Controlled iterative methods for solving polynomial systems

Didier Bondyfalat* Bernard Mourrain* Victor Y. Pan†

INRIA, SAGA,
2004 route des Lucioles, B.P. 93,
06902 Sophia Antipolis, France
{dbondy,mourrain}@sophia.inria.fr
http://www.inria.fr/saga/{dbondy,mourrain}

Dept. of Math. and Computer Science
Lehman College, City Univ. of New York,
Bronx, NY 10468, USA,
VPAN@LCVAX.LEHMAN.CUNY.EDU

Abstract

For a system of polynomial equations, we seek its specified root, maximizing or minimizing the absolute value of a fixed polynomial over all roots of the system. The latter requirement to a root, complicating the already difficult classical problem, is motivated by several practical applications. We first reduce the solution to the computation of an eigenvector of an associated matrix. Our novel treatment of this rather customary stage enables us to apply it uniformly to several resultant constructions and to simplify substantially the solution of an overconstrained polynomial system having only a simple root or a few roots. Likewise, where the reduction of a general polynomial system to an eigenproblem relies on the Gröbner basis techniques, we also obtain substantial simplification. Then we elaborate application of the power method and the (shifted) inverse power method to the solution of the resulting eigenproblem. Our elaboration is not straightforward since we achieve the computation preserving the sparsity and the structure of the associated matrix involved. This enables the decrease of the arithmetic cost by roughly factor N, denoting the dimension of the associated resultant matrix. Furthermore, our experiments show that our computations can be performed numerically, with single or double precision arithmetic, and the iteration converged to a specified root quite fast.

1 Introduction

Many problems in Robotics, Computer Vision, Computational Geometry, Signal Processing, ...involve the solution of polynomial systems of equations. Such polynomial systems are usually defined by a few monomials but may be very 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. [16], [15], [1], [18], [13], [5]), many theoretical and practical problems of the solution of polynomial systems remain largely open. Particular difficulties arise in the case where one needs to select and to compute a specific root of a polynomial system. This is often the case for many practical problems, where the only interesting root is the root that maximizes or minimizes the absolute value of a given functional.

A major representative of a few available approaches is the Gröbner basis method. Its well known weakness is its adherence to modular computations. Indeed, it cannot be applied safely with floating point arithmetic and actually requires to increase the precision of computations dramatically, relative to the input and output precision. The method has also a high arithmetic computational cost. Moreover, even when the method is applied to find a single root, most part of its computation applies to all roots, thus increasing the overall arithmetic cost by a large extra factor of at least D, for D denoting the total number of all roots of a given system. Other known methods have some other deficiencies. For instance, in some cases Newton-type iterative methods may converge quickly to a root but provide no mean to guarantee convergence to a selected root.

In this paper, we propose a new iterative method for computing a selected root of a polynomial system, which extends previous works on the power method (see e.g. [14], [17]) to improve computations with structured matrices associated with multivariate polynomial systems. This method can be implemented with floating point arithmetic, and its 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 which converges linearly to the solution and uses $\mathcal{O}^*(N^2)$ arithmetic operations in each recursive step, whereas the known approaches use order of N^3 operations. Technically, we achieve this by relying on *numerical linear algebra* and exploiting the *structure and sparsity of matrices*, such as resultant matrices, involved in our solution of polynomial systems.

We start with a known reduction of a polynomial system to linear algebra computations, namely, to a matrix eigenproblem. We contribute to this well known and well studied topic by showing 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).

Furthermore, our techniques enable us to achieve substantial additional improvement of the solution in 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 com-

*Partially supported by European ESPRIT project FRISCO, LTR 21.024

†Supported by NSF Grant CCR 9625344 and PSC CUNY Award 668365

putation towards the approximation of only a specified root and to preserve the matrix structure and sparsity. In both overconstrained and Gröbner basis cases, we substantially decrease the computational cost, roughly by factor D (the overall number of roots), versus the known algorithms.

Furthermore, our algorithms rely on a novel observation (which becomes quite simple under our mapping/operator approach to the problem) that the application of some customary matrix computations (namely, of the power method and the shifted inverse power method) can be elaborated to compute a specific solution to the system, that is, a root maximizing or minimizing the absolute value of a fixed polynomial $f_0(x)$. Such an application of the power method and the (shifted) inverse power method to our problem is not straightforward, particularly because we care about preserving the structure and sparsity of matrices involved. We elaborate this application in sections 2 and 3. In section 4, we specify three approaches to the construction of these structured matrices and give a demonstration for a parameterized polynomial system, where we compute a single Sylvester-like matrix, for all parameters, unlike the usual application of Gröbner basis method, which recomputes such a matrix for every parameter.

Some other papers [4], [24], [23], [7], [25] also exploited the reduction of the problem to polynomial multiplication, although by different means and without the techniques cited above, which enabled our computation of the extremal roots (maximizing a chosen functional) and our application to the solution of overconstrained systems.

