}, \cdots, v_1^{(n)}, \cdots, v_{k_n}^{(n)}$ and Δ_0 (resp. Δ_1) to be its first (resp. second) left $(k_1+\cdots+k_n)\times(k_1+\cdots+k_n)$ block. Then, the generalized eigenvalues of Δ_1 and Δ_0 give the set of y-coordinates of the intersection points above $\xi_1 \cdots \xi_n$; in other words, clusters of x-coordinates are thus gathered and regarded as only one point whose multiplicity equals the sum of the multiplicities of the gathered points.*

Proof. We define $\mathbf{\Lambda}_i$ as the matrix corresponding to eigenvectors of ξ_i, it is a matrix of size $k_i \times (d_1 + d_2)$, then $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 \\ \vdots \\ \mathbf{\Lambda}_n \end{bmatrix}$. For $\mathbf{\Lambda_i}$, we take $\Delta_0^{(i)}$ the first left $k_i \times (k_1+\cdots+k_n)$ block, and $\Delta_1^{(i)}$ the second block. Then, according to the proof of proposition 7, $\Delta_1^{(i)} = M_{y,\xi_i}\Delta_0^{(i)}$, where $M_{y,\xi}$ is the matrix of multiplication by y in the quotient ring $\mathbb{K}[y]/\gcd(p(\xi_i, y), q(\xi_i, y))$. It follows that $\Delta_1 = M_y\Delta_0$ where

$$M_y = \begin{pmatrix} M_{y,\xi_1} & & 0 \\ & \ddots & \\ 0 & & M_{y,\xi_n} \end{pmatrix}.$$

Moreover, Δ_0 is invertible and thus $\lambda(\Delta_1, \Delta_0) = \lambda(M_y) = \{\lambda(M_{y,\xi_i}), i = 1 \cdots n\}$.

For difficulties "c", it is necessary to use the singular value decomposition: If x a generalized eigenvalue of A and B, and E the matrix of the associated eigenvectors, then the singular value decomposition of E is

$$E = U\Sigma V^t, \quad \text{with } \Sigma = \begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}$$

and r is the rank of E. Let

$$E' = E \cdot V = U \cdot \Sigma = [\sigma_1 u_1 \cdots \sigma_r u_r \, 0 \cdots 0] = [e'_1 \cdots e'_r \, 0 \cdots 0].$$

The matrix E' is also a matrix of eigenvectors, because the changes are done on the columns, and V is invertible. Hence E' and E have the same rank r, and $E'' = [e'_1 \cdots e'_r]$ is a matrix of linearly independent eigenvectors. Consequently E'' can be used to build Δ_0 and Δ_1.

We now give the complete algorithm which can be used in the presence of singular points:

Algorithm 2 Numerical approximation of the roots of a bivariate polynomial system.

1. *Compute the Bezout matrix $B(x)$ of p and q, and compute the matrices A and B.*
2. *Compute the generalized eigenvalues and eigenvectors of (A, B).*
3. *Eliminate the imaginary eigenvalues and the values at infinity, and gather the close points (with the input ϵ).*
4. *For each of all the close points represented by ξ, take the matrix E of the associated eigenvectors.If their number $\geq d_1 + d_2$ then E gives a basis of* $\ker S^t(\xi)$
5. *Do a singular value decomposition of E.*
6. *Calculate the rank of E by testing the σ_i until $\sigma_{r+1} < \sigma_r \epsilon$ is found (the rank is thus determined to be r).*
7. *Let E'' be the r first columns of $E' = E \cdot V$. Define Δ_0 as the first $r \times r$ block in E'', and Δ_1 as the second $r \times r$ block in E''.*
8. *Compute the generalized eigenvalues of (Δ_1, Δ_0).*

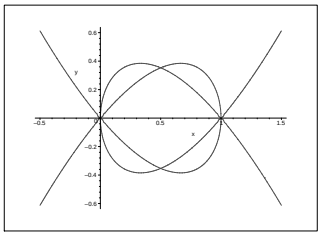
Remark 2. On step 3, there are very often eigenvalues at infinity, because the size of A and B is larger than the degree of the resultant. Therefore, generally one obtains more eigenvalues than roots, some of the eigenvalues being at infinity.

