ELSEVIER SCIENCE B.V. [DTD 4.1.0]
JOURNAL **LAA** ARTICLE No. **6787**
PAGES **1-17** DISPATCH **July 7, 2000**
in all correspondence concerning this paper refer to: **LAA 6787**
PROD. TYPE: **DISK**

A

ELSEVIER Linear Algebra and its Applications 00 (2000) 00–00

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

Computing of a specified root of a polynomial system of equations using eigenvectors☆

Didier Bondyfalat [a,1], Bernard Mourrain [a], Victor Y. Pan [b,*,2]

[a] *INRIA, SAGA, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France*
[b] *Department of Mathematics and Computer Science, Lehman College, City University of New York, 250 Bedford Park B'vard West, Bronx, NY 10468-1589, USA*

Received 18 February 2000; accepted 26 May 2000

Submitted by R.A. Brualdi

Abstract

We propose new techniques and algorithms for the solution of a polynomial system of equations by matrix methods. For such a system, we seek its specified root, at which a fixed polynomial takes its maximum or minimum absolute value on the set of roots. We unify several known approaches and simplify the solution substantially, in particular in the case of an over-constrained polynomial system having only a simple root or a few roots. We reduce the solution to the computation of the eigenvector of an associated dense matrix, but we define this matrix implicitly, as a Schur complement in a sparse and structured matrix, and then modify the known methods for sparse eigenvector computation. This enables the acceleration of the solution by roughly factor D, the number of roots. Our experiments show that the computations can be performed numerically, with no increase of the computational precision, and the iteration converges to the specified root quite fast.

AMS classification: 15A18; 12E12; 65F15; 68Q40

Keywords: Polynomial systems of equations; Resultants; Matrix eigenproblem; Sparse matrices; Over-constrained polynomial systems; Parameterized polynomial systems; Gröbner basis

☆ In 1998 some results of this paper have been presented at the ACM-SIGSAM Annual International Symposium on Symbolic and Algebraic Computation (ISSAC'98) in Rostock, Germany.

* Corresponding author. Tel.: +1-212-960-81.
E-mail addresses: dbondy@sophia.inria.fr (D. Bondyfalat), mourrain@sophia.inria.fr (B. Mourrain), vpan@alpha.lehman.cuny.edu (V.Y. Pan).

1 Partially supported by European ESPRIT project FRISCO, LTR 21.024.
2 Supported by NSF Grants CCR 9625344 and CCR 9732206 and by PSC CUNY Awards 668365 and 669363.

0024-3795/00/$ - see front matter
PII: S 0 0 2 4 - 3 7 9 5 (0 0) 0 0 1 9 0 - 7

1. Introduction

The resolution of polynomial systems of equations is a major problem of computer algebra with applications in areas such as Robotics, Computer Vision, Computational Geometry, and Signal Processing. Polynomial systems are usually defined by a few monomials but may be hard to solve, both from the computational complexity point of view and from the numerical stability point of view. In spite of long and intensive study of this subject and substantial progress (see, e.g., [1,3,11,14–16]), many theoretical and practical problems remain largely open, in particular the computation of a specific root of a polynomial system, which maximizes or minimizes the absolute value of a given functional.

Among a few available approaches, the most popular is the Gröbner basis method, but it has a high arithmetic computational cost and requires exact or modular computations, that is, it is prone to severe numerical stability problems and cannot be applied safely with floating point arithmetic. Moreover, when the method is applied to compute a single root, most of the computation applies to all roots, thus increasing the overall arithmetic cost by a large extra factor of at least D, the number of roots of a given system. Other known methods have other deficiencies. For instance, Newton-type iterative methods converge quickly to a root in some cases but provide no means to converge to a selected root.

The most promising approach to overcoming these difficulties seems to be the reduction of the solution of a polynomial system to linear algebra computations, namely, either to the unsymmetric matrix eigenproblem or the generalized matrix eigenproblem. Our paper explores this approach, which we divide into two stages:

1. of defining appropriate reduction to the matrix eigenvector problem and
2. of devising its effective solution algorithm.

Though the reduction of a polynomial system to the matrix eigenproblem is a well known and well studied topic [1,16,25], there are several variations of this reduction requiring various computational cost and leading to different frameworks, which may facilitate or complicate the subsequent computation. In particular the reduction stage based on the computation of a Gröbner basis has major deficiencies already cited, furthermore; it leads to expensive computations with very large dense matrices. Our contribution at this stage is a simple unifying approach, based on the study of the associated maps, operators and functionals, which enables more effective control over the structure and sparsity of the matrices involved (see Section 4). As a result, we reduce the solution of a polynomial system to the eigenvector problem for the transposed map of multiplication by an element a in the associated quotient algebra of multivariate polynomials. The matrix $\mathtt{M}_a^{\mathtt{t}}$ of this map is dense, and its computation would be nearly as expensive as a Gröbner basis computation, as this matrix contains essentially the same information as a Gröbner basis. We do not compute this matrix explicitly, however, but define it as the Schur complement in a Sylvester-like matrix, $\mathtt{S}^{\mathtt{t}}$, which is sparse and structured. This enables us to approximate the eigenvectors of $\mathtt{M}_a^{\mathtt{t}}$ by means of the known methods for sparse unsymmetric eigenproblem (such

as the Arnoldi, the unsymmetric Lanczos and the power iterations), which we adjust to apply them to a Schur complement (defined implicitly), rather than to the matrix itself.

The resulting algorithms can be implemented with floating point arithmetic, and their computational cost is defined by the number of monomials in the input equations and the dimension N of the associated resultant matrix. More precisely, we solve the polynomial system by an iterative process that converges linearly to the solution and uses the order of N^2 flops (arithmetic operations) in each recursive step, whereas the known approaches use the order of N^3 flops.

Furthermore, our techniques enable us to direct the iterative process towards a specified root of a polynomial system maximizing or minimizing the absolute value of any selected polynomial.

This gives our approach some important advantages for practical solution of a polynomial system, in particular, for the special but practically highly important case where we deal with an overconstrained polynomial system, which has only a few roots or only a single root. Moreover, our modification of the Gröbner basis approach enables us to direct the computation towards the approximation of only a specified root and to preserve the matrix structure and sparsity. This modification also has a major advantage of avoiding the most expensive stage of the recomputation of the Gröbner basis when the input parameters change. In both overconstrained and Gröbner basis cases, we substantially decrease the computational cost, namely, by the factor D (the number of roots), versus the known algorithms.

We organize our paper as follows. We elaborate the transition from a polynomial system to an eigenvector problem in Section 2. We show the transition from a Sylvester-like matrix to a multiplication map and outline iterative solution of the eigenvector problem in Section 3. In Section 4, we specify three approaches to the construction of these structured matrices and give a demonstration for a parameterized polynomial system. The initial results of our experiments, performed for several samples of practical problems and reported in Section 5, show the expected behavior of the algorithms. Even for large input polynomial systems, the algorithms converge sufficiently fast to a specified root minimizing or maximizing the absolute value of a fixed polynomial.