The 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 sufficiently fast converge to a specified root minimizing or maximizing a fixed polynomial. We intend to continue our experimentation to clarify numerical behavior of the algorithms.

We believe that our algorithms have good potentials to became practical and should allow various further improvements, for instance, by using parallel processing.

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 matrix eigenproblem (cf. [1], [29], [8], [21], [22]). 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 $\mathbb{K}$.

To motivate and illustrate the material of this section, we first consider the univariate case, where we have a polynomial $f \in \mathbb{C}[x]$ of degree d with d simple roots: $f(x) = f_d \prod_{i=1}^{d}(x - \zeta_i)$. The quotient algebra of residue polynomials modulo f, denoted by $\mathcal{A} = \mathbb{C}[x]/(f)$, is a vector space of dimension d. Its basis is $(1, x, \ldots, x^{d-1})$. Consider the Lagrange polynomials

$$\mathbf{e}_i = \prod_{j \neq i} \frac{x - \zeta_i}{\zeta_j - \zeta_i}.$$

Verify immediately that $\sum_i \mathbf{e}_i = 1$ and $\mathbf{e}_i \mathbf{e}_j \equiv \mathbf{e}_i(\mathbf{e}_i - 1) \equiv 0$ (for these two polynomials vanish at the roots of f). In other words, the Lagrange polynomials $\mathbf{e}_i$ are orthogonal idempotents in $\mathcal{A}$ and we have $\mathcal{A} = \sum_i \mathbb{C}\, \mathbf{e}_i$. Moreover, for any polynomials $a \in \mathcal{A}$, we also have $(a - a(\zeta_i))\mathbf{e}_i \equiv 0$, so that $\mathbf{e}_i$ is an eigenvector for the operator of multiplication by a in $\mathcal{A}$, for the eigenvalue $a(\zeta_i)$. These multiplication operators have a diagonal form in the basis $(\mathbf{e}_i)$ of $\mathcal{A}$. A basic property of Lagrange polynomials implies that for any $a \in \mathcal{A}$, we have $a \equiv \sum_i a(\zeta_i)\, \mathbf{e}_i(x)$. Therefore, the dual basis of $(\mathbf{e}_i)$ (formed by the coefficients of $\mathbf{e}_i$ in this decomposition) consists of the linear forms associating to a its values at the points ζ_i. We will extend this approach to the case of multivariate polynomial systems, which of course will require substantial further elaboration and algebraic formalism. We refer to [20], [21], [22], [29] for further details.

Let $f_1, \ldots, f_m$ be m polynomials of R, defining the polynomial system $f_1(x) = 0, \ldots, f_m(x) = 0$. Let I be the ideal generated by these polynomials. We consider the case, where the quotient algegra $\mathcal{A} = R/I$ *is of finite dimension* D *over* $\mathbb{C}$. This implies that the set of roots or solutions $\mathcal{Z}(I) = \{\zeta \in \mathbb{C}^n; f_1(\zeta) = \ldots = f_m(\zeta) = 0\}$ is finite: $\mathcal{Z}(I) = \{\zeta_1, \ldots, \zeta_d\}$ with $d \leq D$. To this set of roots, we associate a fundamental set of orthogonal idempotents $\mathbf{e}_1, \ldots, \mathbf{e}_d$ satisfying

$$\mathbf{e}_1 + \cdots + \mathbf{e}_d \equiv 1, \text{ and } \mathbf{e}_i\, \mathbf{e}_j \equiv \left\{ \begin{array}{l} 0 \text{ if } i \neq j, \\ \mathbf{e}_i \text{ if } i = j, \end{array} \right.$$

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

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 $\widehat{\mathcal{A}}$ the dual space of $\mathcal{A}$, that is, the set of elements $\Lambda \in \widehat{R}$ such that $\Lambda(I) = 0$ (also denoted by $I^{\perp}$).

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

$$\begin{array}{rcl} M_a : \mathcal{A} & \rightarrow & \mathcal{A} \\ b & \mapsto & a\, b \end{array}$$

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

$$\begin{array}{rcl} M_a^{\mathsf{t}} : \widehat{\mathcal{A}} & \rightarrow & \widehat{\mathcal{A}} \\ \Lambda & \mapsto & a \cdot \Lambda \end{array}$$

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

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

$$\mathtt{M}_a = \left(\begin{array}{ccc} \mathtt{M}_{a,1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathtt{M}_{a,d} \end{array} \right)$$

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

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

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

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 one.

