Polynomial Equations: Theory and Practice

Simon Telen

▶ **To cite this version:**

Simon Telen. Polynomial Equations: Theory and Practice. Michal Kočvara; Bernard Mourrain; Cordian Riener. Polynomial Optimization, Moments, and Applications, Springer, pp.215-240, In press, 10.48550/arXiv.2210.04939 . hal-04076672

HAL Id: hal-04076672
https://hal.science/hal-04076672v1

Submitted on 21 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Polynomial Equations: Theory and Practice

Simon Telen

Abstract Solving polynomial equations is a subtask of polynomial optimization. This article introduces systems of such equations and the main approaches for solving them. We discuss critical point equations, algebraic varieties, and solution counts. The theory is illustrated by many examples using different software packages.

1 Polynomial equations in optimization

Polynomial equations appear in many fields of science and engineering. Some examples are chemistry [19, 42], molecular biology [23], computer vision [32], economics and game theory [47, Chapter 6], topological data analysis [9], and partial differential equations [47, Chapter 10]. For an overview and more references, see [10, 14]. This article will be a chapter in the forthcoming book *Polynomial optimisation, moments and applications* presenting research acitivies conducted in the European Network POEMA. In that context, polynomial equations arise from optimization problems.

Let us consider the problem of minimizing a polynomial objective function $f(x_1,\dots,x_k)$ over the set $X = \{x \in \mathbb{R}^k : h_1(x) = \cdots = h_\ell(x) = 0\} \subset \mathbb{R}^k$, where also $h_1,\dots,h_\ell$ are polynomials in the k variables $x = (x_1,\dots,x_k)$. This is a *polynomial optimization problem* [35], often written as

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^k} & f(x_1,\dots,x_k), \\ \text{subject to} & h_1(x_1,\dots,x_k) = \cdots = h_\ell(x_1,\dots,x_k) = 0. \end{array} \tag{1}$$

Introducing new variables $\lambda_1,\dots,\lambda_\ell$ we obtain the Lagrangian $L = f - \lambda_1 h_1 - \cdots - \lambda_\ell h_\ell$, whose partial derivatives give the optimality conditions

Simon Telen
CWI, Science Park 123, 1098XG Amsterdam and Max Planck Institute for Mathematics in the Sciences (current), Inselstrasse 22, 04103 Leipzig, e-mail: simon.telen@mis.mpg.de

$$\frac{\partial L}{\partial x_1} = \cdots = \frac{\partial L}{\partial x_k} = h_1 = \cdots = h_\ell = 0. \tag{2}$$

A solution in $\mathbb{R}^k$ is a candidate minimizer. Many methods for solving the equations (2), like those presented in Section 4, compute all complex solutions first and then select the real ones among them. The number of solutions over $\mathbb{C}$ is typically finite. We present two examples of (1). First, we minimize the distance to algebraic varieties.

Example: Euclidean distance degree

Given a point $y = (y_1, \ldots, y_k) \in \mathbb{R}^k$, we consider the squared Euclidean distance function $f(x_1, \ldots, x_k) = \|x - y\|_2^2 = (x_1 - y_1)^2 + \cdots + (x_k - y_k)^2$. As above, X is the set $\{x \in \mathbb{R}^k \mid h_1 = \cdots = h_\ell = 0\}$. The solution x^* of the optimization problem (1) is

$$x^* = \arg\min_{x \in X} \|x - y\|_2^2, \tag{3}$$

i.e. the point on X that is closest to y. The algebraic complexity of this problem is studied in [20]. For instance, let $k = 2, \ell = 1$ and let X be the unit ball with respect to the 4-norm $\|\cdot\|_4$: $h = h_1 = x_1^4 + x_2^4 - 1$. We want to find the point on X closest to $y = (2, 1.4)$. In `Mathematica` [**?**], one solves (2) as follows:

```
f = (x1 - 2)^2 + (x2 - 1.4)^2; h = x1^4 + x2^4 - 1; L = f - lambda*h;
NSolve[{D[L, x1] == 0 &&  D[L, x2] == 0 &&  h == 0}, Reals]
```

This returns two critical points on X. One of them minimizes the distance to y, the other maximizes it. The minimizer is $x^* = (0.904944, 0.757564)$. If we delete the option `Reals`, the program returns 16 complex solutions.

Second, we set up a polynomial optimization problem from system identification.

Example: parameter estimation for system identification

System identification is an engineering discipline that aims to construct models for dynamical systems from measured data. A model explains the relationship between input, output, and noise. It depends on a set of *model parameters*, which are selected to best fit the measured data. A *discrete-time, single-input single-output linear time-invariant system* with input sequence $u : \mathbb{Z} \to \mathbb{R}$, output sequence $y : \mathbb{Z} \to \mathbb{R}$ and white noise sequence $e : \mathbb{Z} \to \mathbb{R}$ is often modeled by

$$A(q)\, y(t) = \frac{B_1(q)}{B_2(q)}\, u(t) + \frac{C_1(q)}{C_2(q)}\, e(t). \tag{4}$$

Here $A, B_1, B_2, C_1, C_2 \in \mathbb{C}[q]$ are unknown polynomials of a fixed degree in the *backward shift operator* q, acting on $s : \mathbb{Z} \to \mathbb{R}$ by $qs(t) = s(t - 1)$. The model parameters are the coefficients of these polynomials, which are to be estimated. Clearing denominators in (4) gives

$$A(q)B_2(q)C_2(q)y(t) = B_1(q)C_2(q)u(t) + B_2(q)C_1(q)e(t). \tag{5}$$

Suppose we have measured $u(0), \ldots, u(N), y(0), \ldots, y(N)$. Let $d = \max(d_A + d_{B_2} + d_{C_2}, d_{B_1} + d_{C_2}, d_{B_2} + d_{C_1})$, where $d_A, d_{B_1}, d_{B_2}, d_{C_1}, d_{C_2}$ are the degrees of our polynomials. Writing (5) for $t = d, d+1, \ldots, N$, we find algebraic relations among the coefficients of A, B_1, B_2, C_1, C_2. The model parameters are estimated by solving

$$\min_{\Theta \in \mathbb{R}^k} \quad e(0)^2 + \ldots + e(N)^2 \quad \text{subject to} \quad (5) \text{ is satisfied for } t = d, \ldots, N$$

where Θ consists of $e(0), \ldots, e(N)$ and the coefficients of A, B_1, B_2, C_1, C_2. We refer to [3, Section 1.1.1] for a worked-out example and more references.