2. Reduction of the solution of a polynomial system to matrix eigenproblem

In this section, we formalize the reduction of the solution of a polynomial system to the matrix eigenproblem (cf. [1,5,19,22,25]). We denote by $R = \mathbb{C}[x_1, \ldots, x_n]$ the ring of polynomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$, with coefficients in the field of complex numbers $\mathbb{C}$. Many of our results are valid for any algebraically closed field $\mathbb{K}$. $\mathbb{N}$ will denote the set of nonnegative integers.

Let $f_1, \ldots, f_m$ be m polynomials of R, defining the polynomial system $f_1(\mathbf{x}) = 0, \ldots, f_m(\mathbf{x}) = 0$. Let I be the ideal generated by these polynomials. We consider

the case, where the quotient algebra $\mathscr{A} = R/I$ *is of finite dimension D over* $\mathbb{C}$. This implies that the set of roots or solutions $\mathscr{Z}(I) = \{\zeta \in \mathbb{C}^n;\ f_1(\zeta) = \cdots = f_m(\zeta) = 0\}$ is finite: $\mathscr{Z}(I) = \{\zeta_1, \ldots, \zeta_d\}$ with $d \leqslant D$. To this set of roots, we associate a fundamental set of orthogonal idempotents $\mathbf{e}_1, \ldots, \mathbf{e}_d$ satisfying

$$\mathbf{e}_1 + \cdots + \mathbf{e}_d \equiv 1 \quad \text{and} \quad \mathbf{e}_i\,\mathbf{e}_j \equiv \begin{cases} 0 & \text{if } i \neq j, \\ \mathbf{e}_i & \text{if } i = j, \end{cases}$$

such that if $I = Q_1 \cap \cdots \cap Q_d$ is the minimal primary decomposition of I, we have $\mathbf{e}_i\mathscr{A} \sim R/Q_i$, where $\mathscr{A}_i = \mathbf{e}_i\,\mathscr{A}$ is a local algebra, for the maximal ideal $\mathbf{m}_{\zeta_i}$ defining the root ζ_i. This also implies that $\mathscr{A} = \mathscr{A}_1 \oplus \cdots \oplus \mathscr{A}_d$ (see [19, p. 717; 29]).

We denote by $\widehat{R}$ the dual space of R, that is, the set of maps (linear forms) from R to $\mathbb{C}$ and by $\widehat{\mathscr{A}}$ the dual space of $\mathscr{A}$, that is, the set of elements $\Lambda \in \widehat{R}$ such that $\Lambda(I) = 0$ (also denoted by $I^{\perp}$).

For any element $a \in \mathscr{A}$, we denote by

$$\begin{aligned} M_a : \mathscr{A} &\to \mathscr{A} \\ b &\mapsto a\,b \end{aligned}$$

the map of multiplication by a in $\mathscr{A}$, and we denote by

$$\begin{aligned} M_a^{\mathrm{t}} : \widehat{\mathscr{A}} &\to \widehat{\mathscr{A}} \\ \Lambda &\mapsto a \cdot \Lambda \end{aligned}$$

its transposed map. By definition of the transposed operator, for any $\Lambda \in \widehat{\mathscr{A}}$, we have $\Lambda(a\,b) = \Lambda(M_a(b)) = M_a^{\mathrm{t}}(\Lambda)(b) = (a \cdot \Lambda)(b)$.

Theorem 2.1. *There exists a basis of* $\mathscr{A}$ *such that for all* $a \in \mathscr{A}$ *the matrix* $\mathtt{M}_a$ *of* M_a *in this basis is of the form*

$$\mathtt{M}_a = \begin{pmatrix} \mathtt{M}_{a,1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathtt{M}_{a,d} \end{pmatrix},$$

where $\mathtt{M}_{a,i}$ *of the form*

$$\mathtt{M}_{a,i} = \begin{pmatrix} a(\zeta_i) & & * \\ & \ddots & \\ \mathbf{0} & & a(\zeta_i) \end{pmatrix}$$

is the matrix of multiplication by a in $\mathscr{A}_i$.

Proof. The block decomposition of $\mathtt{M}_a$ is induced by the decomposition of $\mathscr{A}$ in the direct sum of subalgebras $\mathscr{A}_i$. The matrices $\mathtt{M}_{a,i}$, $a \in \mathscr{A}$, are commuting and thus have a triangular decomposition in a common basis [30]. See also [3]. □

In the case of a simple root ζ_i, we have $\mathtt{M}_{a,i} = \mathrm{diag}(a(\zeta_i))$. If ζ_i is a multiple root, it may happen that $\mathtt{M}_{a,i}$ has several Jordan blocks and its set of eigenvectors is not of dimension 1.

It is also possible to characterize the eigenspace of M_a^{t} in terms of evaluations and differentials at the roots ζ_i, which are defined as follows. At first, for any point $\zeta \in \mathbb{C}$, let us write

$$\begin{aligned} \mathbf{1}_\zeta \ : \ & R \to \mathbb{C} \\ & p \mapsto p(\zeta) \end{aligned}$$

and note that $\mathbf{1}_\zeta \in \widehat{\mathscr{A}}$ if and only if $\zeta \in \mathscr{Z}(I)$. Furthermore, for any pair of $a, b \in \mathscr{A}$, we have

$$M_a^{\mathrm{t}}(\mathbf{1}_{\zeta_i})(b) = \mathbf{1}_{\zeta_i}(a\,b) = a(\zeta_i)b(\zeta_i) = a(\zeta_i)\mathbf{1}_{\zeta_i}(b)$$

so that $\mathbf{1}_{\zeta_i}$ is an eigenvector of M_a^{t} for the eigenvalue $a(\zeta_i)$. In the case of the systems of polynomial equations having multiple roots, a complete description of the eigenspace involves higher order differential forms, specifically, the maps (linear forms)

$$\begin{aligned} \mathbf{d}_\zeta^{\mathbf{a}} \ : \ & R \to \mathbb{C} \\ & p \mapsto \frac{1}{\prod_{i=1}^n a_i!} \left(d_{x_1}\right)^{a_1} \cdots \left(d_{x_n}\right)^{a_n} (p)(\zeta), \end{aligned} \tag{1}$$

where $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$, d_{x_i} is the derivative with respect to the variable x_i. We write $\mathbf{d}_\zeta^{\mathbf{a}} = (\mathbf{d}_{1,\zeta})^{a_1} \cdots (\mathbf{d}_{n,\zeta})^{a_n}$. See [7,17,18] for further details.