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

$$\begin{array}{rcl} \mathbf{1}_\zeta : R & \rightarrow & \mathbb{C} \\ p & \mapsto & p(\zeta) \end{array}$$

and note that $\mathbf{1}_\zeta \in \widehat{\mathcal{A}}$ if and only if $\zeta \in \mathcal{Z}(I)$. For the systems of polynomial equations having multiple roots, we introduce additional techniques involving higher order differential forms (the reader may first examine these techniques in the univariate case of a single equation). Specifically, we also consider the map (linear form)

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

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 denote this linear form $\mathbf{d}_\zeta^{\mathbf{a}} = (\mathbf{d}_{1,\zeta})^{a_1} \cdots (\mathbf{d}_{n,\zeta})^{a_n}$.

We remark that for any pair of $a, b \in \mathcal{A}$, 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 multiple roots, a complete description of the eigenspace involves the maps $\mathbf{d}_{\zeta_i}^{\mathbf{a}}$. See [9], [19], [20], ...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 $a, b \in \mathcal{A}$, 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 [21]):

Proposition 2.3 *The common eigenvectors of M_a^t for all $a \in \mathcal{A}$ 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).*

Remark 2.5 *If $(\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_D})$ is a basis of $\mathcal{A}$, then the coordinates of $\mathbf{1}_{\zeta_i}$ in its dual basis are $(\zeta_i^{\alpha_1}, \ldots, \zeta_i^{\alpha_D})$, by 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 of the matrix of multiplication by $a \in \mathcal{A}$ in a basis of the form $(1, x_1, \ldots, x_n, \ldots)$.*

2. *Compute its eigenvectors $\mathbf{v}_i = (v_{i,1}, v_{i,x_1}, \ldots, v_{i,x_n}, \ldots)$ for $i = 1, \ldots, d$.*

3. *For $i = 1, \ldots, d$, compute and output*

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

3 Controlled iterative methods for solving polynomial systems

3.1 The classical power method and inverse power methods

Algorithm 2.6 reduces the solution of a polynomial system to eigenvector computations, and for the latter task we will adjust the classical (inverse) power method (cf. [14]).

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

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

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

Due to remarks 2.4, 2.5 and to the well known convergence results for the (inverse) power method [14], we have the following proposition:

Proposition 3.1 *Let $\zeta \in \mathcal{Z}(I)$ be a simple root such that $|a(\zeta')| < |a(\zeta)|$ (resp. $0 < |a(\zeta)| < |a(\zeta')|$) for all $\zeta' \in \mathcal{Z}(I)$, $\zeta' \neq \zeta$. Let $\rho = max\{|\frac{a(\zeta')}{a(\zeta)}|,\ \zeta' \in \mathcal{Z}(I), \zeta' \neq \zeta\} < 1$ (resp. $\rho = max\{|\frac{a(\zeta)}{a(\zeta')}|,\ \zeta' \in \mathcal{Z}(I), \zeta' \neq \zeta\} < 1$). 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 $\mathbf{w}^* = \frac{\mathbf{w}}{\|\mathbf{w}\|}$ (resp. $\mathbf{v}^* = \frac{\mathbf{v}}{\|\mathbf{v}\|}$). Then for the generic choice of the vector $\mathbf{v}_0$, we have*

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

for some constant $c \in \mathbb{R}^+$.

The proposition provides a way to selecting an eigenvector corresponding to the roots which minimize or maximize the modulus of a.

If 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 $\mathbf{w}^*$.

If there are k roots $\zeta_1, \ldots, \zeta_k$ such that $|a(\zeta_1)| = \cdots = |a(\zeta_k)|$ is the minimum value of $|a|$ on $\mathcal{Z}(I)$, it is also possible to recover the eigenvectors corresponding to $a(\zeta_1), \ldots, a(\zeta_k)$ from the successive vectors $\mathbf{v}_n, \ldots, \mathbf{v}_{n+k-1}$ (see [14]).

If the root ζ is a multiple root, we may also apply recursive multiplication of a random vector by the matrix $\mathtt{M}_a$ or $\mathtt{M}_a^{\mathrm{t}}$. In this case the recursive process also converges probabilistically to an eigenvector of $\mathtt{M}_a$ or $\mathtt{M}_a^{\mathrm{t}}$, albeit more slowly than in the case of simple eigenvalues, and the coordinates of the roots can be computed from the eigenvalues.

3.2 Implicit inverse power method

Usually, the multiplication map is not available directly from the input equations. However, in many interesting cases, the matrix of this map can be recovered from Sylvester-like matrices $\mathtt{S}$, representing multiples of these input equations. We will see in the next section, how such matrices can be built. In order to apply the method that we propose, we will construct matrices $\mathtt{S}$ with the following properties:

Hypotheses 3.2 *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* $(\mathbf{x}^{\alpha})_{\alpha \in F}$,
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* $\mathcal{A} = R/(f_1, \ldots, f_m)$,
3. *the columns of* $\begin{pmatrix} \mathtt{A} \\ \mathtt{C} \end{pmatrix}$ *represent the elements* $\mathbf{x}^{\alpha} f_0$ *for* $\alpha \in E_0$, *expressed as linear combinations of the monomials* $(\mathbf{x}^{\beta})_{\beta \in F}$,
4. *the columns of* $\begin{pmatrix} \mathtt{B} \\ \mathtt{D} \end{pmatrix}$ *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. [8], [23]):