More general versions of (1) add inequality constraints of the type $q_1(x) \geq 0, \ldots, q_{\ell'}(x) \geq 0$, where $q_1, \ldots, q_{\ell'}$ are polynomials. Such problems can be handled using sums-of-squares relaxations [34].

Our aim in this article is to introduce systems of polynomial equations in general, and methods for solving them. The reader is encouraged to try out these methods for polynomial optimization, for instance, in a Euclidean distance computation (3) or in system identification. Section 2 discusses solution sets to polynomial equations, also called *algebraic varieties*, and root finding over different fields. In Section 3, we present several classical upper bounds on the number of solutions. Section 4 is about normal form methods and homotopy continuation methods, which are two different important approaches to solving polynomial equations. Finally, Section 5 contains a case study in which we apply these methods to compute 27 lines on a cubic surface.

Acknowledgements. This article is based on an introductory lecture given at the workshop *Solving polynomial equations and applications* organized at CWI, Amsterdam in October 2022. I thank Monique Laurent for involving me in this workshop, and all other speakers and attendants for making it a success. I was supported by a Veni grant from the Netherlands Organisation for Scientific Research (NWO).

2 Systems of equations and algebraic varieties

Let K be a field with algebraic closure $\overline{K}$, e.g., $K = \mathbb{R}$ and $\overline{K} = \mathbb{C}$. The polynomial ring with n variables and coefficients in K is $R = K[x_1, \ldots, x_n]$. We abbreviate $x = (x_1, \ldots, x_n)$ and use variable names x, y, z rather than x_1, x_2, x_3 when n is small. Elements of R are polynomials, which are functions $f : \overline{K}^n \to \overline{K}$ of the form

$$f(x) = \sum_{\alpha \in \mathbb{N}^n} c_{(\alpha_1, \ldots, \alpha_n)}\, x_1^{\alpha_1} \cdots x_n^{\alpha_n} = \sum_{\alpha \in \mathbb{N}^n} c_\alpha\, x^\alpha,$$

with finitely many nonzero coefficients $c_\alpha \in K$. A system of polynomial equations is

$$f_1(x) = \cdots = f_s(x) = 0, \tag{6}$$

where $f_1, \ldots, f_s \in R$. By a *solution* of (6), we mean a point $x \in \overline{K}^n$ satisfying all of these s equations. *Solving* usually means finding coordinates for all solutions. This makes sense only when the set of solutions is finite, which typically happens when $s \geq n$. However, systems with infinitely many solutions can be 'solved' too, in an appropriate sense [45]. We point out that one is often mostly interested in solutions $x \in K^n$ over the ground field K. The reason for allowing solutions over the algebraic closure $\overline{K}$ is that many solution methods, like those discussed in Section 4, intrinsically compute all such solutions. For instance, (2) is a polynomial system with $n = s = k + \ell$, and the field is $K = \mathbb{R}$. Here are some other examples.

Example: univariate polynomials ($n = 1$) and linear equations

When $n = s = 1$, solving the polynomial system defined by $f = a_0 + a_1 x + \cdots + a_d x^d \in K[x]$, with $a_d \neq 0$, amounts to finding the roots of $f(x) = 0$ in $\overline{K}$. These are the eigenvalues of the $d \times d$ *companion matrix*

$$C_f = \begin{pmatrix} & & -a_0/a_d \\ 1 & & -a_1/a_d \\ & \ddots & \vdots \\ & 1 & -a_{d-1}/a_d \end{pmatrix} \tag{7}$$

of f, whose characteristic polynomial is $\det(x \cdot \mathrm{id} - C_f) = a_d^{-1} \cdot f$.

When $f_i = \sum_{j=1}^n a_{ij} x_j - b_i$ are given by affine-linear functions, (6) is a linear system of the form $Ax = b$, with $A \in K^{s \times n}$, $b \in K^s$.

This example shows that, after a trivial rewriting step, the univariate and affine-linear cases are reduced to a *linear algebra* problem. Here, we are mainly interested in the case where $n > 1$, and some equations are of degree > 1. Such systems require tools from *nonlinear algebra* [38]. We proceed with an example in two dimensions.

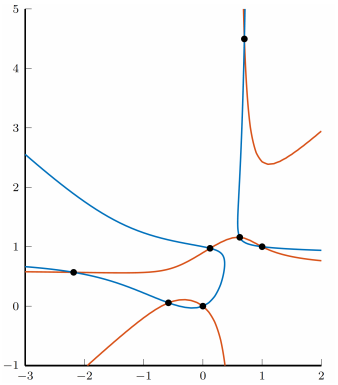
Example: intersecting two curves in the plane

Let $K = \mathbb{Q}$ and $n = s = 2$. We work in the ring $R = \mathbb{Q}[x, y]$ and consider the system of equations $f(x, y) = g(x, y) = 0$ where

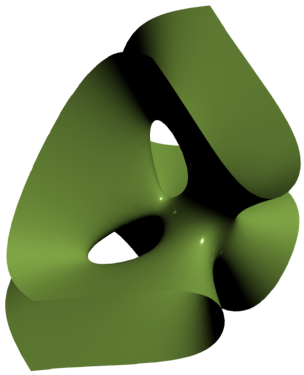
$$\begin{aligned} f &= -7x - 9y - 10x^2 + 17xy + 10y^2 + 16x^2y - 17xy^2, \\ g &= 2x - 5y + 5x^2 + 5xy + 5y^2 - 6x^2y - 6xy^2. \end{aligned} \tag{8}$$

Geometrically, we can think of $f(x, y) = 0$ as defining a curve in the plane. This is the orange curve shown in Fig. 1a. The curve defined by $g(x, y) = 0$ is shown in blue. The set of solutions of $f = g = 0$ consists of points $(x, y) \in \overline{\mathbb{Q}}^2$ satisfying $f(x, y) = g(x, y) = 0$. These are the intersection points of the two curves. There are seven such points in $\overline{\mathbb{Q}}^2$, of which two lie in $\mathbb{Q}^2$. These are the points $(0, 0)$ and

$(1,1)$. Note that all seven solutions are real: replacing $\mathbb{Q}$ by $K = \mathbb{R}$, we count as many solutions over K as over $\overline{K} = \mathbb{C}$.



(a) Curves defined by f, g from (8).



(b) The Clebsch surface.

Fig. 1: Algebraic curves in the plane ($n = 2$) and an algebraic surface ($n = 3$).