Proposition 2.2. *The eigenspace of M_a^{t} associated to the eigenvalue $a(\zeta_i)$ is generated by $\mathbf{1}_{\zeta_i}$ and by some linear combinations of the differentials $\mathbf{d}_{\zeta_i}^{\mathbf{a}}$.*

As for any pair of $a, b \in \mathscr{A}$, the multiplication maps M_a, M_b commute with each other, it follows that they share common eigenvector spaces. Indeed, we have the following property (see [19]):

Proposition 2.3. *The common eigenvectors of M_a^{t} for all $a \in \mathscr{A}$ are the non-zero multiples of $\mathbf{1}_{\zeta_i}$ for $i = 1, \ldots, d$.*

Remark 2.4. If the root ζ_i is simple, the eigenvector associated to the eigenvalue $a(\zeta_i)$ is $\mathbf{1}_{\zeta_i}$ (up to a scalar factor).

Remark 2.5. If $(\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_D})$ is a basis of $\mathscr{A}$, then the coordinates of $\mathbf{1}_{\zeta_i}$ in its dual basis are $(\zeta_i^{\alpha_1}, \ldots, \zeta_i^{\alpha_D})$, by the definition of the dual basis.

Summarizing, we arrive at the following algorithm for the computation of the simple roots.

Algorithm 2.6. Computing the simple roots of a polynomial system $f_1 = \cdots = f_m = 0$.

1. Compute the transpose $\mathtt{M}_a^{\mathrm{t}}$ of the matrix of multiplication by $a \in \mathscr{A}$ in a basis of the form $(1, x_1, \ldots, x_n, \ldots)$.
2. Compute its right eigenvectors $\mathbf{v}_i = (v_{i,1}, v_{i,x_1}, \ldots, v_{i,x_n}, \ldots)$ or, equivalently, the left eigenvectors $\mathbf{v}_i^{\mathrm{t}}$ of the matrix M_a for $i = 1, \ldots, d$, so that $\mathbf{v}_i^{\mathrm{t}} M t_a = a(\xi_i)\, \mathbf{v}^{\mathrm{t}}$ for all i.
3. For $i = 1, \ldots, d$, compute and output

$$\zeta_i = \left(\frac{v_{i,x_1}}{v_{i,1}}, \ldots, \frac{v_{i,x_n}}{v_{i,1}} \right).$$

Stage 2 of Algorithm 2.6 amounts to the computation of the left eigenvectors of the matrix $\mathtt{M}_a$. The known effective algorithms perform this stage by recursively pre- and/or post-multiplying the matrix $\mathtt{M}_a$ by vectors. Next, we will show how to perform such multiplications efficiently.

3. Definition of multiplication maps and the approximation of their eigenvectors

Algorithm 2.6 reduces the solution of a polynomial system of equations to the computation of the eigenvectors of the transpose $\mathtt{M}_a^{\mathrm{t}}$ of the matrix of a multiplication map in $\mathscr{A}$. Usually, however, the matrices $\mathtt{M}_a$ and $\mathtt{M}_a^{\mathrm{t}}$ are not available directly from the input polynomial system but in many interesting cases can be recovered from Sylvester-like matrices $\mathtt{S}$ representing multiples of the input equations. We will show the construction of the latter matrices $\mathtt{S}$ in Section 4. Now, we specify their desired properties.

Hypothesis 3.1. The matrix $\mathtt{S}$ is a square matrix of the form

$$\mathtt{S} = \begin{pmatrix} \mathtt{A} & \mathtt{B} \\ \mathtt{C} & \mathtt{D} \end{pmatrix} \tag{2}$$

such that

1. its rows are indexed by monomials $(\mathbf{x}^\alpha)_{\alpha \in F}$ for a fixed set $F \subset \mathbb{N}^n$,
2. the set of monomials $B_0 = (\mathbf{x}^\alpha)_{\alpha \in E_0}$ indexing the rows of the block $(\mathtt{A}\ \mathtt{B})$ is a basis of $\mathscr{A} = R/(f_1, \ldots, f_m)$,
3. the columns of $\binom{\mathtt{A}}{\mathtt{C}}$ represent the elements $\mathbf{x}^\alpha f_0$ for a fixed f_0 and for $\alpha \in E_0$, expressed as linear combinations of the monomials $(\mathbf{x}^\beta)_{\beta \in F}$,
4. the columns of $\binom{\mathtt{B}}{\mathtt{D}}$ represent some multiples of the polynomials $f_1, \ldots, f_m$, expressed as linear combinations of the monomials $(\mathbf{x}^\beta)_{\beta \in F}$,
5. the block $\mathtt{D}$ is invertible.

For any matrix $\mathtt{S}$ satisfying these hypotheses, we may obtain the map of multiplication by f_0 modulo $f_1, \ldots, f_n$ as follows (cf. [5,20]):

Proposition 3.2. *Under Hypothesis* 3.1, *the matrix of multiplication by* $a = f_0$ *in the basis* $B_0 = (\mathbf{x}^\alpha)_{\alpha \in E_0}$ *of the quotient algebra* $\mathscr{A} = R/(f_1, \ldots, f_m)$ *is the Schur complement of* $\mathtt{D}$ *in* $\mathtt{S}$:

$$\mathtt{M}_{f_0} = \mathtt{A} - \mathtt{B}\,\mathtt{D}^{-1}\,\mathtt{C}. \tag{3}$$

The Sylvester-like matrices $\mathtt{S}$ that we will construct in the next section and, therefore, also their blocks $\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}$ are structured matrices. They are also sparse—the number of non-zero terms per columns is bounded by the number of monomials in the polynomials $f_0, \ldots, f_m$, which is practically small compared to the size of these matrices. The matrices also have a structure that generalizes the structure of Toeplitz matrices to the multivariate case and can be exploited to simplify multiplication of such a matrix by a vector (see [6,20–22,27]). Even without using this structure, however, we may multiply such matrices by vectors fast, based solely on exploiting their sparsity. In particular, the $N \times N$ matrices $\mathtt{S}$ are frequently sufficiently sparse to allow their multiplication by a vector using the order of N flops and words of storage space, whereas explicit computation of $\mathtt{M}_a$ would have required the order of N^3 flops and the order of N^2 words of storage space. Thus, we will multiply the matrices $\mathtt{M}_a = \mathtt{A} - \mathtt{B}\,\mathtt{D}^{-1}\,\mathtt{C}$ and $(\mathtt{M}_a - \sigma I)^{-1}$ by vectors efficiently by reducing these operations to multiplication of the matrices

$$\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}^{-1} \quad \text{and} \quad \mathtt{S}_\sigma^{-1} = \begin{pmatrix} \mathtt{A} - \sigma I & \mathtt{B} \\ \mathtt{C} & \mathtt{D} \end{pmatrix}^{-1}$$

by vectors. To multiply the matrices $\mathtt{D}^{-1}$ and $\mathtt{S}_\sigma^{-1}$ by vectors, we apply the conjugate gradient (GC) method, which amounts to recursive multiplication of *sparse* matrices $\mathtt{D}, \mathtt{D}^{\mathrm{t}}, \mathtt{S}_\sigma$ and $\mathtt{S}_\sigma^{\mathrm{t}}$ by vectors. The product $(\mathtt{M}_a - \sigma I)^{-1}\mathbf{v}$ is obtained as the leading subvector of the vector $\mathtt{S}_\sigma^{-1}\binom{\mathbf{v}}{\mathbf{0}}$ because the matrix $\mathtt{M}_a - \sigma I$ is a leading principal submatrix, that is, a north-western block, of $\mathtt{S}_\sigma$. Similarly, the vector $\mathbf{v}^{\mathrm{t}}\,(\mathtt{M}_a - \sigma\, I)^{-1}$ is obtained as the leading subvector of the vector $(\mathbf{v}, \mathbf{0})^{\mathrm{t}}\,\mathtt{S}_\sigma^{-1}$.