5 Numerical experiments and application

Our algorithm[1] is implemented in C++, in the library C++ called "SYNAPS" (SYmbolic and Numeric ApplicationS)[2]. It uses the linear algebra library "LAPACK" (in FORTRAN), for approximate computation of eigenvalues and eigenvectors. The following computations were done on a Pentium4, 3.06 Ghz computer.

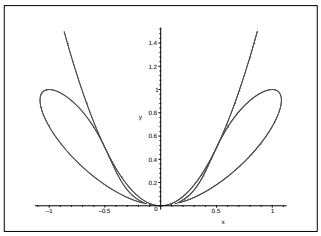
5.1 Examples

Example 1: $\begin{cases} p = y^2 - x^2 + x^3 \\ q = y^2 - x^3 + 2x^2 - x \end{cases}$



```
The intersection points are: (epsilon=10^-6).
(x1, y1) = (0,0)           of multiplicity 2
(x2, y2) = (0.5, -0.35)    of multiplicity 1
(x3, y3) = (0.5,0.35)      of multiplicity 1
(x4, y4) = (1, -8.2e-25)   of multiplicity 2
Execution time = 0.004s
```

Example 2: $\begin{cases} p = x^4 - 2x^2y + y^2 + y^4 - y^3 \\ q = y - 2x^2 \end{cases}$

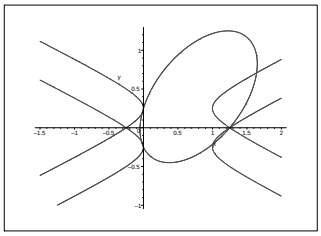


```
The intersection points are: (epsilon=10^-6).
(x1, y1) = (1.6e-09,0)     of multiplicity 4
(x2, y2) = (-0.5,0.5)      of multiplicity 2
(x3, y3) = (0.5,0.5)       of multiplicity 2
Execution time = 0.005s
```

[1] http://www-sop.inria.fr/galaad/logiciels/synaps/html/bivariate__res_8H-source.html

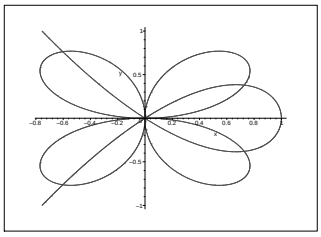
[2] http://www-sop.inria.fr/galaad/logiciels/synaps/

Example 3: $\begin{cases} p = 400y^4 - 160y^2x^2 + 16x^4 + 160y^2x - 32x^3 - 50y^2 + 6x^2 + 10x + \frac{25}{16} \\ q = y^2 - x + x^2 - \frac{5}{6} - \frac{1}{6} \end{cases}$



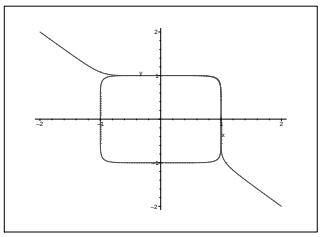
```
the intersection points are: (epsilon=10^-6).
(x1, y1) = (1.4e-16, -0.25) of multiplicity 1
(x2, y2) = (1.4e-16,0.25)   of multiplicity 1
(x3, y3) = (-0.03, -0.16)   of multiplicity 1
(x4, y4) = (-0.01,0.18)     of multiplicity 1
(x5, y5) = (1.64,0.70)      of multiplicity 1
(x6, y6) = (1, -0.21)       of multiplicity 1
(x7, y7) = (1.25,1.5e-08)   of multiplicity 2
Execution time = 0.007s
```

Example 4: $\begin{cases} p = x^6 + 3x^4y^2 + 3x^2y^4 + y^6 - 4x^2y^2 \\ q = y^2 - x^2 + x^3 \end{cases}$



```
the intersection points are: (epsilon=10^-3).
(x1, y1) = (-3e-16,2e-17)  of multiplicity 8
(x2, y2) = (-0.60,-0.76)   of multiplicity 1
(x3, y3) = (-0.60,0.76)    of multiplicity 1
(x4, y4) = (0.72, -0.37)   of multiplicity 1
(x5, y5) = (0.72,0.37)     of multiplicity 1
Execution time = 0.011s
```

Example 5: $\begin{cases} p = x^9 + y^9 - 1 \\ q = x^{10} + y^{10} - 1 \end{cases}$



```
The intersection points are: (epsilon=2.10^-2).
(x1, y1) = (-0.01,1)      of multiplicity 9
(x2, y2) = (1,0)          of multiplicity 9
Execution time = 0.111s
```

5.2 Table of comparison

We give here a table[3] comparing our method (called GEB below), with two other algorithms implemented in SYNAPS for solving the curve intersection problem. The first uses Sturm-Habicht sequences [10], the second uses normal form computations (the "Newmac" method) [14], [15].