The set of solutions of the polynomial system (6) is called an *affine variety*. We denote this by $V_{\overline{K}}(f_1, \ldots, f_s) = \{x \in \overline{K}^n \mid f_1(x) = \cdots = f_s(x) = 0\}$, and replace $\overline{K}$ by K in this notation to mean only the solutions over the ground field. Examples of affine varieties are the red curve $V_{\overline{K}}(f)$ and the set of black dots $V_{\overline{K}}(f, g)$ in Fig. 1a. In the case of $V_{\overline{K}}(f)$, Fig. 1a only shows the real part $V_{\overline{K}}(f) \cap \mathbb{R}^2$.

Example: surfaces in $\mathbb{R}^3$

Let $K = \mathbb{R}$ and consider the affine variety $V = V_{\mathbb{C}}(f)$ where

$$\begin{aligned} f \;&= 81(x^3 + y^3 + z^3) - 189(x^2y + x^2z + y^2x + y^2z + xz^2 + yz^2) + 54xyz \\ &\quad + 126(xy + xz + yz) - 9(x^2 + y^2 + z^2) - 9(x + y + z) + 1. \end{aligned} \tag{9}$$

Its real part $V_{\mathbb{R}}(f)$ is the surface shown in Fig. 1b. The variety V is called the *Clebsch surface*. It is a *cubic* surface because it is defined by an equation of degree three. We will revisit this surface in Section 5. Note that f is invariant under permutations of the variables, i.e., $f(x,y,z) = f(y,x,z) = f(z,y,x) = f(x,z,y) = f(z,x,y) = f(y,z,x)$. This reflects in the symmetries of the surface $V_{\mathbb{R}}(f)$. Many polynomials from applications have similar symmetry properties. Exploiting this in computations is an active area of research, see for instance [29].

More pictures of real affine varieties can be found, for instance, in [16, Chapter 1, §2], or in the algebraic surfaces gallery hosted at

https://homepage.univie.ac.at/herwig.hauser/bildergalerie/gallery.html.

We now briefly discuss commonly used fields K. In many engineering applications, the coefficients of $f_1, \ldots, f_s$ lie in $\mathbb{R}$ or $\mathbb{C}$. Computations in such fields use floating point arithmetic, yielding approximate results. The required quality of the approximation depends on the application. Other fields also show up: polynomial systems in cryptography often use $K = \mathbb{F}_q$, see for instance [44]. Equations of many prominent algebraic varieties have integer coefficients, i.e., $K = \mathbb{Q}$. Examples are determinantal varieties (e.g., the variety of all $m \times n$ matrices of rank $< \min(m, n)$), Grassmannians in their Plücker embedding [38, Chapter 5], discriminants and resultants [49, Sections 3.4, 5.2] and toric varieties obtained from monomial maps [50, Section 2.3]. In number theory, one is interested in studying *rational points* $V_{\mathbb{Q}}(f_1, \ldots, f_s) \subset V_{\overline{\mathbb{Q}}}(f_1, \ldots, f_s)$ on varieties defined over $\mathbb{Q}$. Recent work in this direction for del Pezzo surfaces can be found in [39, 18]. Finally, in *tropical geometry*, coefficients come from *valued fields* such as the p-adic numbers $\mathbb{Q}_p$ or the Puiseux series $\mathbb{C}\{\{t\}\}$ [36]. Solving over the field of Puiseux series is also relevant for *homotopy continuation methods*, see Section 4.2. We end the section with two examples highlighting the difference between $V_K(f_1, \ldots, f_s)$ and $V_{\overline{K}}(f_1, \ldots, f_s)$.

Example: Fermat's last theorem

Let $k \in \mathbb{N} \setminus \{0\}$ be a positive integer and consider the equation $f = x^k + y^k - 1 = 0$. For any k, the variety $V_{\overline{\mathbb{Q}}}(f)$ has infinitely many solutions in $\overline{\mathbb{Q}}^2$. For $k = 1, 2$, there are infinitely many *rational* solutions, i.e. solutions in $\mathbb{Q}^2$. For $k \geq 3$, the only solutions in $\mathbb{Q}^2$ are $(1, 0)$, $(0, 1)$ and, when k is even, $(-1, 0)$, $(0, -1)$ [17].

Example: computing real solutions

The variety $V_{\mathbb{C}}(x^2 + y^2)$ consists of the two lines $x + \sqrt{-1} \cdot y = 0$ and $x - \sqrt{-1} \cdot y = 0$ in $\mathbb{C}^2$. However, the real part $V_{\mathbb{R}}(x^2 + y^2) = \{(0, 0)\}$ has only one point. If we are interested only in this real solution, we may replace $x^2 + y^2$ with the two polynomials x, y, which have the property that $V_{\mathbb{R}}(x^2 + y^2) = V_{\mathbb{R}}(x, y) = V_{\mathbb{C}}(x, y)$. After this replacing step, an algorithm that computes all *complex* solutions will still recover only the interesting solutions. It turns out that such a 'better' set of equations can always be computed. The new polynomials generate the *real radical ideal* associated to the original equations [38, Sec. 6.3]. For recent computational progress, see [1]. A different approach for real root finding in bounded domains is *subdivision*, see [41].

3 Number of solutions

A univariate polynomial $f \in \mathbb{C}[x]$ of degree d has at most d roots in $\mathbb{C}$. Moreover, d is the *expected* number of roots. We now formalize this. Consider the *family*

$$\mathcal{F}(d) \;=\; \{a_0 + a_1 x + \cdots + a_d x^d \mid (a_0, \ldots, a_d) \in \mathbb{C}^{d+1}\} \;\simeq\; \mathbb{C}^{d+1} \tag{10}$$

of polynomials of degree at most d. There is an affine variety $\nabla_d \subset \mathbb{C}^{d+1}$, such that all $f \in \mathcal{F}(d) \setminus \nabla_d$ have precisely d roots in $\mathbb{C}$. Here $\nabla_d = V_{\mathbb{C}}(\Delta_d)$, where Δ_d is a polynomial in the coefficients a_i of $f \in \mathcal{F}(d)$. Equations for small d are

$$\begin{aligned}
\Delta_1 &= a_1,\\
\Delta_2 &= a_2 \cdot (a_1^2 - 4\,a_0 a_2),\\
\Delta_3 &= a_3 \cdot (a_1^2 a_2^2 - 4\,a_0 a_2^3 - 4\,a_1^3 a_3 + 18\,a_0 a_1 a_2 a_3 - 27\,a_0^2 a_3^2),\\
\Delta_4 &= a_4 \cdot (a_1^2 a_2^2 a_3^2 - 4\,a_0 a_2^3 a_3^2 - 4\,a_1^3 a_3^3 + 18\,a_0 a_1 a_2 a_3^3 + \cdots + 256\,a_0^3 a_4^3).
\end{aligned}$$

