

Truncated Normal Forms for Solving Polynomial Systems

Simon Telen, Bernard Mourrain, Marc Van Barel

simon.telen@cs.kuleuven.be, bernard.mourrain@inria.fr, marc.vanbarel@cs.kuleuven.be

Abstract

In this poster we present the results of [10]. We consider the problem of finding the common roots of a set of polynomial functions defining a zero-dimensional ideal I in a ring R of polynomials over \mathbb{C} . We propose a general algebraic framework to find the solutions and to compute the structure of the quotient ring R/I from the cokernel of a resultant map. This leads to what we call Truncated Normal Forms (TNFs). Algorithms for generic dense and sparse systems follow from the classical resultant constructions. In the presented framework, the concept of a border basis is generalized by relaxing the conditions on the set of basis elements. This allows for algorithms to adapt the choice of basis in order to enhance the numerical stability. We present such an algorithm. The numerical experiments show that the methods allow to compute all zeros of challenging systems (high degree, with a large number of solutions) in small dimensions with high accuracy.

1 Introduction

Several problems in science and engineering boil down to the problem of finding the (isolated) common roots of a set of multivariate polynomial equations. In mathematical terms, if $R = \mathbb{C}[x_1, \dots, x_n]$ is the ring of polynomials over \mathbb{C} in the n indeterminates x_1, \dots, x_n and $I = (f_1, \dots, f_s) \subset R$, is the ideal generated by polynomials $f_i \in R$ ($i = 1, \dots, s$), the problem can be formulated as finding the points in the algebraic set $\mathbb{V}(I) = \{z \in \mathbb{C}^n : f_i(z) = 0, i = 1, \dots, s\} = \{z \in \mathbb{C}^n : f(z) = 0, \forall f \in I\}$.

There exist several methods to find all the roots of a set of polynomial equations [9, 1]. Here, we are mainly interested in algebraic methods [5, 3, 7, 11]. When the roots of the ideal I are isolated points (I is called *zero-dimensional* in this case), eigenvalue, eigenvector techniques can be used to find these roots [4, 5, 8]. One can find the origins of these techniques in ancient works on resultants by Bézout, Sylvester, Cayley, Macaulay, The key observation to translate the root finding problem into a linear algebra problem is a standard result in algebraic geometry that states that R/I is finitely generated over \mathbb{C} as a \mathbb{C} -algebra (it is a finite dimensional \mathbb{C} -vector space) if and only if I is zero-dimensional. Moreover, $\dim_{\mathbb{C}}(R/I) = \delta$, where δ is the number of points defined by I , counting multiplicities. See for instance [2, Theorem 6, Chapter 5, §3]. The map $M_f : R/I \rightarrow R/I : g + I \mapsto fg + I$, representing ‘multiplication by $f + I$ ’ in R/I is linear. Fixing a basis for R/I , M_f is a $\delta \times \delta$ matrix. A well known result is that the eigenvalue structure of such *multiplication matrices* reveals the coordinates of the points in $\mathbb{V}(I)$ [4, 3, 8].

In general, normal form algorithms execute the following two main steps.

1. Compute the multiplication matrices M_{x_1}, \dots, M_{x_n} with respect to a suitable basis of R/I .
2. Compute the points $\mathbb{V}(I)$ from the eigenvalue structure of these matrices.

In this work we focus on step (1). Once a basis $\mathcal{B} = \{b_1 + I, \dots, b_\delta + I\}$ of R/I is fixed, the i -th column of M_{x_j} corresponds to the coordinates of $x_j b_i + I$ in \mathcal{B} . These coordinates are found by projecting $x_j b_i$ onto $B = \text{span}(b_1, \dots, b_\delta)$ along I . In [11] it is shown that the choice of basis for R/I can be crucial for the accuracy of the computed multiplication maps and a monomial basis \mathcal{B} is chosen using column pivoted QR factorization on a large Macaulay-type matrix for solving generic, dense problems. In [10] a general

algebraic framework is proposed for constructing normal forms with respect to a numerically justified basis for R/I using similar numerical linear algebra techniques. Some algorithms are proposed for finding affine, toric, homogeneous and multihomogeneous roots from multiplication tables. The resulting bases consist of monomials, although the theorems allow much more general constructions. In the next section, we present the main results of [10]. Section 4 shows some numerical experiments. In conclusion, we summarize the results.

2 Truncated normal forms

Throughout the paper, let $R = \mathbb{C}[x_1, \dots, x_n]$ be the ring of polynomials over \mathbb{C} in the variables x_1, \dots, x_n and let $I \subset R$ be a zero-dimensional ideal such that $\dim_{\mathbb{C}}(R/I) = \delta$, the number of points defined by I , counting multiplicities.

Definition 1. A normal form on R w.r.t. I is an R -map $\mathcal{N} : R \rightarrow B$ where $B \subset R$ is a vector subspace of dimension δ over \mathbb{C} such that

$$0 \longrightarrow I \longrightarrow R \xrightarrow{\mathcal{N}} B \longrightarrow 0$$

is exact and $\mathcal{N}|_B = \text{id}_B$.