Proposition 3.3 *Under hypothesis 3.2, the matrix of multiplication by* f_0 *in the basis* $B_0 = (\mathbf{x}^{\alpha})_{\alpha \in E_0}$ *of* $\mathcal{A} = R/(f_1, \ldots, f_m)$ *is the Schur complement of* $\mathtt{D}$ *in* $\mathtt{S}$*:*

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

Based on proposition 2.4, we may apply the power method to compute the root $\zeta \in \mathcal{Z}(I)$, for which $|f_0(\zeta)|$ is maximum:

Algorithm 3.4 THE POWER METHOD FOR THE SCHUR COMPLEMENT.

1. *Choose a random vector* $\mathbf{u}_0$.
2. *For* $n \geq 1$,
   - *compute* $\mathbf{v}_{n+1} = \mathtt{B}^t \mathbf{u}_n$,
   - *solve* $\mathtt{D}^t \mathbf{w}_{n+1} = \mathbf{v}_{n+1}$,
   - *compute* $\mathbf{r}_{n+1} = \mathtt{A}^t \mathbf{u}_{n+1} - \mathtt{C}^t \mathbf{w}_{n+1}$,
3. *Let* $\mathbf{r}_{n+1,1}$ *be the first coordinate of* $\mathbf{r}_{n+1}$. *If it is not zero, then compute* $\mathbf{u}_{n+1} = \frac{1}{\mathbf{r}_{n+1,1}} \mathbf{r}_{n+1}$, *replace* n *by* $n+1$ *and go to stage* 2, *else stop.*

Due to proposition 3.3, we obtain the following estimate.

Proposition 3.5 *Assume that* $\zeta \in \mathcal{Z}(I)$ *is a simple root such that* $|f_0(\zeta')| < |f_0(\zeta)|$ *for all* $\zeta' \in \mathcal{Z}(I)$, $\zeta' \neq \zeta$. *Moreover, assume that the first row of* $\mathtt{S}$ *is indexed by* 1. *Let* $\sigma_+ = (\zeta^{\alpha})_{\alpha \in E_0}$ *and* $\rho_+ = max\{|\frac{f_0(\zeta')}{f_0(\zeta)}|; \zeta' \in \mathcal{Z}(I), \zeta' \neq \zeta\} < 1$. *Then we have* $\|\mathbf{u}_{n+1} - \sigma_+\| \leq c_+\, \rho_+^n$, *for some constant* $c_+ \in \mathbb{R}^+$.

Next, we will show how to apply the inverse power method to compute a root of our system that minimizes $|f_0|$. A known advantage versus the power method is the possible use of shifts of the variable (that is the transformation from the matrix $\mathtt{M}_a^{-1}$ to the matrix $(\mathtt{M}_a - \sigma I)^{-1}$, where σ is close to the selected eigenvalue of $\mathtt{M}_a$) for the convergence acceleration.

Proposition 3.6 *Under the hypothesis 3.2, if* $\mathtt{S}$ *of (2) is invertible and* $\mathtt{S}^{-1}$ *has block decomposition of the form*

$$\mathtt{S}^{-1} = \begin{pmatrix} \mathtt{U} & \mathtt{V} \\ \mathtt{Z} & \mathtt{W} \end{pmatrix}$$

(cf. (2)), then $\mathtt{M}_{f_0}$ *is invertible and* $\mathtt{U} = \mathtt{M}_{f_0}^{-1}$.

Proof. Under the hypothesis 3.2 and according to the decomposition

$$\begin{pmatrix} \mathtt{A} & \mathtt{B} \\ \mathtt{C} & \mathtt{D} \end{pmatrix} \begin{pmatrix} \mathbb{I} & \mathbf{0} \\ -\mathtt{D}^{-1}\mathtt{C} & \mathbb{I} \end{pmatrix} = \begin{pmatrix} \mathtt{A} - \mathtt{B}\,\mathtt{D}^{-1}\,\mathtt{C} & \mathtt{B} \\ \mathbf{0} & \mathtt{D} \end{pmatrix},$$