Notice that $\Delta_d = a_d \cdot \widetilde{\Delta}_d$, where $\widetilde{\Delta}_d$ is the *discriminant* for degree d polynomials. Similar results exist for families of polynomial systems with $n > 1$, which bound the number of isolated solutions from above by the *expected* number. This section states some of these results. *It assumes that $K = \overline{K}$ is algebraically closed.*

3.1 Bézout's theorem

Let $R = K[x] = K[x_1, \ldots, x_n]$. A *monomial* in R is a finite product of variables: $x^\alpha = x^{\alpha_1} \cdots x^{\alpha_n}$, $\alpha \in \mathbb{N}^n$. The *degree* of the monomial x^α is $\deg(x^\alpha) = \sum_{i=1}^n \alpha_i$, and the degree of a polynomial $f = \sum_\alpha c_\alpha x^\alpha$ is $\deg(f) = \max_{\{\alpha\,:\,c_\alpha \neq 0\}} \deg(x^\alpha)$. We define the vector subspaces

$$R_d \;=\; \{f \in R \,:\, \deg(f) \leq d\}, \quad d \in \mathbb{N}.$$

For an n-tuple of degrees $(d_1, \ldots, d_n)$, we define the family of polynomial systems

$$\mathcal{F}(d_1, \ldots, d_n) \;=\; R_{d_1} \times \cdots \times R_{d_n}.$$

That is, $F = (f_1, \ldots, f_n) \in \mathcal{F}(d_1, \ldots, d_n)$ satisfies $\deg(f_i) \leq d_i, i = 1, \ldots, n$, and represents the polynomial system $F = 0$ with $s = n$. We leave the fact that $\mathcal{F}(d_1, \ldots, d_n) \simeq K^D$, with $D = \sum_{i=1}^n \binom{n+d_i}{n}$, as an exercise to the reader. Note that this is a natural generalization of (10). The set of solutions of $F = 0$ is denoted by $V_K(F) = V_K(f_1, \ldots, f_n)$, and a point in $V_K(F)$ is *isolated* if it does not lie on a component of $V_K(F)$ with dimension ≥ 1.

Theorem 1 (Bézout)

For any $F = (f_1, \ldots, f_n) \in \mathcal{F}(d_1, \ldots, d_n)$, the number of isolated solutions of $f_1 = \cdots = f_n = 0$, i.e., the number of isolated points in $V_K(F)$, is at most $d_1 \cdots d_n$. Moreover, there exists a proper subvariety $\nabla_{d_1,\ldots,d_n} \subsetneq K^D$ such that, when $F \in \mathcal{F}(d_1, \ldots, d_n) \setminus \nabla_{d_1,\ldots,d_n}$, $V_K(F)$ consists of exactly $d_1 \cdots d_n$ isolated points.

The proof of this theorem can be found in [22, Theorem III-71]. As in our univariate example, the variety $\nabla_{d_1,\ldots,d_n}$ can be described using discriminants and resultants. See, for instance, the discussion at the end of [49, Section 3.4.1]. Theorem 1 is an important result and gives an easy way to bound the number of isolated solutions of a system of n equations in n variables. The bound is *almost always tight*, in the sense that the only systems with fewer solutions lie in $\nabla_{d_1,\ldots,d_n}$. Unfortunately, many systems coming from applications lie inside $\nabla_{d_1,\ldots,d_n}$. Here is an example.

Example: a planar robot arm

This example comes from robotics. Consider a planar robot arm whose shoulder is fixed at the origin $(0, 0)$ in the plane, and whose two arm segments have fixed lengths L_1 and L_2. We determine the possible positions of the elbow (x, y), given that the hand of the robot touches a given point (a, b). The situation is illustrated in Fig. 2. The Pythagorean theorem gives the identities

$$x^2 + y^2 - L_1^2 = (a - x)^2 + (b - y)^2 - L_2^2 = 0, \tag{11}$$

which is a system of $s = 2$ equations in $n = 2$ variables x, y. The plane curves corresponding to these equations are shown in Fig. 2. Their intersection points are the possible configurations. Naturally, more complicated robots lead to more involved equations, see [56]. The system (11) with $K = \mathbb{C}$ lies in $\nabla_{2,2}$: the two real solutions seen in Fig. 2 are the only solutions over $\mathbb{C}$, and $2 < d_1 \cdot d_2 = 4$. However, the slightest perturbation of the equations introduces two extra solutions. For instance, replace the first equation with $x^2 + \epsilon \cdot xy + y^2 - L_1^2 = 0$, for small ϵ. The resulting system lies in $\mathcal{F}(2, 2) \setminus \nabla_{2,2}$. It has four complex solutions, two of which lie close to the intersection points in Figure 2. The other two are *large*, see Remark 1.

Remark 1 Bézout's theorem more naturally counts solutions in *projective space* $\mathbb{P}^n_K$, and it accounts for solutions with *multiplicity* > 1. More precisely, if f_i is a homogeneous polynomial in $n + 1$ variables of degree d_i, and $f_1 = \cdots = f_n = 0$ has finitely many solutions in $\mathbb{P}^n_K$, the number of solutions (counted with multiplicity) is *always* $d_1 \cdots d_n$. We encourage the reader who is familiar with projective geometry to check that (11) defines two solutions *at infinity*, when each of the equations is viewed as an equation on $\mathbb{P}^2_{\mathbb{C}}$ by homogenizing. Introducing ϵ brings these solutions back into $\mathbb{C}^2$. Since they come from infinity, they have large coordinates.

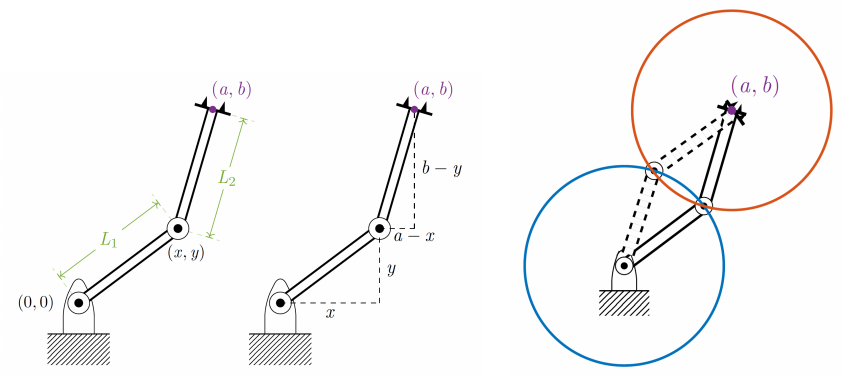


Fig. 2: The two configurations of a robot arm are the intersection points of two circles.