Note that from this definition it follows that $B \simeq R/I$ and multiplication with $f + I$ in R/I is given by $M_{x_i} : B \rightarrow B : b \mapsto \mathcal{N}(x_i b)$.

Definition 2. Let $B \subset V \subset R$ with B, V finite dimensional vector subspaces and $\dim_{\mathbb{C}}(B) = \delta$. A Truncated Normal Form (TNF) on V w.r.t. I is a linear map $\mathcal{N} : V \rightarrow B$ such that \mathcal{N} is the restriction of a normal form to V . That is,

$$0 \longrightarrow I \cap V \longrightarrow V \xrightarrow{\mathcal{N}} B \longrightarrow 0$$

is exact and $\mathcal{N}|_B = \text{id}_B$.

A TNF is a linear map between finite dimensional vector spaces, a normal form is not. If we have a TNF such that $x_i \cdot B \subset V, 1 \leq i \leq n$ then multiplication with $x_i + I$ in R/I is given by $M_{x_i} : B \rightarrow B : b \mapsto \mathcal{N}(x_i b)$, since \mathcal{N} comes from a normal form. The following theorem is the main result of [10] and it gives a sufficient condition under which a map $N : V \rightarrow \mathbb{C}^\delta$ leads to a TNF.

Theorem 1. Suppose we have a finite dimensional $V \subset R$ with subspace $W = \{v \in V : x_i \cdot v \in V, \forall i\}$ and a linear map $N : V \rightarrow \mathbb{C}^\delta$ such that

1. $\ker N \subset I \cap V$,
2. $\exists u \in V$ such that $u + I$ is a unit in R/I ,
3. $N|_W$ is surjective.

Then for any δ -dimensional subspace $B \subset W$ with $N|_B$ invertible, $\mathcal{N} = N|_B^{-1} \circ N : V \rightarrow B$ is a TNF coming from the restriction of a normal form to V .

The theorem reduces the problem of computing multiplication maps (referred to as step (1) of any normal form algorithm earlier) to the problem of finding N, V with the right properties.

3 Cokernel of resultant maps

Our aim is to compute TNFs in a numerically stable way. To do so, we will use resultant maps.

Definition 3. A resultant map is a map

$$\text{Res} : V_1 \times \cdots \times V_n \longrightarrow V : (q_1, \dots, q_n) \longmapsto q_1 f_1 + \cdots + q_n f_n.$$

with $V_i, V \subset R$ finite dimensional vector subspaces.

Note that $\text{im}(\text{Res}) \subset I \cap V$. Suppose $\dim_{\mathbb{C}}(V/\text{im}(\text{Res})) = \delta$. If $N : V \rightarrow \mathbb{C}^\delta$ is the cokernel of Res, $B \subset V$ and $N|_B$ is invertible, then $\ker N = \text{im Res} \subset I \cap V$,

$$(N|_B^{-1} \circ N) \circ \text{Res} = 0, \quad (N|_B^{-1} \circ N)|_B = \text{id}_B.$$

This indicates that we may hope to compute TNFs from the cokernel of resultant maps. It turns out that in the following interesting cases, the cokernel of a specific resultant map satisfies the assumptions of Theorem 1. These maps can be constructed from the given generators of I .

1. When the equations are dense of degree $d_i = \deg(f_i)$, N is the cokernel of the resultant map defined by

$$V_i = R_{\leq \sum_{j \neq i} (d_j - 1)}, \quad V = R_{\leq \sum_{i=1}^n d_i - (n-1)},$$

where $R_{\leq d}$ is the vector subspace of polynomials of degree at most d .

2. If the equations are sparse and generic with respect to their Newton polytopes, N is the cokernel of the resultant map defined as follows. Denote $P_i \subset \mathbb{R}^n$ for the Newton polytope of f_i and let $v =$ be a generic small real n -vector.

$$V_i = \bigoplus_{\alpha \in A_i} \mathbb{C} \cdot x^\alpha, \quad V = \bigoplus_{\alpha \in A} \mathbb{C} \cdot x^\alpha$$

with $A_i = (P_1 + \dots + \hat{P}_i + \dots + P_n + \Delta_n + v) \cap \mathbb{Z}^n$ ($\hat{\cdot}$ means this term is left out of the sum), $A = (P_1 + \dots + P_n + \Delta_n + v) \cap \mathbb{Z}^n$ and Δ_n the simplex.

Once the map N is computed, we use it to construct a normal form by computing $\mathcal{N} = N|_B^{-1} \circ N$, with $B \subset W \subset V$, $\dim_{\mathbb{C}}(B) = \delta$. For numerical stability, it is important that the subspace B is chosen such that $N|_B$ is well conditioned. This is done by using a column pivoted QR decomposition of $N|_W$ as in [11].

4 Numerical results

We show some numerical results to illustrate the effectiveness of the TNF method. More experiments can be found in [10, 11].