$\mathtt{S}$ is invertible if and only if $\mathtt{M}_{f_0} = \mathtt{A} - \mathtt{B}\,\mathtt{D}^{-1}\,\mathtt{C}$ is invertible, and then we have

$$\begin{aligned} \mathtt{S}^{-1} &= \begin{pmatrix} \mathbb{I} & \mathbf{0} \\ \mathtt{D}^{-1}\mathtt{C} & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathtt{M}_{f_0} & \mathtt{B} \\ \mathbf{0} & \mathtt{D} \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \mathbb{I} & \mathbf{0} \\ \mathtt{D}^{-1}\mathtt{C} & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathtt{M}_{f_0}^{-1} & \mathtt{V} \\ \mathbf{0} & \mathtt{D}^{-1} \end{pmatrix} = \begin{pmatrix} \mathtt{M}_{f_0}^{-1} & \mathtt{V} \\ \mathtt{Z} & \mathtt{D}^{-1} \end{pmatrix}, \end{aligned}$$

which proves the proposition. □

Thus, if $\mathbf{v}$ is the coordinate vector of an element V of $\mathcal{A}$, the first block $\mathbf{w}$ of the solution of the linear system

$$\mathtt{S} \begin{bmatrix} \mathbf{w} \\ \mathbf{w}' \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix}$$

represents the element $f_0^{-1} V$ in the monomial basis B_0 of $\mathcal{A}$.

Similarly, if $\mathbf{l}$ is the coordinate vector of a linear form Λ in the dual basis of B_0 in $\widehat{\mathcal{A}}$, then the subvector $\mathbf{s}$ of the solution of the system

$$\mathtt{S}^t \begin{bmatrix} \mathbf{s} \\ \mathbf{s}' \end{bmatrix} = \begin{bmatrix} \mathbf{l} \\ \mathbf{0} \end{bmatrix}$$

represents the element $f_0^{-1} \cdot \Lambda \in \widehat{\mathcal{A}}$ in the dual basis of B_0.

This yields the following iterative algorithm for selecting the root that minimizes $|f_0|$.

Algorithm 3.7 IMPLICIT INVERSE POWER METHOD.

1. *Choose a random vector* $\mathbf{u}_0$.
2. *Solve the system* $\mathtt{S}^t \begin{bmatrix} \mathbf{v}_{n+1} \\ \mathbf{v}'_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_n \\ \mathbf{0} \end{bmatrix}$.
3. *Let* $\mathbf{v}_{n+1,1}$ *be the first coordinate of* $\mathbf{v}_{n+1}$. *If it is not zero, then compute* $\mathbf{u}_{n+1} = \frac{1}{\mathbf{v}_{n+1,1}} \mathbf{v}_{n+1}$, *replace* n *by* $n+1$ *and go to stage 2; else stop.*

By combining propositions 3.1 and 3.3, we obtain the following proposition:

Proposition 3.8 *Assume that* $\zeta \in \mathcal{Z}(I)$ *is a simple root such that* $0 < |f_0(\zeta)| < |f_0(\zeta')|$ *for all* $\zeta' \in \mathcal{Z}(I)$, $\zeta' \neq \zeta$. *Assume, moreover, that the first row of* $\mathtt{S}$ *is indexed by* 1. *Let* $\sigma = (\zeta^{\alpha})_{\alpha \in E_0}$ *and* $\rho_- = max\{|\frac{f_0(\zeta)}{f_0(\zeta')}|, \zeta' \in \mathcal{Z}(I), \zeta' \neq \zeta\} < 1$. *Then we have* $\|\mathbf{u}_{n+1} - \sigma\| \leq c_-\, \rho_-^n$, *for some constant* $c_- \in \mathbb{R}^+$.

To accelerate this algorithm, we will first compute the LU-decomposition of the matrix $\mathtt{S}^t$. Moreover, as we will see in the following section, the matrix $\mathtt{S}$ and its blocks $\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}$ are structured matrices. They are 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. It also has a structure, which generalizes the structure of Toeplitz matrices to the multivariate case and can be used to simplify multiplication of such a matrix by a vector (see [24], [23], [22], [7]). In this paper, we do not use this structure, but nevertheless we may multiply such matrices by vectors and solve linear systems with such matrices effectively, based on exploiting the sparsity and on the next proposition:

Proposition 3.9 *(see [2]). Let N be the size of* $\mathtt{S}$ *and let C be a bound on the number of arithmetic operations, required to multiply* $\mathtt{S}$ *by a vector. Then each step of algorithms 3.4 and 3.7 can be performed in $\mathcal{O}(C\,N + N\log^2(N))$ arithmetic operations.*

In particular, if $\mathtt{S}$ is sparse, having $\mathcal{O}(N)$ non-zero entries, then $C = \mathcal{O}(N)$, and each step of algorithms 3.4 and 3.7 can be performed by using $\mathcal{O}(N^2)$ arithmetic operations versus known bounds of order N^3 (cf. eg. [27]). This remark is confirmed by our experiments (see section 5).

4 Construction of the Sylvester-like matrices

In this section, we specify three approaches to the construction of matrices $\mathtt{S}$ satisfying the hypotheses 3.2. As soon as these hypotheses are satisfied, we will be able to apply the techniques of section 3.