3.2 Kushnirenko's theorem

An intuitive consequence of Theorem 1 is that *random* polynomial systems given by polynomials of fixed degree always have the same number of solutions. Looking at f and g from (8), we see that they do not look so *random*, in the sense that some monomials of degree ≤ 3 are missing. For instance, x^3 and y^3 do not appear. Having zero coefficients standing with some monomials in $\mathcal{F}(d_1, \ldots, d_n)$ is sometimes enough to conclude that the system lies in $\nabla_{d_1,\ldots,d_n}$. That is, the system is not *random* in the sense of Bézout's theorem. The *(monomial) support* of $f = \sum_\alpha c_\alpha x^\alpha$ is

$$\operatorname{supp}\Big(\sum_\alpha c_\alpha x^\alpha\Big) \;=\; \{\alpha \;:\; c_\alpha \neq 0\} \;\subset\; \mathbb{N}^n.$$

This subsection considers families of polynomial systems with fixed support. Let $\mathcal{A} \subset \mathbb{N}^n$ be a finite subset of exponents of cardinality $|\mathcal{A}|$. We define

$$\mathcal{F}(\mathcal{A}) \;=\; \{\,(f_1, \ldots, f_n) \in R^n \;:\; \operatorname{supp}(f_i) \subset \mathcal{A},\, i = 1, \ldots, n\,\} \;\simeq\; K^{n\cdot|\mathcal{A}|}.$$

The next theorem expresses the number of solutions for systems in this family in terms of the volume $\operatorname{Vol}(\mathcal{A}) = \int_{\operatorname{Conv}(\mathcal{A})} \mathrm{d}\alpha_1 \cdots \mathrm{d}\alpha_n$ of the convex polytope

$$\operatorname{Conv}(\mathcal{A}) \;=\; \left\{ \sum_{\alpha\in\mathcal{A}} \lambda_\alpha \cdot \alpha \;:\; \lambda_\alpha \geq 0,\; \sum_{\alpha\in\mathcal{A}} \lambda_\alpha = 1 \right\} \;\subset \mathbb{R}^n. \tag{12}$$

The *normalized volume* $\operatorname{vol}(\mathcal{A})$ is defined as $n! \cdot \operatorname{Vol}(\mathcal{A})$.

Theorem 2 (Kushnirenko)

For any $F = (f_1, \ldots, f_n) \in \mathcal{F}(\mathcal{A})$, the number of isolated solutions of $f_1 = \cdots = f_n = 0$ in $(K \setminus \{0\})^n$, i.e., the number of isolated points in $V_K(F) \cap (K \setminus \{0\})^n$, is

at most $\mathrm{vol}(\mathcal{A})$. *Moreover, there exists a proper subvariety* $\nabla_{\mathcal{A}} \subsetneq K^{n\cdot|\mathcal{A}|}$ *such that, when* $F \in \mathcal{F}(\mathcal{A}) \setminus \nabla_{\mathcal{A}}$, $V_K(F) \cap (K \setminus \{0\})^n$ *consists of* $\mathrm{vol}(\mathcal{A})$ *isolated points.*

For a proof, see [33]. The theorem necessarily counts solutions in $(K\setminus\{0\})^n \subset K^n$, as multiplying all equations with a monomial x^α may change the number of solutions in the coordinate hyperplanes (i.e., there may be new solutions with zero-coordinates). However, it does not change the normalized volume $\mathrm{vol}(\mathcal{A})$. The statement can be adapted to count solutions in K^n, but becomes more involved [28]. We point out that, with the extra assumption that $0 \in \mathcal{A}$, one may replace $(K \setminus \{0\})^n$ by K^n in Theorem 2. To compare Kushnirenko's theorem with Bézout, note that $\mathcal{F}(\mathcal{A})$ for

$$\mathcal{A} = \{\alpha \in \mathbb{N}^n : \deg(x^\alpha) \leq d\} \tag{13}$$

is $\mathcal{F}(d, \ldots, d)$, and $d^n = \mathrm{vol}(\mathcal{A})$. Theorem 2 recovers Theorem 1 for $d_1 = \cdots = d_n$.

Example: back to plane curves

The polynomial system $f = g = 0$ from (8) belongs to the family $\mathcal{F}(\mathcal{A})$ with $\mathcal{A} = \{(1,0), (0,1), (2,0), (1,1), (0,2), (2,1), (1,2)\}$. The convex hull $\mathrm{Conv}(\mathcal{A})$ is a hexagon in $\mathbb{R}^2$, see Fig. 3. Its normalized volume is $\mathrm{vol}(\mathcal{A}) = n!\cdot\mathrm{Vol}(\mathcal{A}) = 2!\cdot 3 = 6$. Theorem 2 predicts six solutions in $(\overline{\mathbb{Q}} \setminus \{0\})^2$. These are six of the seven black dots seen in the left part of Fig. 1a: the solution $(0,0)$ is not counted. We have a chain of inclusions $\nabla_{\mathcal{A}} \subset \mathcal{F}(\mathcal{A}) \subset \nabla_{3,3} \subset \mathcal{F}(3,3)$ and $(f,g) \in \mathcal{F}(\mathcal{A}) \setminus \nabla_{\mathcal{A}}$.

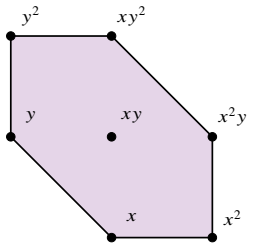


Fig. 3: The polytope $\mathrm{Conv}(\mathcal{A})$ in this example is a hexagon.

Remark 2 The analog of Remark 1 for Theorem 2 is that $\mathrm{vol}(\mathcal{A})$ counts solutions on the *projective toric variety* $X_{\mathcal{A}}$ associated with $\mathcal{A}$. It equals the degree of $X_{\mathcal{A}}$ in its embedding in $\mathbb{P}_K^{|\mathcal{A}|-1}$ (after multiplying with a lattice index). A proof and examples are given in [50, Section 3.4]. When $\mathcal{A}$ is as in (13), we have $X_{\mathcal{A}} = \mathbb{P}^n$.

Remark 3 The convex polytope $\mathrm{Conv}(\mathrm{supp}(f))$ is called the *Newton polytope* of f. Its importance goes beyond counting solutions: it is dual to the *tropical hypersurface* defined by f, which is a *combinatorial shadow* of $V_K(f)\cap(K\setminus\{0\})^n$ [36, Prop. 3.1.6].