Dense systems

We define a *generic dense system* of degree d in n variables to be a system defined by n polynomials in $\mathbb{C}[x_1, \dots, x_n]$ such that all polynomials have coefficients with all monomials of degree $\leq d$ drawn from a normal distribution with zero mean and $\sigma = 1$. If we consider n such generic hypersurfaces, they intersect in $\delta = d^n$ points according to Bézout's theorem. In Table 1, r_{\max} gives an upper bound for the residual of all δ solutions and t is the total computation time in minutes. For a large number of variables, the algebraic method is slow compared to homotopy continuation solvers. However, our solver finds *all* solutions, whereas homotopy solvers do not for systems of high degree. A detailed comparison can be found in [10].

n	d	δ	r_{\max}	t (min)
2	50	2500	$5.55 \cdot 10^{-11}$	0.3
2	80	6400	$1.97 \cdot 10^{-10}$	4.9
2	100	10000	$1.31 \cdot 10^{-9}$	18
2	150	22500	$8.84 \cdot 10^{-9}$	184
2	170	28900	$1.08 \cdot 10^{-7}$	370
3	7	343	$6.71 \cdot 10^{-12}$	0.03
3	13	2197	$2.86 \cdot 10^{-11}$	3.7
4	4	256	$1.52 \cdot 10^{-13}$	0.15
5	3	243	$1.84 \cdot 10^{-12}$	1.18

Table 1: Numerical results for generic dense systems.

A sparse example

We consider the system given by

$$\begin{aligned} f_1 &= 12x_1x_2x_3^{12} + 7x_1^2x_2^7x_3^6 + 4x_1^{10}x_2^{11}x_3^8 + 4x_1^6x_2^4x_3^7 + 5, \\ f_2 &= 15x_1^{10}x_2^4x_3^2 + 4x_1^3x_2^6x_3^6 + 10x_1x_2^{10}x_3^8 + 11x_1^6x_2^{11}x_3^8 + 12, \\ f_3 &= 10x_1^7x_2^4x_3^6 + 4x_1^{10}x_2x_3 + 4x_1^2x_2^{12}x_3^9 + 14x_1^{10}x_2^5x_3 + 2. \end{aligned}$$

The mixed volume (computed using PHCpack [12]) is 2352. Our algorithm finds 2352 solutions, 2 of which are real. All solutions are simple. They are found by a Schur decomposition of the $M_{x_i}, i = 1, \dots, 3$. Computations with polytopes (except for the mixed volume) are done using polymake [6]. The total computation time is about 294 seconds. All solutions are found with a residual smaller than $3.1 \cdot 10^{-12}$.

5 Conclusion

We have presented a general algebraic framework for computing the algebra structure of R/I where I is a zero-dimensional ideal. Specific algorithms for generic square systems are presented. Numerical results show that the multiplication operators can be constructed using floating point arithmetic with only little loss of accuracy. The resulting algorithm is competitive with homotopy continuation solvers for small n and it computes numerical approximations of *all* solutions.

References

- [1] E. Cattani, D. A. Cox, G. Chèze, A. Dickenstein, M. Elkadi, I. Z. Emiris, A. Galligo, A. Kehrein, M. Kreuzer, and B. Mourrain. *Solving Polynomial equations: foundations, algorithms, and applications (Algorithms and Computation in Mathematics)*. Springer-Verlag New York, Inc., 2005.
- [2] D. A. Cox, J. Little, and D. O’Shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992.
- [3] D. A. Cox, J. Little, and D. O’Shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.
- [4] M. Elkadi and B. Mourrain. *Introduction à la résolution des systèmes polynomiaux*, volume 59 of *Mathématiques et Applications*. Springer, 2007.
- [5] I. Z. Emiris and B. Mourrain. Matrices in Elimination Theory. *Journal of Symbolic Computation*, 28(1-2):3–44, 1999.
- [6] M. Joswig, B. Müller, and A. Paffenholz. **polymake** and lattice polytopes. In *21st International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2009)*, Discrete Math. Theor. Comput. Sci. Proc., AK, pages 491–502. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2009.
- [7] B. Mourrain. A New Criterion for Normal Form Algorithms. In *Proceedings of the 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, LNCS, pages 430–443, London, UK, 1999. Springer-Verlag.
- [8] H. J. Stetter. Matrix eigenproblems are at the heart of polynomial system solving. *ACM SIGSAM Bulletin*, 30(4):22–25, 1996.
- [9] B. Sturmfels. *Solving Systems of Polynomial Equations*. Number 97 in CBMS Regional Conferences. Amer. Math. Soc., 2002.
- [10] S. Telen, B. Mourrain, and M. Van Barel. Solving polynomial systems via a stabilized representation of quotient algebras. *arXiv preprint arXiv:1711.04543*, 2017.
- [11] S. Telen and M. Van Barel. A stabilized normal form algorithm for generic systems of polynomial equations. *Journal of Computational and Applied Mathematics*, 342:119–132, 2018.
- [12] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2):251–276, 1999.