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 [11]). 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 [16], [31]), and where X is a toric variety, which yields the so called toric resultant (see [11], [30], [3]). These resultants can be computed as a factor of the determinant of a map, which generalizes the Sylvester map for two polynomials in one variable. Let $\mathcal{V}_0, \ldots, \mathcal{V}_n$ 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 $\mathcal{V}$ 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:

$$\begin{array}{rcl} S : \mathcal{V}_0 \times \cdots \times \mathcal{V}_n & \rightarrow & \mathcal{V} \\ (q_0, \ldots, q_n) & \mapsto & \displaystyle\sum_{i=0}^{n} f_i\, q_i. \end{array} \qquad (3)$$

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

$$\begin{array}{cc}
 & \begin{array}{ccccc} \overbrace{\hspace{6em}}^{\mathcal{V}_0} & & \hspace{6em} & \overbrace{\hspace{6em}}^{\mathcal{V}_n} \end{array} \\
\mathcal{V}\left\{\begin{array}{l} \mathbf{x}^{\alpha\,1} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}^{\alpha\,N} \end{array}\right. &
\left[\begin{array}{cc|c|cc} \cdot & & & \cdot & \\ \cdot & & & \cdot & \\ \mathbf{x}^{\beta_{0,1}} f_0 & \cdots & \cdots\cdots\cdots & \mathbf{x}^{\beta_{n,1}} f_n & \cdots \\ \cdot & & & \cdot & \\ \cdot & & & \cdot & \end{array}\right]
\end{array}.$$

It is decomposed into $\mathtt{S} = [\mathtt{S}_0, \ldots, \mathtt{S}_n]$, where $\mathtt{S}_i$ represent the monomial multiples of the polynomial f_i. The rows of this matrix are indexed by the monomials $\mathbf{x}^F$ so that the hypothesis 3.2.1 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 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 [16]). 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 $\leq \nu$ 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 $E_i = (\sum_{j \neq i} C_j)^\delta$ and $F = (\sum_j C_j)^\delta$ allows us to define the desired map S. We refer the reader to [11], [30], [3], …for further details.

Now let us check, step by step, that hypotheses 3.2 are satisfied. In the examples, we will 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 $\mathcal{V}$, it implies that the set of the monomials $\mathbf{x}^F$ which index the rows contains the set $\mathbf{x}^{E_0}$. Therefore, we can partition the matrix $\mathtt{S}$ according to (2), so that $\mathtt{S}_0 = \begin{pmatrix} \mathtt{A} \\ \mathtt{C} \end{pmatrix}$ and $[\mathtt{S}_1, \ldots, \mathtt{S}_n] = \begin{pmatrix} \mathtt{B} \\ \mathtt{D} \end{pmatrix}$, and the hypotheses 3.2.3, 3.2.4 are satisfied.

In the classical case over $\mathbb{P}^n$, the set E_0 is $E_0 = \{(a_1, \ldots, a_n);\ 0 \leq a_i \leq d_i - 1\}$. For generic polynomials $f_1, \ldots, f_n$ of degree n, this set is a basis of $\mathcal{A} = R/(f_1, \ldots, f_n)$ (see [16]). 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 $\mathcal{A} = R/(f_1, \ldots, f_n)$ (see [8], [26]), and the hypothesis 3.2.2 is also satisfied.

To check if hypothesis 3.2.5 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 hypotheses 3.2 are satisfied, we can apply the forward or implicit inverse power iteration method, 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.

4.2 Overconstrained systems

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

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

such that the matrix of this map satisfies hypotheses 3.2. 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 monomials such that $\langle \mathbf{x}^F \rangle = \langle \mathbf{x}^{E_0} \rangle \oplus \langle L \rangle$, (cf. [15], [12]). This yields a square matrix $\mathtt{S}$, which will satisfy hypotheses 3.2.

A case of special interest is the case where $\mathcal{A}$ is of dimension 1, so that there is only one simple root, $\chi = (\chi_1, \ldots, \chi_n)$. A basis of $\mathcal{A}$ is 1, and the matrix of multiplication by x_i is $[\chi_i]$. Then for any matrix $\mathtt{S}$ satisfying hypotheses 3.2 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, and we may apply either of algorithms 3.4 and 3.7.

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 [10]). 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 roundoff errors of the coefficients of the input polynomials, they define an overconstrained system, which has no zeroes, but only pseudo-zeroes, 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 $\mathcal{L}(p)$ 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 $\mathcal{L}(m_{i,j} f_{i,j}) \geq \mathcal{L}(m_{i,j+1} f_{i,j+1})$.

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 $(\mathcal{L}(g_1), \ldots, \mathcal{L}(g_s))$. This set is a basis of $\mathcal{A} = R/(f_1, \ldots, f_m) = R/I$ (see [6]) and contains 1 if $\mathcal{Z}(I) \neq \emptyset$.