3.3 Bernstein's theorem

There is a generalization of Kushnirenko's theorem which allows different supports for the polynomials $f_1, \ldots, f_n$. We fix n finite subsets of exponents $\mathcal{A}_1, \ldots, \mathcal{A}_n \subset \mathbb{N}^n$ with respective cardinalities $|\mathcal{A}_i|$. These define the family of polynomial systems

$$\mathcal{F}(\mathcal{A}_1, \ldots, \mathcal{A}_n) \;=\; \{\, (f_1, \ldots, f_n) \in R^n \;:\; \mathrm{supp}(f_i) \subset \mathcal{A}_i,\, i = 1, \ldots, n \,\} \;\simeq\; K^D,$$

where $D = |\mathcal{A}_1| + \cdots + |\mathcal{A}_n|$. The number of solutions is characterized by the *mixed volume* of $\mathcal{A}_1, \ldots, \mathcal{A}_n$, which we now define. The *Minkowski sum* $S + T$ of two sets $S, T \subset \mathbb{R}^n$ is $\{s + t \;:\; s \in S, t \in T\}$, where $s + t$ is the usual addition of vectors in $\mathbb{R}^n$. For a nonnegative real number λ, the *λ-dilation* of $S \subset \mathbb{R}^n$ is $\lambda \cdot S = \{\lambda \cdot s \;:\; s \in S\}$, where $\lambda \cdot s$ is the usual scalar multiplication in $\mathbb{R}^n$. Each of the supports $\mathcal{A}_i$ gives a convex polytope $\mathrm{Conv}(\mathcal{A}_i)$ as in (12). The function $\mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ given by

$$(\lambda_1, \ldots, \lambda_n) \longmapsto \mathrm{Vol}(\, \lambda_1 \cdot \mathrm{Conv}(\mathcal{A}_1) \,+\, \cdots \,+\, \lambda_n \cdot \mathrm{Conv}(\mathcal{A}_n)\,) \tag{14}$$

is a homogeneous polynomial of degree n, meaning that all its monomials have degree n [15, Chapter 7, §4, Proposition 4.9]. The *mixed volume* $\mathrm{MV}(\mathcal{A}_1, \ldots, \mathcal{A}_n)$ is the coefficient of the polynomial (14) standing with the monomial $\lambda_1 \cdots \lambda_n$.

Theorem 3 (Bernstein-Kushnirenko)

For any $F = (f_1, \ldots, f_n) \in \mathcal{F}(\mathcal{A}_1, \ldots, \mathcal{A}_n)$, the number of isolated solutions of $f_1 = \cdots = f_n = 0$ in $(K \setminus \{0\})^n$, i.e., the number of isolated points in $V_K(F) \cap (K \setminus \{0\})^n$, is at most $\mathrm{MV}(\mathcal{A}_1, \ldots, \mathcal{A}_n)$. Moreover, there exists a proper subvariety $\nabla_{\mathcal{A}_1,\ldots,\mathcal{A}_n} \subset K^D$ such that, when $F \in \mathcal{F}(\mathcal{A}_1, \ldots, \mathcal{A}_n) \setminus \nabla_{\mathcal{A}_1,\ldots,\mathcal{A}_n}$, $V_K(F) \cap (K \setminus \{0\})^n$ consists of precisely $\mathrm{MV}(\mathcal{A}_1, \ldots, \mathcal{A}_n)$ isolated points.

This theorem was originally proved by Bernstein for $K = \mathbb{C}$ in [6]. The proof by Kushnirenko in [33] works for algebraically closed fields. Several alternative proofs were found by Khovanskii [31]. Theorem 3 is sometimes called the *BKK theorem*, after the aforementioned mathematicians. Like Kushnirenko's theorem, Theorem 3 can be adapted to count roots in K^n rather than $(K \setminus \{0\})^n$ [28], and if $0 \in \mathcal{A}_i$ for all i, one may replace $(K \setminus \{0\})^n$ by K^n.

When $\mathcal{A}_1 = \cdots = \mathcal{A}_n = \mathcal{A}$, we have $\mathcal{F}(\mathcal{A}_1, \ldots, \mathcal{A}_n) = \mathcal{F}(\mathcal{A})$, and when $\mathcal{A}_i = \{\alpha \in \mathbb{N}^n : \deg(x^\alpha) \leq d_i\}$, we have $\mathcal{F}(\mathcal{A}_1, \ldots, \mathcal{A}_n) = \mathcal{F}(d_1, \ldots, d_n)$. Hence, all families of polynomials we have seen before are of this form, and Theorem 3 generalizes Theorems 1 and 2. Note that, in particular, we have $\mathrm{MV}(\mathcal{A}, \ldots, \mathcal{A}) = \mathrm{vol}(\mathcal{A})$.

Example: mixed areas

A useful formula for $n = 2$ is $\mathrm{MV}(\mathcal{A}_1, \mathcal{A}_2) = \mathrm{Vol}(\mathcal{A}_1 + \mathcal{A}_2) - \mathrm{Vol}(\mathcal{A}_1) - \mathrm{Vol}(\mathcal{A}_2)$. For instance, the following two polynomials appear in [49, Example 5.3.1]:

$$f = a_0 + a_1 x^3 y + a_2 x y^3, \quad g = b_0 + b_1 x^2 + b_2 y^2 + b_3 x^2 y^2.$$

The system $f = g = 0$ is a general member of the family $\mathcal{F}(\mathcal{A}_1, \mathcal{A}_2) \simeq K^7$, where $\mathcal{A}_1 = \{(0,0),(3,1),(1,3)\}$ and $\mathcal{A}_2 = \{(0,0),(2,0),(0,2),(2,2)\}$. The Newton polygons, together with their Minkowski sum, are shown in Fig. 4. By applying the

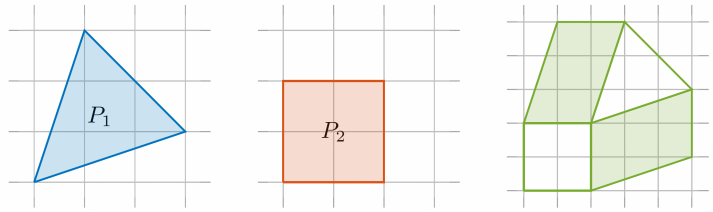


Fig. 4: The green area counts the solutions to equations with support in P_1, P_2.

formula for $\mathrm{MV}(\mathcal{A}_1, \mathcal{A}_2)$ seen above, we find that the mixed volume for the system $f = g = 0$ is the green area in the right part of Fig. 4, which is 12. Note that the Bézout bound (Theorem 1) is 16. and Theorem 2 also predicts 12 solutions, with $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$. Hence $\mathcal{F}(\mathcal{A}_1, \mathcal{A}_2) \subset \mathcal{F}(\mathcal{A}) \subset \nabla_{4,4} \subset \mathcal{F}(4,4)$ and $\mathcal{F}(\mathcal{A}_1, \mathcal{A}_2) \not\subset \nabla_{\mathcal{A}}$.