Proposition 3.3. *Let* $\mathtt{S}$ *be an* $N \times N$ *satisfying Hypothesis* 3.1 *and assume that* M *flops suffice to multiply the matrix* $\mathtt{S}$ *by a vector. Then* $\mathrm{O}((M+N)N)$ *flops suffice to multiply the matrices* $\mathtt{M}_a = \mathtt{A} - \mathtt{B}\,\mathtt{D}^{-1}\,\mathtt{C}$, $(\mathtt{M}_a - \sigma I)^{-1}$ (*for a scalar* σ) *and their transposes by a vector.*

Proof. The conjugate gradient (CG) algorithm enables multiplication of $\mathtt{D}^{-1}$ by a vector in $\mathrm{O}((M+N)N)$ flops, which dominates the cost of the multiplication of the matrix $\mathtt{M}_a$ and its transpose by a vector. Now, observe that $(\mathtt{M}_a - \sigma\, I)^{-1}$ is a leading principal submatrix of the matrix

$$\mathtt{S}_\sigma^{-1} = \begin{pmatrix} \mathtt{A} - \sigma I & \mathtt{B} \\ \mathtt{C} & \mathtt{D} \end{pmatrix}^{-1}.$$

Therefore, the vector $(\mathtt{M}_a - \sigma I)^{-1}\mathbf{v}$ is the leading subvector of the vector $\mathtt{S}_\sigma^{-1}\begin{pmatrix}\mathbf{v}\\ \mathbf{0}\end{pmatrix}$ and similarly the vector $\mathbf{v}^{\mathrm{t}}(\mathtt{M}_a - \sigma I)^{-1}$ is the leading subvector of the vector $(\mathbf{v}, \mathbf{0})^{\mathrm{t}}\,\mathtt{S}_\sigma^{-1}$. The CG algorithm computes the latter vector in $\mathrm{O}((M+N)\,N)$ flops, and we may similarly treat multiplication of the matrix $(\mathtt{M}_a - \sigma I)^{\mathrm{t}}$ and its inverse by a vector (cf. [24]). □

Remark 3.4. The computational cost bound of Proposition 3.3 relies on the cost bound for the CG algorithm. In practice, this algorithm and various other known iterative methods for sparse linear systems converge much faster [12,13,24].

Due to the above proposition and remark, it seems most effective to approximate the eigenvectors of the matrix $\mathtt{M}_a^{\mathrm{t}}$ by the known algorithms that rely on repeated multiplication of the matrices $\mathtt{M}_a, \mathtt{M}_a^{\mathrm{t}}$ for $(\mathtt{M}_a - \sigma I)^{-1}$, $(\mathtt{M}_a^{\mathrm{t}} - \sigma I)^{-1}$ by vectors, such as Arnoldi, the unsymmetric Lanczos, and the (shifted inverse) power algorithms (cf. [12, pp. 362–364, 499–507]). In Section 5 we describe some results of our preliminary experiments based on the shifted inverse power iteration. The algorithms repeatedly multiply $\mathtt{M}_a^{\mathrm{t}}$ (resp. $(\mathtt{M}_a^{\mathrm{t}} - \sigma I)^{-1}$) by a vector and converge to the root ζ of the polynomial system which maximizes (resp. minimizes) the value $|a(\zeta)|$.

The usual analysis of these algorithms is immediately extended to our case. To set a framework, let us next specify the application of the power and the inverse power iterations and their convergence rate.

Let $\mathtt{M}_a$ be the matrix of multiplication by a in a basis B of $\mathscr{A}$ and let us assume that $a(\zeta_i) \neq 0$ for $i = 1, \ldots, d$. Then, by Theorem 2.1, $\mathtt{M}_a$ is invertible and a is invertible in $\mathscr{A}$. Let $\mathbf{v}_0 = \mathbf{w}_0$ be the coordinate vector of an element of $\widehat{\mathscr{A}}$ in the dual basis of B. In our case, the power method and the inverse power method amount to the inductive computation of the sequences:

$$\mathbf{w}_k = \frac{1}{\|\mathbf{w}_{k-1}\|}\mathtt{M}_a^{\mathrm{t}}\mathbf{w}_{k-1} \quad \text{and} \quad \mathbf{v}_k = \frac{1}{\|\mathbf{v}_{k-1}\|}(\mathtt{M}_a^{\mathrm{t}})^{-1}\mathbf{v}_{k-1},$$

$k = 1, 2, \ldots$, respectively.

Due to Remarks 2.4 and 2.5 and to the well-known convergence results for the power and the inverse power methods [12], we have:

Proposition 3.5. *Let $\zeta \in \mathscr{Z}(I)$ be a simple root such that*

$$|a(\zeta')| < |a(\zeta)| \quad (\textit{resp. } 0 < |a(\zeta)| < |a(\zeta')|) \quad \textit{for all } \zeta' \in \mathscr{Z}(I),\ \zeta' \neq \zeta.$$

Let

$$\rho = \max\left\{\left|\frac{a(\zeta')}{a(\zeta)}\right|,\ \zeta' \in \mathscr{Z}(I), \zeta' \neq \zeta\right\} < 1$$
$$\left(\textit{resp. } \rho = \max\left\{\left|\frac{a(\zeta)}{a(\zeta')}\right|,\ \zeta' \in \mathscr{Z}(I), \zeta' \neq \zeta\right\} < 1\right).$$

Let $\mathbf{w} = (\zeta^\alpha)_{\alpha\in E}$ (*resp.* $\mathbf{v}(\zeta^\alpha)_{\alpha\in E}$) *be the monomial basis evaluated at the root* ζ *and let*

$$\mathbf{w}^* = \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad \left(\textit{resp. } \mathbf{v}^* = \frac{\mathbf{v}}{\|\mathbf{v}\|}\right).$$

Then for the generic choice of the vector $\mathbf{v}_0$, *we have*

$$\|\mathbf{w}_k - \mathbf{w}^*\| \leqslant c\,\rho^k \quad (\textit{resp. } \|\mathbf{v}_k - \mathbf{v}^*\| \leqslant c\,\rho^k)$$

for $\mathbf{w}_k$ *and* $\mathbf{v}_k$ *defined above and for some constant* $c \in \mathbb{R}^+$.