Ex.	Degree	N.s	Execution time			M
			GEB	Sturm	Newmac	
ex001	3,3	4	0.004	0.002	0.029	1.1e-16
ex002	4,2	3	0.005	0.005	0.026	8.8e-16
ex003	4,2	7	0.007	0.010	0.029	6.8e-10
ex004	6,3	5	0.011	0.005	0.041	1.7e-15
ex005	9,10	2	0.111	0.130	1.705	6.6e-15
ex006	6,4	4	0.013	0.010	0.063	1.8e-10
ex007	8,7	41	0.089	0.211	0.306	1.7e-05
ex008	8,6	27	0.085	0.091	0.283	2.1e-05
ex009	6,4	2	0.006	0.003	0.032	3.2e-11
ex010	4,3	5	0.005	0.003	0.033	2.5e-13
ex011	4,3	3	0.004	0.003	0.030	7.1e-15
ex012	5,4	2	0.008	0.010	0.053	4.7e-14
ex013	6,5	5	0.021	0.02	0.058	1.7e-11
ex014	11,10	5	0.083	0.194	0.332	4.4e-06
ex015	6,5	4	0.015	0.01	0.049	2.6e-15
ex016	8,7	41	0.088	0.208	0.305	1.7e-05

(N.s denotes the number of solutions and M denotes the number $\max(\max_i |p(x_i, y_i)|, \max_i |q(x_i, y_i)|)$ where (x_i, y_i) are the computed solutions.

[3] See `http://www-sop.inria.fr/galaad/data/curve2d/`.

5.3 Cylinders passing through five points

An important problem in CAD modeling is the extraction of a set of geometric primitives properly describing a given 3D point cloud obtained by scanning a real scene. If the extraction of planes is considered as a well-solved problem, the extraction of circular cylinders, these geometric primitives are basically used to represent "pipes" in an industrial environment, is not easy and has been recently addressed. In this section, we describe an application of our algorithm to this problem which has been experimented by Thomas Chaperon from the MENSI[4] company.

In [2], the extraction of circular cylinders was done in two steps: after the computation of estimated unit normals of the given 3D set of points, their Gaussian images give the possible cylinders as great circles on the Gaussian sphere. In order to avoid the estimation of the unit normals another approach was presented in [1]. First, being given 5 points randomly selected in our 3D point cloud, the author gave a polynomial system whose roots correspond to the cylinders passing through them (recall that 5 is the minimum number of points defining generically a finite number of cylinders, actually 6 in the *complex* numbers). Then the method consists in finding these cylinders, actually their direction, for almost all set of 5 points in the whole point could, and extract the "clusters of directions" as a primitive cylinder. In [5], motivated by another application in metrology, the authors carefully studied the problem of finding cylinders passing through 5 given points in the space; in particular they produce two polynomials in three homogeneous variables whose roots corresponds to the expected cylinders. In the sequel, using the same polynomial system that we will briefly recall, we show that our new solver can speed-up the solving step.

Modelisation. We briefly recall from [5] how the problem of finding the 6 complex cylinders passing through 5 (sufficiently generic) points can be translated into a polynomial system. We actually only seek the direction of these cylinders since their axis and radius follow then easily (see e.g. [2][appendix A]). We will denote by p_1, p_2, p_3, p_4, p_5 the five given points defining six cylinders. We first look for the cylinders passing only through the first four points that we can assume, w.l.o.g., to be $p_1 = (0,0,0), p_2 = (x_2,0,0), p_3 = (x_3,y_3,0)$ and $p_4 = (x_4,y_4,z_4)$. We also denote by $\mathbf{t} = (l,m,n)$, where $l^2+m^2+n^2 = 1$, the unitary vector identifying a direction in the 3D space; note that $\mathbf{t}$ also identified with the projective point $(l : m : n)$ in $\mathbb{P}^2$.