Theorem 3 provides an upper bound on the number of isolated solutions to any system of polynomial equations with $n = s$. Although it improves significantly on Bézout's bound for many systems, it still often happens that the bound is not tight for systems in applications. That is, one often encounters systems $F \in \nabla_{\mathcal{A}_1,\ldots,\mathcal{A}_n}$. Even more refined root counts exist, such as those based on *Newton-Okounkov bodies* [30]. In practice, with today's computational methods (see Section 4), we often count solutions reliably by simply solving the system. *Certification methods* provide a proof for a *lower* bound on the number of solutions [11]. The actual number of solutions is implied if one can match this with a theoretical upper bound.

4 Computational methods

We give a brief introduction to two of the most important computational methods for solving polynomial equations. The first method uses *normal forms*, the second is based on *homotopy continuation*. We keep writing $F = (f_1, \ldots, f_s) = 0$ for the system we want to solve. We require $s \geq n$, and assume finitely many solutions over $\overline{K}$. All methods discussed here compute all solutions over $\overline{K}$, *so we keep assuming that* $K = \overline{K}$ *is algebraically closed.* An important distinction between normal forms and homotopy continuation is that the former works over any field K, while the latter needs $K = \mathbb{C}$. If the coefficients are contained in a subfield (e.g. $\mathbb{R} \subset \mathbb{C}$), a significant part of the computation in normal form algorithms can be done over this subfield. Also, homotopy continuation is most natural when $n = s$, whereas $s > n$ is not

so much a problem for normal forms. However, if $K = \mathbb{C}$ and $n = s$, continuation methods are extremely efficient and can compute millions of solutions.

4.1 Normal form methods

Let $I = \langle f_1, \ldots, f_s \rangle \subset R = K[x_1, \ldots, x_n]$ be the ideal generated by our polynomials. For ease of exposition, we assume that I is *radical*, which is equivalent to all points in $V_K(I) = V_K(f_1, \ldots, f_s)$ having multiplicity one. In other words, the Jacobian matrix $(\partial f_i / \partial x_j)$, evaluated at any of the points in $V_K(I)$, has rank n. Let us write $V_K(I) = \{z_1, \ldots, z_\delta\} \subset K^n$ for the set of solutions, and R/I for the quotient ring obtained from R by the equivalence relation $f \sim g \Leftrightarrow f - g \in I$. The main observation behind normal form methods is that the coordinates of z_i are encoded in the eigenstructure of the K-linear endomorphisms $M_g : R/I \to R/I$ given by $[f] \mapsto [g \cdot f]$, where $[f]$ is the residue class of f in R/I.

We will now make this precise. First, we show that $\dim_K R/I = \delta$. We define $\mathrm{ev}_i : R/I \to K$ as $\mathrm{ev}_i([f]) = f(z_i)$, and combine these to get

$$\mathrm{ev} = (\,\mathrm{ev}_1, \ldots, \mathrm{ev}_\delta\,) : R/I \longrightarrow K^\delta, \quad \text{given by} \quad \mathrm{ev}([f]) = (\,f(z_1), \ldots, f(z_\delta)\,).$$

By Hilbert's Nullstellensatz [16, Chapter 4], a polynomial $f \in R$ belongs to I if and only if $f(z_i) = 0, i = 1, \ldots, \delta$. In other words, the map ev is injective. It is also surjective: there exist *Lagrange polynomials* $\ell_i \in R$ satisfying $\ell_i(z_j) = 1$ if $i = j$ and $\ell_i(z_j) = 0$ for $i \neq j$ [49, Lemma 3.1.2]. We conclude that $R/I \overset{\mathrm{ev}}{\simeq} K^\delta$.

The following statement makes our claim that *the zeros $z_1, \ldots, z_\delta$ are encoded in the eigenstructure of M_g* concrete.

Theorem 4 *The left eigenvectors of the K-linear map M_g are the evaluation functionals* $\mathrm{ev}_i, i = 1, \ldots, \delta$. *The eigenvalue corresponding to* ev_i *is* $g(z_i)$.

Proof We have $(\mathrm{ev}_i \circ M_g)([f]) = \mathrm{ev}_i([g \cdot f]) = g(z_i) f(z_i) = g(z_i) \cdot \mathrm{ev}_i([f])$, which shows that ev_i is a left eigenvector with eigenvalue $g(z_i)$. Moreover, the ev_i form a complete set of eigenvectors, since $\mathrm{ev} : R/I \to K^\delta$ is a K-linear isomorphism. □

We encourage the reader to check that the residue classes of the Lagrange polynomials $[\ell_i] \in R/I$ form a complete set of right eigenvectors. We point out that, after choosing a basis of R/I, the functional ev_i is represented by a row vector $w_i^\top$ of length δ, and M_g is a *multiplication matrix* of size $\delta \times \delta$. The eigenvalue relation in the proof of Theorem 4 reads more familiarly as $w_i^\top M_g = g(z_i) \cdot w_i^\top$. Theorem 4 suggests breaking up the task of computing $V_K(I) = \{z_i\}_{i=1}^\delta$ into two parts:

(A) Compute multiplication matrices M_g and
(B) extract the coordinates of z_i from their eigenvectors or eigenvalues.

For step (B), let $\{[b_1], \ldots, [b_\delta]\}$ be a K-basis for R/I, with $b_j \in R$. The vector w_i is explicitly given by $w_i = (b_1(z_i), \ldots, b_\delta(z_i))$. If the coordinate functions $x_1, \ldots, x_n$ are among the b_j, one reads the coordinates of z_i directly from the entries of w_i. If

not, some more processing might be needed. Alternatively, one can choose $g = x_j$ and read the j-th coordinates of the z_i from the eigenvalues of M_{x_j}. There are many things to say about these procedures, in particular about their efficiency and numerical stability. We refer the reader to [49, Remark 4.3.4] for references and more details, and do not elaborate on this here.