The proposition enables us to approximate the eigenvector corresponding to a root that minimizes or maximizes the modulus of the value of $a(\mathbf{x})$ at a root. If the basis B contains $1, x_1, \ldots, x_n$, Algorithm 2.6 immediately computes the coordinates of the root ζ, from the coordinates of $\mathbf{v}^*$ or $\mathbf{w}^*$.

To compute the next root of the polynomial system, one may repeat the same process with a new polynomial $a(\mathbf{x})$, e.g., one may choose a polynomial $a(\mathbf{x}) \in \mathscr{A}$ that vanishes at all previously computed roots and then compute the root of the system that maximizes $|a(\zeta)|$.

4. Construction of the Sylvester-like matrices

In this section, we specify three constructions of matrices S satisfying Hypothesis 3.1.

4.1. Resultant matrices

The first approach is related to the resultants of $n+1$ polynomials $f_0, \ldots, f_n$ in n variables. The vanishing of the resultant over a projective variety X of these polynomials is the necessary and sufficient condition on the coefficients of the polynomials $f_0, \ldots, f_n$ to have a common root in X (see [9]). Our presentation unifies several known approaches under the same terminology of Sylvester map. In particular, we will cover the cases where $X = \mathbb{P}^n$ is the projective space of dimension n, which yields the classical resultant (see [15,28]), and where X is a toric variety, which yields the so-called toric resultant (see [2,9,26]). The resultant can be computed as a divisor of the determinant of a map, which generalizes the Sylvester map for two polynomials in one variable. Let $\mathscr{V}_0, \ldots, \mathscr{V}_n$ be the $n+1$ vector spaces generated by monomials $\mathbf{x}^{E_i} = \{\mathbf{x}^\alpha, \alpha \in E_i\}$, where E_i is the set of the exponents,

$$E_i = \{\beta_{i,1}, \beta_{i,2}, \ldots\}.$$

Let $\mathscr{V}$ be the vector space generated by all the monomials of the polynomials $f_i\,\mathbf{x}^{\beta_i}$, for $\beta_i \in E_i$. This set of monomials is denoted by $\mathbf{x}^F = (\mathbf{x}^\beta)_{\beta\in F}$. We define the following map:

$$\begin{aligned} S : \mathscr{V}_0 \times \cdots \times \mathscr{V}_n &\to \mathscr{V} \\ (q_0, \ldots, q_n) &\mapsto \sum_{i=0}^{n} f_i\, q_i. \end{aligned} \tag{4}$$

The matrix $\mathtt{S}$ of S in the monomial basis of $\mathscr{V}_0 \times \cdots \times \mathscr{V}_n$ and $\mathscr{V}$ is of the form

$$\begin{array}{c} \overbrace{\phantom{\mathbf{x}^{\beta_{0,1}} f_0}}^{\mathscr{V}_0} \qquad\qquad\qquad\qquad \overbrace{\phantom{\mathbf{x}^{\beta_{n,1}} f_n}}^{\mathscr{V}_n} \\ \mathscr{V}\left\{\begin{array}{l} \mathbf{x}^{\alpha_1} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}^{\alpha_N} \end{array}\right. \left[\begin{array}{c|c|c} \begin{array}{c}\cdot\\ \cdot\\ \mathbf{x}^{\beta_{0,1}} f_0 \\ \cdot\\ \cdot\end{array} & \cdots & \begin{array}{c}\cdot\\ \cdot\\ \mathbf{x}^{\beta_{n,1}} f_n \\ \cdot\\ \cdot\end{array} \\ & \cdots\cdots\cdots & \\ & & \end{array} \cdots\right]. \end{array}$$

It is decomposed into $\mathtt{S} = [\mathtt{S}_0, \ldots, \mathtt{S}_n]$, where $\mathtt{S}_i$ represents the monomial multiples of the polynomial f_i. The rows of this matrix are indexed by the monomials $\mathbf{x}^F$, so that Hypothesis 3.1(1) is satisfied. The columns are indexed by the monomials in $\mathbf{x}^{E_i}$, the matrix is filled with the coefficients of $f_0, \ldots, f_n$ so that the entry indexed by $\mathbf{x}^\alpha \in \mathbf{x}^F$ and $\mathbf{x}^\beta \in \mathbf{x}^{E_i}$ is filled by the coefficient of $\mathbf{x}^\alpha$ in $\mathbf{x}^\beta\, f_i$ (in particular the entry is 0 if $\mathbf{x}^\alpha$ does not belong to $\mathbf{x}^\beta\, f_i$).

In the classical case, we consider the construction due to Macaulay (see [15]). Let $d_0, \ldots, d_n$ be the degree of the polynomials $f_0, \ldots, f_n$ and let $\nu = d_0 + \cdots + d_n - n$. The set $\mathbf{x}^F$ will be the set of all monomials of degree $\leqslant \nu$ in the variables $x_1, \ldots, x_n$, and E_i will be a subset of the monomials of degree $\nu - d_i$ so that the map S is well-defined.

In the toric case, we consider the support of the polynomials f_i, that is, the set of monomials with non-zero coefficients in f_i, and we denote by C_i the convex hull of the exponents of these monomials (also called the Newton polytope of f_i). In order to construct the map S that yields the toric resultant, we fix (at random) a direction $\delta \in \mathbb{Q}^n$. For any polytope C, let C^δ denote the polytope obtained from C by removing its facets whose normals have positive inner products with δ. Taking

$$E_i = \left(\sum_{j\neq i} C_j\right)^\delta \quad \text{and} \quad F = \left(\sum_j C_j\right)^\delta$$

allows us to define the desired map S. We refer the reader to [2,9,26] for further details.

Now, let us check, step by step, that Hypothesis 3.1 are satisfied. In the experiments (cf. Section 5), we choose a linear form for f_0. Here, we only assume that f_0

contains a constant term. As all the monomials of $f_0\,\mathbf{x}^{E_0}$ are in $\mathscr{V}$, this implies that the set of the monomials $\mathbf{x}^F$ that index the rows contains the set $\mathbf{x}^{E_0}$. Therefore, we can partition the matrix $\mathtt{S}$ according to (2), so that

$$\mathtt{S}_0 = \begin{pmatrix} \mathtt{A} \\ \mathtt{C} \end{pmatrix} \quad \text{and} \quad [\mathtt{S}_1, \ldots, \mathtt{S}_n] = \begin{pmatrix} \mathtt{B} \\ \mathtt{D} \end{pmatrix}$$

and Hypotheses 3.1(3) and 3.1(4) are satisfied.

In the classical case over $\mathbb{P}^n$, the set E_0 is $E_0 = \{(a_1, \ldots, a_n);\ 0 \leqslant a_i \leqslant d_i - 1\}$. For generic polynomials $f_1, \ldots, f_n$ of degree n, this set is a basis of $\mathscr{A} = R/(f_1, \ldots, f_n)$ (see [15]). In the toric case, the set E_0 is a set of points in the mixed cell of a subdivision of the (Minkowski) sum of the polytopes $C_1, \ldots, C_n$. For generic polynomials $f_1, \ldots, f_n$ with support in $C_1, \ldots, C_n$, this is a monomial basis of $\mathscr{A} = R/(f_1, \ldots, f_n)$ (see [5,23]), and Hypothesis 3.1(2) is also satisfied.