Let π be the plane passing through the origin and orthogonal to $\mathbf{t}$ and let (X,Y,Z) be a coordinate system such that the two first axes are in π and the third one has direction $\mathbf{t}$. Among all the coordinate changes sending (x,y,z) onto (X,Y,Z) we choose the one given by the following matrix, where $\rho = m^2+n^2$:

$$\begin{pmatrix} \rho & -\frac{lm}{\rho} & -\frac{nl}{\rho} \\ 0 & \frac{n}{\rho} & -\frac{m}{\rho} \\ l & m & n \end{pmatrix}.$$

[4] `http://www.mensi.fr/`

The orthogonal projection q_i of p_i on π has coordinates, in the system (X, Y, Z) :

$$(X_i, Y_i, Z_i) := \left(\rho x_i - \frac{lm}{\rho} y_i - \frac{nl}{\rho} z_i, \frac{n}{\rho} y_i - \frac{m}{\rho} z_i, 0\right).$$

The points p_1, p_2, p_3, p_4 belong to a cylinder of direction $\mathbf{t}$ if and only if the points q_1, q_2, q_3, q_4 are cocyclic in π, i.e. if and only if

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ X_1^2 + Y_1^2 & X_2^2 + Y_2^2 & X_3^2 + Y_3^3 & X_4^2 + Y_4^2 \end{vmatrix} = 0.$$

We denote this previous determinant by $\mathcal{C}_{p_1,p_2,p_3,p_4}(l, m, n)$. The coordinates of q_1 and q_2 satisfy $X_1 = Y_1 = 0$, $X_2 = \rho x_2$, and $Y_2 = 0$. Moreover, for $i = 3, 4$ we have

$$X_i^2 + Y_i^2 = |q_i|^2 = (\mathbf{t}.\mathbf{t})|p_i|^2 - (t.p_i)^2.$$

Therefore, by developing the determinant $\mathcal{C}_{p_1,p_2,p_3,p_4}(l, m, n)$ one can show that it equals (see [5] for more details)

$$x_2^2(m^2 + n^2) \begin{vmatrix} l & x_3 & x_4 \\ m & y_3 & y_4 \\ n & 0 & z_4 \end{vmatrix} - x_2 \begin{vmatrix} m & y_3 & y_4 \\ n & 0 & z_4 \\ 0 & (\mathbf{t}.\mathbf{t})|p_3|^2 - (t.p_3)^2 & (\mathbf{t}.\mathbf{t})|p_4|^2 - (t.p_4)^2 \end{vmatrix}.$$

Seeing $\mathbf{t}$ as a projective point in $\mathbb{P}^2$, this equation thus defines an algebraic curve of degree 3 in $\mathbb{P}^2$.

We deduce that a cylinder of direction $\mathbf{t} = (l, m, n)$ goes through the 5 points p_1, p_2, p_3, p_4, p_5 if (a necessary condition)

$$\mathcal{C}_{p_1,p_2,p_3,p_4}(l, m, n) = 0 \quad \text{and} \quad \mathcal{C}_{p_1,p_2,p_3,p_5}(l, m, n) = 0. \tag{4}$$

These two equations define two algebraic curves in $\mathbb{P}^2$ and hence intersect into exactly 9 points (counted with multiplicity) from the Bezout theorem. But the directions $\boldsymbol{p_1p_2}$, $\boldsymbol{p_1p_3}$ and $\boldsymbol{p_2p_3}$ are solutions of (4) but correspond to a cylinder if and only if they are of multiplicity at least 2. Consequently the system (4) gives us the 6 solutions we are looking for and three extraneous solutions that we know by advance and that we can easily eliminate after the resolution of (4).

For instance, there are exactly 6 *real* cylinders passing through the 5 following points:

$$\begin{array}{lll} x_1 := 0, & y_1 := 0, & z_1 := 0, \\ x_2 := 1, & y_2 := 0, & z_2 := 0, \\ x_3 := \frac{1}{2}, & y_3 := \frac{\sqrt{3}}{2}, & z_3 := 0, \\ x_4 := \frac{1}{2}, & y_4 := \frac{1}{2\sqrt{3}}, & z_4 := \frac{\sqrt{2}}{\sqrt{3}}, \\ x_4 := \frac{1}{2}, & y_4 := \frac{1}{2\sqrt{3}}, & z_4 := -\frac{\sqrt{2}}{\sqrt{3}}. \end{array}$$