We describe how to construct a Sylvester-type matrix $\mathtt{S}$, satisfying hypotheses 3.2, 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 $\mathcal{L}(g_{c(\alpha)})$ of $g_{c(\alpha)}$ for some $c(\alpha) \in \{1, \ldots, m\}$: $\mathbf{x}^\alpha = n_\alpha \mathcal{L}(g_{c(\alpha)})$. 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

$$\begin{array}{rcl} F_{n+1} & = & \cup_{\alpha \in F_n - F_{n-1}} F_{n+1,\alpha} \cup F_n, \\ L_{n+1} & = & \cup_{\alpha \in F_n - F_{n-1}} L_{n+1,\alpha} \cup L_n. \end{array}$$

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

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 $n \geq K$. □

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 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 $\tilde{\mathtt{S}}_1 = \begin{pmatrix} R_1 \\ R'_1 \end{pmatrix}$, 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 $\mathtt{S}_1 = \begin{pmatrix} \mathtt{B} \\ \mathtt{D} \end{pmatrix}$, 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 the hypotheses 3.2 are satisfied, so that this matrix can be used in algorithms 3.4 and 3.7.

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 S is used for the other values of the parameters, assuming that the geometric properties of these systems do not change.

Algorithm 4.2 Solve a parameterized polynomial system for different values of the parameters.

1. *Compute a Gröbner basis of this system, for rational values of the parameters, over a prime field $\mathbb{Z}_p$ for a (good) prime number p.*
2. *Construct the matrix* S *corresponding to this computation and depending on the parameters.*
3. *Substitute the value of the parameters in* S*.*
4. *If the matrix* D *is not invertible, then stop. Otherwise apply algorithms 3.4 or 3.7.*

All the steps of this algorithm can be applied by using the machine precision arithmetic (modular or floating point arithmetic). Here again, the matrix S is structured and sparse, so that each step of algorithms 3.4 or 3.7 can be performed efficiently, taking into account the tolerance to the coefficient errors.

5 Experimental tests

We report here on the results of our (still continuing) experimentation for the implicit inverse power method, applied for computing a selected root. In fact, in our experiments we applied the shifted inverse power method, defining 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[1] developed by R. Pozo; more precisely, we used the GMRES solver with an ILU-preconditionner (see [28] for more details on these solvers). The matrices are generated by the C++ library ALP[2], which implements Macaulay's construction of resultant matrices. We also plan to perform similar experiments based on the implementation of toric resultant matrices by I. Emiris ([3]).

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

In this table, N is the dimension of the matrix S (that is, the matrix has size $N \times N$), S is the number of non-zero entries of the matrix S, D is the dimension of $\mathcal{A}$, n is the number of variables, k is 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 the unix command `time`. This experimentation has been carried out on a Dec Alpha 500 AU workstation with 512M of local memory.

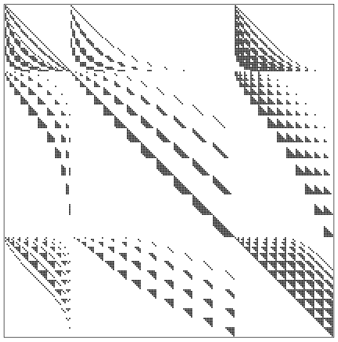
The examples s44, …, s22445 are examples with a few monomials, where Macaulay construction can be applied. The number of solutions is the product of the degree. The first example is a system of 2 equations in 2 variables, both of degree 4, the second is a system of 3 equations in 3 variables of degree 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

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

(see [5]). 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 beeing represented by a point:



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 real), and by this method we are able to select the point whose first coordinate is the nearest to 4. Notice that the matrix of multiplication by this linear form in $\mathcal{A}$ 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 [10]) reduced to an overconstrained system of 6 quadrics in a space of dimension 5. We construct the Macaulay matrix associated to these 6 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.

[1] see http://math.nist.gov/tnt/

[2] see http://www.inria.fr/saga/logiciels/ALP/

References

[1] Auzinger, W., and Stetter, H. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. Intern. Conf. on Numerical Math.* (1988), vol. 86 of *Int. Series of Numerical Math.*, Birkhäuser, pp. 11–30.

[2] Bini, D., and Pan, V. *Polynomial and Matrix Computations, Vol. 1 : Fundamental Algorithms.* Birkhäuser, Boston, 1994.

[3] Canny, J., and Emiris, I. An efficient algorithm for the sparse mixed resultant. In *Proc. Intern. Symp. Applied Algebra, Algebraic Algor. and Error-Corr. Codes (Puerto Rico)* (1993), G. Cohen, T. Mora, and O. Moreno, Eds., vol. 263 of *Lect. Notes in Comp. Science*, Springer, pp. 89–104.

[4] Canny, J., Kaltofen, E., and Lakshman, Y. Solving Systems of Non-linear Polynomial Equations Faster. In *Proc. Intern. Symp. Symbolic and Algebraic Computation (ISSAC'89)* (1989), ACM Press, New York, pp. 121–128.