To check if Hypothesis 3.1(5) holds, it is possible to specialize the coefficients of the polynomials $f_1, \ldots, f_n$ in such a way that the matrix $\mathtt{D}$ has a dominant diagonal. Thus the determinant of $\mathtt{D}$, as a polynomial in the coefficients of $f_1, \ldots, f_n$, is not identically zero. Consequently, it is not zero for generic values of these coefficients.

Since Hypothesis 3.1 are satisfied, we can apply the forward or implicit inverse power iteration, for generic systems of equations of fixed degree or fixed support. These resultant constructions take into account only the monomial structure of the input polynomials, but not the values of their coefficients. It may happen, of course, that for specific values of these coefficients, the matrix $\mathtt{D}$ would become singular. In this case, we may use the construction described in Section 4.3, requiring a little higher computational cost.

4.2. Overconstrained systems

The method for constructing $\mathtt{S}$ admits a natural generalization to overconstrained but consistent polynomial systems, that is, to the systems of equations $f_1 = 0, \ldots, f_m = 0$, with $m > n$, defining a finite number of roots. We obtain a substantial simplification in the cases where such a system has only one or only a few roots (or pseudo roots, see below). We still consider a map of the form

$$\begin{aligned} S : \mathscr{V}_0 \times \cdots \times \mathscr{V}_m &\to \mathscr{V} \\ (q_0, \ldots, q_m) &\mapsto \sum_{i=0}^{m} f_i\, q_i, \end{aligned}$$

such that the matrix of this map satisfies Hypothesis 3.1. Such a map can be constructed by using the techniques of the previous section and by adding new columns corresponding to the multiples of the polynomials $f_{n+1}, \ldots, f_m$. This yields a rectangular matrix $\tilde{\mathtt{S}}_1$, from which we extract a submatrix $\mathtt{R}_1$, having as many rows, and whose number of columns is exactly its rank. Let L be the list of polynomials corresponding to these columns. Let us next choose a minimal cardinality subset $E_0 \subset F$

of the exponents of the monomials such that $\langle \mathbf{x}^F \rangle = \langle \mathbf{x}^{E_0} \rangle \oplus \langle L \rangle$, (cf. [10,14]). This yields a square matrix $\mathtt{S}$, which satisfies Hypothesis 3.1.

A case of special interest is the case where $\mathscr{A}$ is of dimension 1, so that there is only one simple root, $\chi = (\chi_1, \ldots, \chi_n)$. A basis of $\mathscr{A}$ is 1, and the matrix of multiplication by x_i is $[\chi_i]$. Then for any matrix $\mathtt{S}$ satisfying Hypothesis 3.1 with $f_0 = x_i$, $\mathtt{A}$ is a one-by-one matrix and we have $[\chi_i]\mathtt{A} - \mathtt{B}\,\mathtt{D}^{-1}\mathtt{C}$. In this case, only one solution of a linear system is required.

This occurs, for instance, in problems of reconstruction in Computer Vision, where any pair of points, in correspondence to the images, gives a polynomial equation (see [8]). This is also the case for kinematic problems where more sensors than needed are used, and in computational biology where the distances from an atom to more than three other atoms are known. Furthermore, due to truncation and round-off errors of the coefficients of the input polynomials, they define an overconstrained system, which has no roots, but only pseudo roots, at which the values of $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ are not equal to but close to zero. Even in this case, our techniques yield an approximation to the solution of the exact equations.

4.3. Computing Sylvester matrices by using Gröbner basis

In this section, we assume that a reduced Gröbner basis $(g_1, \ldots, g_s)$ of I, for some monomial order, refining the degree order, is available. For any $p \in R$, let $\mathscr{L}(p)$ be its leading monomial. We also assume that we know a decomposition of each g_i in terms of the input polynomials

$$g_i = \lambda_{i,1} m_{i,1} f_{i,1} + \lambda_{i,2} m_{i,2} f_{i,2} + \cdots + \lambda_{i,k_i} m_{i,k_i} f_{i,k_i 1},$$

where $\lambda_{i,j} \in \mathbb{C}$, $f_{i,j} \in \{f_1, \ldots, f_m\}$ and $m_{i,j}$ is a monomial of R. We order these terms in such a way that $\mathscr{L}(m_{i,j} f_{i,j}) \geqslant \mathscr{L}(m_{i,j+1} f_{i,j+1})$.

Let us denote by $B_0 = \mathbf{x}^{E_0} = (\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_D})$ the set of all monomials that are not in the ideal generated by $(\mathscr{L}(g_1), \ldots, \mathscr{L}(g_s))$. This set is a basis of $\mathscr{A} = R/(f_1, \ldots, f_m) = R/I$ (see [4]) and contains 1 if $\mathscr{L}(I) \neq \emptyset$.

We describe how to construct a Sylvester-type matrix $\mathtt{S}$, satisfying Hypothesis 3.1, with $f_0 = u_0 + u_1 x_1 + \cdots + u_n x_n$. The set of monomials F and a list of multiples of the polynomials $f_1, \ldots, f_m$ will be defined by induction as follows:

Let $F_0 = B_0$, $L_0 = \emptyset$ and let $F_1 = F_0 \cup x_1 F_0 \cup \cdots \cup x_n F_0$, $L_1 = \emptyset$. Assume that $F_0, \ldots, F_n$ have been defined and note that they contain B_0. Then any monomial $\mathbf{x}^\alpha$ in $F_n - F_{n-1}$ is a multiple of the initial monomial $\mathscr{L}(g_{c(\alpha)})$ of $g_{c(\alpha)}$ for some $c(\alpha) \in \{1, \ldots, m\}$: $\mathbf{x}^\alpha = n_\alpha \mathscr{L}(g_{c(\alpha)})$. Let

$$L_{n+1,\alpha} = \{n_\alpha m_{c(\alpha),j} f_{c(\alpha),j};\ j = 1, \ldots, k_{c(\alpha)}\}$$

and let $F_{n+1,\alpha}$ be the set of all monomials of the polynomials of this set. Then we define

$$F_{n+1} = \bigcup_{\alpha \in F_n - F_{n-1}} F_{n+1,\alpha} \cup F_n,$$

$$L_{n+1} = \bigcup_{\alpha \in F_n - F_{n-1}} L_{n+1,\alpha} \cup L_n.$$

Lemma 4.1. *There exists some* $K \geqslant 1$ *such that* $\forall n \geqslant K,\ F_n = F_K$.