By applying our algorithm we obtain the 6 directions:

```
(-0.81722,-1.8506e-17) of multiplicity 1
(-0.40929,-0.70721)    of multiplicity 1
(-0.40929,0.70721)     of multiplicity 1
(0.40929,-0.70721)     of multiplicity 1
(0.40929,0.70721)      of multiplicity 1
(0.81722,-1.8506e-17)  of multiplicity 1
```

Experimentations. We now turn to the experimentation of our method. We took 1000 sets of 5 random points, with integer coordinates between -100 and 100 and computed the 6 (complex) corresponding cylinders. Comparing the previously mentioned methods, we obtained the following timing:

method	GEB	Sturm	Newmac
time (sec.)	0.67	1.83	1.61

which shows a significant improvement, in the context of intensive computation.

6 Conclusion

In this paper, we presented a new algorithm for solving systems of two bivariate polynomial equations using generalized eigenvalues and eigenvectors of resultant matrices (both Sylvester and Bézout types). We proposed a new method to treat multiple roots, described experiments on tangential problems which show the efficiency of the approach and provided an industrial application consisting in recovering cylinders from a large cloud of points. We also performed a first numerical analysis of our approach; we plan to investigate it further following the ideas developed in [16], in order to control a posteriori the error on the roots.

References

1. T. Chaperon. A note on the construction of right circular cylinders through five 3d points. Technical report, Centre de Robotique, École des mines de Paris, january 2003.
2. T. Chaperon, F. Goulette, and C. Laurgeau. Extracting cylinders in full 3d data using a random sampling method and the gaussian image. In T. Ertl, B. Girod, G. Greiner, H. Nieman, and H. Seidel, editors, *Vision, Modeling and Visualization (VMV'01)*, Stuttgart, November 2001.
3. R. Corless, P. Gianni, and B. Trager. A reordered schur factorization method for zero-dimensional polynomial systems with multiple roots. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, pages 133–140 (electronic), New York, 1997. ACM.
4. R. M. Corless, L. Gonzalez-Vega, I. Necula, and A. Shakoori. Topology determination of implicitly defined real algebraic plane curves. *An. Univ. Timişoara Ser. Mat.-Inform.*, 41(Special issue):83–96, 2003.
5. O. Devillers, B. Mourrain, F. P. Preparata, and P. Trebuchet. Circular cylinders through four or five points in space. *Discrete Comput. Geom.*, 29(1):83–104, 2003.

6. A. Dickenstein and I. Emiris, editors. *Solving Polynomial Equations: Foundations, Algorithms, and Applications*, volume 14 of *Algorithms and Computation in Mathematics*. Springer-Verlag, april 2005.
7. M. Elkadi and B. Mourrain. Symbolic-numeric tools for solving polynomial equations and applications. In A. Dickenstein and I. Emiris, editors, *Solving Polynomial Equations: Foundations, Algorithms, and Applications.*, volume 14 of *Algorithms and Computation in Mathematics*, pages 125–168. Springer, 2005.
8. W. Fulton. *Introduction to intersection theory in algebraic geometry*, volume 54 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 1984.
9. G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.
10. L. González-Vega and I. Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Comput. Aided Geom. Design*, 19(9):719–743, 2002.
11. N. Kravitsky. Discriminant varieties and discriminant ideals for operator vessels in Banach space. *Integral Equations Operator Theory*, 23(4):441–458, 1995.
12. S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, third edition, 2002.
13. D. Manocha and J. Demmel. Algorithms for intersecting parametric and algebriac curves ii; multiple intersections. *Computer vision, Graphics and image processing: Graphical models and image processing*, 57:81–100, 1995.
14. B. Mourrain. A new criterion for normal form algorithms. In *Applied algebra, algebraic algorithms and error-correcting codes (Honolulu, HI, 1999)*, volume 1719 of *Lecture Notes in Comput. Sci.*, pages 430–443. Springer, Berlin, 1999.
15. B. Mourrain and P. Trebuchet. Solving projective complete intersection faster. In *Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation (St. Andrews)*, pages 234–241 (electronic), New York, 2000. ACM.
16. S. Oishi. Fast enclosure of matrix eigenvalues and singular values via rounding mode controlled computation. *Linear Algebra Appl.*, 324(1-3):133–146, 2001. Special issue on linear algebra in self-validating methods.
17. B. L. Van der Waerden. *Modern algebra, Vol. II.* New-York, Frederick Ungar Publishing Co, 1948.