[5] Corless, R., Gianni, P., and Trager, B. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In *Proc. Intern. Symp. Symbolic and Algebraic Computation (ISSAC'97)* (1997) ACM Press, New York, pp. 133–140.

[6] Cox, D., Little, J., and O'Shea, D. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer, 1992.

[7] Emiris, I., and Pan, V. The structure of sparse resultant matrices. In *Proc. Intern. Symp. Symbolic and Algebraic Computation (ISSAC'97)* (1997) ACM Press, New York, pp. 189–196.

[8] Emiris, I., and Rege, A. Monomial bases and polynomial system solving. In *Proc. Intern. Symp. on Symbolic and Algebraic Computation (ISSAC'94)* (1994), ACM Press, New York, pp. 114–122.

[9] Emsalem, J. Géométrie des points épais. *Bull. Soc. Math. France 106* (1978), 399–416.

[10] Faugeras, O. *Three-dimensional computer vision : a geometric viewpoint*, vol. XXXII. Cambridge, MA ; London : MIT Press, 1993.

[11] Gelfand, I., Kapranov, M., and Zelevinsky, A. *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser, Boston-Basel-Berlin, 1994.

[12] Gianni, P., and Trager, B. Computations with approximate ideals. Preprint, SNAP'96, 1997.

[13] Giusti, M., and Heintz, J. La détermination des points isolés et de la dimension d'une variété algebrique peut se faire en temps polynomial. In *Proc. Intern. Meeting on Commutative Algebra* (Cortona, 1991), vol. XXXIV of *Symp. Mathematica*, pp. 216–255.

[14] Golub, G. H., and Van Loan, C. F. *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, Maryland (1996) (third edition).

[15] Lazard, D. Résolution des systèmes d'équations algébriques. *Theor. Comp. Science 15* (1981), 77–110.

[16] Macaulay, F. Some formulae in elimination. *Proc. London Math. Soc. 1*, 33 (1902), 3–27.

[17] Manocha, D. Computing selected solutions of polynomial equations. In *Proc. Intern. Symp. Symbolic and Algebraic Computation* (ISSAC'94) (1994), ACM Press, New York, pp. 1–8.

[18] Manocha, D., and Canny, J. Multidimensional resultant algorithms. *J. Symbolic Computation 15* (1993), 99–122.

[19] Marinari, M., Mora, T., and Möller, H. Gröbner duality and multiplicities in polynomial system solving. In *Proc. Intern. Symp. Symbolic and Algebraic Computation (ISSAC'95)* (1995), ACM Press, New York, pp. 167–179.

[20] Mourrain, B. Isolated points, duality and residues. *J. Pure Applied Algebra. Special Issue on Algorithms for Algebra 117 & 118* (May 1997), 469–494.

[21] Mourrain, B. Solving polynomial systems by matrix computations. Manuscript. INRIA Sophia-Antipolis, France. Submitted for publication, 1997.

[22] Mourrain, B., and Pan, V. Multivariate polynomials, duality and structured matrices, submitted, 1997.

[23] Mourrain, B., and Pan, V. Y. Multidimensional structured matrices and polynomial systems. *Calcolo, Special Issue, workshop on Toeplitz matrices: Structure, Algorithms and Applications 33* (1996), 389–401.

[24] Mourrain, B., and Pan, V. Y. Solving special polynomial systems by using structured matrices and algebraic residues. In *Proc. of the Workshop on Foundations of Comp. Math.* (Rio de Janeiro, 1997) (1997), F. Cucker and M. Shub, Eds., Springer, pp. 287–304.

[25] Mourrain, B., and Pan, V. Y. Asymptotic acceleration of solving polynomial systems, *30th Ann. Symp. on Theory of Computing* (1998) ACM Press, New York.

[26] Pedersen, P. S., and Sturmfels, B. Mixed monomial basis. In *Effective Methods in Algebraic Geometry (MEGA'94)* (Santander (Spain), 1994), vol. 143 of *Progress in Math.*, Birkhäuser, pp. 285–306.

[27] Renegar, J. On the worst-case arithmetic complexity of approximating zeros of system of polynomials. *SIAM J. on Computing 18* (1989), 350–370.

[28] Saad, Y. *Iterative methods for sparse linear systems.* PWS series in Computer Science. PWS, 1996.

[29] Stetter, H. J. Eigenproblems are at the Heart of Polynomial System Solving. *SIGSAM Bulletin 30*, 4 (1996) ACM Press, New York, 22–25.

[30] Sturmfels, B. Sparse elimination theory. In *Computational Algebraic Geometry and Commutative Algebra* (1993), D. Eisenbud and L. Robianno, Eds., Cambridge Univ. Press, pp. 264–298. (Proc. Cortona, June 1991).

[31] Van der Waerden, B. *Modern algebra, Vol. II.* Frederick Ungar Publishing Co., 1948.