Proof. By construction, for all n in $\mathbb{N}$, the set of monomials F_n is included into the set of monomials, which precedes the monomials $x_i m_{j,1}$ (and $m_{j,1}$), for $i = 1, \ldots, n$ and $j = 1, \ldots, s$, according to the fixed ordering. By the hypotheses about the monomial ordering, this set is finite, so that the increasing sequence F_n is stationary, for $n \geqslant K$. □

By construction, any polynomial in $L := L_K$ can be decomposed as a linear combination of the monomials in $F := F_K$. Let $\tilde{\mathtt{S}}_1$ be the coefficient matrix of the polynomials in L, in this monomial basis $\mathbf{x}^F$.

By definition, any monomial of $\mathbf{x}^{F_{n+1}-F_n}$ can be reduced by monomial multiples of the polynomials $g_1, \ldots, g_s$ (that is, by linear combinations of the polynomials in L_{n+1}) to a linear combination of monomials in $\mathbf{x}^{F_n}$. By induction, this shows that any monomial in F can be reduced modulo the polynomials of L to a linear combination of monomials in B_0. In other words, $\langle \mathbf{x}^F \rangle = \langle B_0 \rangle \oplus \langle L \rangle$.

If we divide the matrix $\tilde{\mathtt{S}}_1$ into blocks as $\tilde{\mathtt{S}}_1 = \binom{R_1}{R_1'}$, according to whether the rows are indexed by the monomials in B_0 or not, the decomposition $\langle \mathbf{x}^F \rangle = \langle B_0 \rangle \oplus \langle L \rangle$ implies that R_1' is of maximal rank. Let $\mathtt{S}_1$ be the submatrix of $\tilde{\mathtt{S}}$ such that the corresponding submatrix of $\mathtt{R}_1'$ is invertible. It is of the form $\mathtt{S}_1 = \binom{\mathtt{B}}{\mathtt{D}}$ with $\mathtt{D}$ invertible.

Let $\mathtt{S}_0$ be the coefficient matrix of the polynomials $(f_0 \mathbf{x}^\alpha)_{\alpha \in E_0}$ in the monomial basis F and let $\mathtt{S} = [\mathtt{S}_0, \mathtt{S}_1]$.

We easily check that Hypothesis 3.1 are satisfied.

This method is most interesting when we have to solve a polynomial system depending on parameters, for various values of these parameters. The classical Gröbner approach requires to recompute a Gröbner basis for each value of these parameters. Moreover, it cannot be applied safely with floating point coefficients. With the approach we propose, it is sufficient to compute numerically a single Gröbner basis, and the matrix $\mathtt{S}$ is used for the other values of the parameters, assuming that the geometric properties of these systems do not change. Let us summarize our procedure.

Algorithm 4.2. Solution of a parameterized polynomial system under the variation of the parameters.

1. Over a prime field $\mathbb{Z}_p$ for a fixed prime p, compute a Gröbner basis of the given polynomial system, for rational values of the parameters.
2. Construct the matrix $\mathtt{S}$ associated to the input system (and depending on the parameters).
3. Substitute the value of the parameters in $\mathtt{S}$.

Table 1

	N	S	D	n	k	T (s)
s44	36	138	16	2	7	0.050
s442	165	821	32	3	6	0.151
s4422	715	3704	64	4	8	1.179
s455	364	1664	100	3	6	2.331
s2445	1820	8795	160	4	8	4.323
s22445	8568	41942	320	5	8	28.213
sq4	126	585	16	4	5	0.313
sq5	462	2175	32	5	44	2.135
sq6	1716	7973	64	6	52	49.397
sing	210	4998	21	2	14	0.438
kruppa	792	15822	1	5	1	0.698

4. If the matrix D is singular, then stop. Otherwise approximate the eigenvectors of the matrix $\mathtt{M}_a^{\mathtt{t}}$ defined by (2) and (3) (cf. Section 3).

All steps of this algorithm can be applied by using either modular or floating point arithmetic. Here again, the matrix S is structured and sparse, so that the eigenvectors of $\mathtt{M}_a^{\mathtt{t}}$ can be computed efficiently.

5. Experimental tests

We report the initial results of our experimentation with the implicit shifted inverse power method, applied for computing a selected root of a polynomial system. In our experiments we defined the shifts dynamically, as the iteration converged to a root. For solving the sparse linear system $\mathtt{S}\,x = b$, we used the library TNT[3] developed by R. Pozo; more precisely, we used the GMRES solver with an ILU-preconditionner (see [24] for more details on these solvers). The matrices are generated by the C++ library ALP[4], which implements Macaulay's construction of resultant matrices. We plan to perform similar experiments based on the implementation of toric resultant matrices by Emiris [2]. The results of the experiments are shown in Table 1.

In Table 1, N is the dimension of the matrix S (that is, the matrix has size $N \times N$), S the number of non-zero entries of the matrix S, D the dimension of $\mathscr{A}$, n the number of variables, k the number of iterations required for an error less than $\epsilon = 10^{-4}$, and T is the total time of the computation. This time is the "user" time, obtained by

[3] See http://math.nist.gov/tnt/.

[4] See http://www.inria.fr/saga/logiciels/ALP/.

the unix command `time`. This experimentation has been carried out on a Dec Alpha 500 AU workstation with 512M of local memory.

The examples `s44`, ..., `s22445` are examples with a few monomials, where Macaulay construction can be applied. The number of solutions is the product of the degree. The first example is a system of two equations in two variables, both of degree 4, the second is a system of three equations in three variables of degree 2, 4, 4, and so on.

The examples `sq4`, ..., `sq6` correspond to the intersection of quadrics in a space of dimension 4, 5, 6, with no point at infinity (this problem came from Signal Processing).

The example `sing`, corresponds to the singular points of the plane curve defined by

$$\begin{aligned} p := {} & x^8 - 8x^7y + 28x^6y^2 - 56x^5y^3 + 70x^4y^4 - 56x^3y^5 \\ & + 28x^2y^6 - 8xy^7 + y^8 - 128x^7 + 448x^6y - 672x^5y^2 \\ & + 560x^4y^3 - 280x^3y^4 + 84x^2y^5 - 14xy^6 + y^7 - 8x^6 \\ & + 48x^5y - 120x^4y^2 + 160x^3y^3 - 120x^2y^4 + 48xy^5 - 8y^6 \\ & + 224x^5 - 560x^4y + 560x^3y^2 - 280x^2y^3 + 70xy^4 - 7y^5 \\ & + 20x^4 - 80x^3y + 120x^2y^2 - 80xy^3 + 20y^4 - 112x^3 \\ & + 168x^2y - 84xy^2 + 14y^3 - 16x^2 + 32xy - 16y^2 + 14x \\ & - 7y + 2 \end{aligned}$$

(see [3]). Such singular points are defined by $p = 0, d_x(p) = 0, d_y(p) = 0$. This leads to an overconstrained system whose associated matrix `S` is of size 210. We construct this matrix from the Macaulay matrix of $p, d_x(p), d_y(p) + d_x(p)$ (which is of rank 189), by replacing the first $210 - 189 = 21$ columns by multiples of the linear form $x - 4$. Here is a picture of the structure of the Macaulay matrix, the non-zero entries represented by a point (see Fig. 1).

Though the polynomial p has many monomials, only 11% of the coefficients of the matrix `S` are not zero. There are 21 singular points on this curve (which are all

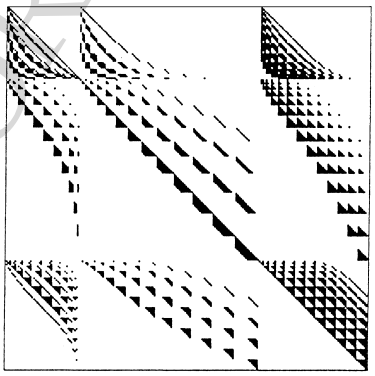

Fig. 1. The Macaulay matrix of $p, d_x(p), d_y(p)$.

real), and by this method we are able to select the point whose first coordinate is the nearest to 4. Note that the matrix of multiplication by this linear form in $\mathscr{A}$ can be computed by solving 21 systems associated to the matrix `S`.

The system `kruppa` corresponds to the Kruppa equations of a reconstruction problem in Computational Vision (see [8]) reduced to an overconstrained system of six quadrics in a space of dimension 5. We construct the Macaulay matrix associated to these six equations and replace its first column by a multiple of a linear form. By solving one system of the form $\mathtt{S}\,\mathbf{x} = \mathbf{b}$, we obtain one coordinate of the solution. The time needed to compute this coordinate is reported in the table.

References

[1] W. Auzinger, H.J. Stetter, An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations, in: Proceedings of the International Conference on Numerical Mathematics, Int. Series of Numerical Math, vol. 86, Birkhäuser, Basel, 1988, pp. 12–30.
[2] J. Canny, I. Emiris, An efficient algorithm for the sparse mixed resultant, in: G. Cohen, T. Mora, O. Moreno (Eds.), Proceedings of the International Symposium on Applied Algebra, Algebraic Algorithm and Error-Correction Codes (Puerto Rico), Lecture Notes in Computer Science, vol. 263, Springer, Berlin, 1993, pp. 89–104.
[3] R.M. Corless, P.M. Gianni, B.M. Trager, A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots, in: Proceedings of the ACM International Symposium on Symbolic and Algebraic Computation, ACM Press, New York, 1997, pp. 133–140.
[4] D. Cox, J. Little, D.O'Shea, Ideals, Varieties, and Algorithms, Undergraduate Texts in Mathematics, Springer, New York, 1992.
[5] I. Emiris, A. Rege, Monomial bases and polynomial system solving, in: Proceedings of the ACM International Symposium on Symbolic and Algebraic Computation, ACM Press, New York, 1994, pp. 114–122.
[6] I.Z. Emiris, V.Y. Pan, The structure of sparse resultant matrices, in: Proceedings of the ACM International Symposium on Symbolic and Algebraic Computer (ISSAC), ACM Press, New York, 1997, pp. 189–196.
[7] J. Emsalem, Géométrie des points épais. Bull. Soc. Math. France 106 (1978) 399–416.
[8] O. Faugeras, Three-dimensional Computer Vision: A Geometric Viewpoint, vol. XXXII, MIT, Cambridge, MA, 1993.
[9] I.M. Gelfand, M.M. Kapranov, A.V. Zelevinsky, Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Basel, 1994.
[10] P.M. Gianni, B.M. Trager, Computations with approximate ideals, Preprint, SNAP'96, 1997.
[11] M. Giusti, J. Heintz, La détermination des points isolés et de la dimension d'une variété algebrique peut se faire en temps polynomial, in: Proceedings of the International Meeting on Commutative Algebra, Symposium Mathematica, vol. XXXIV, Cortona, 1991, pp. 216–255.
[12] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., John Hopkins University Press, Baltimore, MD, 1996.
[13] A. Greenbaum, Iterative Methods for Solving Linear Systems, SIAM, Philadelphia, PA, 1997.
[14] D. Lazard, Résolution des systèmes d'équations algébriques, Theoret. Comp. Sci. 15 (1981) 77–110.
[15] F.S. Macaulay, Some formulae in elimination, Proc. London Math. Soc. 1 (33) (1902) 3–27.
[16] D. Manocha, J. Canny, Multipolynomial resultant algorithms, J. Symbolic Comput. 15 (2) (1902) 99–122.

[17] M.G Marinari, T. Mora, H.M. Möller, Grobner duality and multiplicities in polynomial system solving, in: A.H.M. Levelt (Ed.), Proceedings of the ACM International Symposium on Symbolic and Algebraic Computation, ACM Press, New York, 1995, pp. 167–179.
[18] B. Mourrain, Isolated points, duality and residues, J. Pure Appl. Algebra, 117&118 (1996) 469–493; Special issue for the Proceedings of the Fourth International Symposium on Effective Methods in Algebraic Geometry (MEGA).
[19] B. Mourrain, Computing isolated polynomial roots by matrix methods, J. Symbolic Comput. Special Issue on Symbolic-Numeric Algebra for Polynomials (1998) 715–738.
[20] B. Mourrain, V.Y. Pan, Multidimensional structured matrices and polynomial systems, Calcolo, Special Issue, workshop on Toeplitz matrices: Structure, Algorithms and Applications 33 (1997) 389–401.
[21] B. Mourrain, V.Y. Pan, Solving special polynomial systems by using structured matrices and algebraic residues, in: F. Cucker, M. Shub (Eds.), Proceedings of the workshop on Foundations of Computational Mathematics (Rio de Janeiro, 1997), Springer, Berlin, 1997, pp. 287–304.
[22] B. Mourrain, V.Y. Pan, Multivariate polynomials, duality and structured matrices, J. Complexity 16 (1) (2000) 110–180.
[23] P.S. Pedersen, B. Sturmfels, Mixed monomial basis, in: Effective Methods in Algebraic Geometry (MEGA'94), Progress in Math., vol. 143, Santander (Spain), 1994, Birkhäuser, Basel, pp. 285–306.
[24] Y. Saad, Iterative methods for sparse linear systems, PWS series in computer science, PWS, 1996.
[25] H.J. Stetter, Eigenproblems are at the heart of polynomial system solving, ACM SIGSAM Bulletin 30 (4) (1996) 22–25.
[26] B. Sturmfels, Sparse elimination theory, in: D. Eisenbud, L. Robianno (Eds.), Computational Algebraic Geometry and Commutative Algebra, Cambridge University Press, Cambridge (Proc. Cortona, June 1991), 1993, pp. 264–298.
[27] E.E. Tyrtyshnikov, A unfying approach to some old and new theorems on distribution and clustering, Linear Algebra Appl. 232 (1996) 1–43.
[28] B.L. Van der Waerden, Modern algebra, vol. II, Frederick Ungar, 1948.
[29] W.V. Vasconcelos, Computational Methods in Commutative Algebra and Algebraic Geometry, Algorithms and Computation in Mathematics, vol. 2, Springer, Berlin, 1998.
[30] J. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, London, 1965.