We turn to step (A), which is where *normal forms* come into play. Suppose a basis $\{[b_1], \dots, [b_\delta]\}$ of R/I is fixed. We identify R/I with $B = \operatorname{span}_K(b_1, \dots, b_\delta) \subset R$. For any $f \in R$, there are unique constants $c_j(f) \in K$ such that

$$f - \sum_{j=1}^{\delta} c_j(f) \cdot b_j \in I. \tag{15}$$

These are the coefficients in the unique expansion of $[f] = \sum_{j=1}^{\delta} c_j(f) \cdot [b_j]$ in our basis. The K-linear map $\mathcal{N} : R \to B$ which sends f to $\sum_{j=1}^{\delta} c_j(f) \cdot b_j$ is called a *normal form*. Its key property is that $\mathcal{N}$ *projects* R onto B *along* I, meaning that $\mathcal{N} \circ \mathcal{N} = \mathcal{N}$ ($\mathcal{N}_{|B}$ is the identity), and $\ker \mathcal{N} = I$. The multiplication map $M_g : B \to B$ is simply given by $M_g(b) = \mathcal{N}(g \cdot b)$. More concretely, the i-th column of the matrix representation of M_g contains the coefficients $c_j(g \cdot b_i)$, $j = 1, \dots, \delta$ of $\mathcal{N}(g \cdot b_i)$. Here is a familiar example.

Example: normal forms for $n = 1$

Let $I = \langle f \rangle = \langle a_0 + a_1 x + \cdots + a_d x^d \rangle$ be the ideal generated by the univariate polynomial $f \in K[x]$. For general a_i, there are $\delta = d$ roots with multiplicity one, hence I is radical. The dimension $\dim_K K[x]/I$ equals d, and a canonical choice of basis is $\{[1], [x], \dots, [x^{d-1}]\}$. Let us construct the matrix M_x in this basis. That is, we set $g = x$. We compute the normal forms $\mathcal{N}(x \cdot x^{i-1})$:

$$\mathcal{N}(x^i) = x^i,\ i = 1, \dots, d-1 \quad \text{and} \quad \mathcal{N}(x^d) = -a_d^{-1}(a_0 + a_1 x + \cdots + a_{d-1} x^{d-1}).$$

One checks this by verifying that $x^i - \mathcal{N}(x^i) \in \langle f \rangle$. The coefficients $c_j(x^i)$, $j = 1, \dots, d$ of $\mathcal{N}(x^i)$ form the i-th column of the companion matrix C_f in (7). Hence $M_x = C_f$, and Theorem 4 confirms that the eigenvalues of C_f are the roots of f.

Computing normal forms can be done using linear algebra on certain structured matrices, called *Macaulay matrices*. We illustrate this with an example from [52].

Example: Macaulay matrices

Consider the ideal $I = \langle f, g \rangle \subset \mathbb{Q}[x, y]$ given by $f = x^2 + y^2 - 2$, $g = 3x^2 - y^2 - 2$. The variety $V_{\overline{\mathbb{Q}}}(I) = V_{\mathbb{Q}}(I)$ consists of 4 points $\{(-1,-1), (-1,1), (1,-1), (1,1)\}$, as predicted by Theorem 1. We construct a *Macaulay matrix* whose rows are indexed by f, xf, yf, g, xg, yg, and whose columns are indexed by all monomials of degree ≤ 3:

$$\mathcal{M} = \begin{array}{c} \\ f \\ xf \\ yf \\ g \\ xg \\ yg \end{array} \begin{array}{c} \begin{array}{cccccc|cccc} x^3 & x^2y & xy^2 & y^3 & x^2 & y^2 & 1 & x & y & xy \end{array} \\ \left[\begin{array}{cccccc|cccc} & & & & 1 & 1 & -2 & & & \\ 1 & & 1 & & & & & -2 & & \\ & 1 & & 1 & & & & & -2 & \\ & & & & 3 & -1 & -2 & & & \\ 3 & & -1 & & & & & -2 & & \\ & 3 & & -1 & & & & & -2 & \end{array}\right] \end{array}.$$

The first row reads $f = 1 \cdot x^2 + 1 \cdot y^2 - 2 \cdot 1$. A basis for $\mathbb{Q}[x, y]/I$ is $\{[1], [x], [y], [xy]\}$. These monomials index the last four columns. We now invert the leftmost 6×6 block and apply this inverse from the left to $\mathcal{M}$:

$$\tilde{\mathcal{M}} = \begin{array}{c} \\ x^3 - x \\ x^2y - y \\ xy^2 - x \\ y^3 - y \\ x^2 - 1 \\ y^2 - 1 \end{array} \begin{array}{c} \begin{array}{cccccc|cccc} x^3 & x^2y & xy^2 & y^3 & x^2 & y^2 & 1 & x & y & xy \end{array} \\ \left[\begin{array}{cccccc|cccc} 1 & & & & & & & -1 & & \\ & 1 & & & & & & & -1 & \\ & & 1 & & & & & -1 & & \\ & & & 1 & & & & & -1 & \\ & & & & 1 & & -1 & & & \\ & & & & & 1 & -1 & & & \end{array}\right] \end{array}.$$

The rows of $\tilde{\mathcal{M}}$ are linear combinations of the rows of $\mathcal{M}$, representing polynomials in I. The first row reads $x^3 - 1 \cdot x \in I$. Comparing this with (15), we see that we have found that $\mathcal{N}(x^3) = x$. Using $\tilde{\mathcal{M}}$ we can construct M_x and M_y:

$$M_x = \begin{array}{c} \\ [1] \\ [x] \\ [y] \\ [xy] \end{array} \begin{array}{c} \begin{array}{cccc} [x] & [x^2] & [xy] & [x^2y] \end{array} \\ \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right] \end{array}, \quad M_y = \begin{array}{c} \\ [1] \\ [x] \\ [y] \\ [xy] \end{array} \begin{array}{c} \begin{array}{cccc} [y] & [xy] & [y^2] & [xy^2] \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \end{array}.$$

The reader is encouraged to verify Theorem 4 for these matrices.

Remark 4 The entries of a Macaulay matrix $\mathcal{M}$ are the coefficients of the polynomials $f_1, \ldots, f_s$. An immediate consequence of the fact that normal forms are computed using linear algebra on Macaulay matrices is that when the coefficients of f_i are contained in a subfield $\tilde{K} \subset K$, all computations in step (A) can be done over $\tilde{K}$. This assumes the polynomials g for which we want to compute M_g have coefficients in $\tilde{K}$.

As illustrated in the example above, to compute the matrices M_g it is sufficient to determine the restriction of the normal form $\mathcal{N} : R \to B$ to a finite-dimensional K-vector space $V \subset R$, containing $g \cdot B$. The restriction $\mathcal{N}_{|V} : V \to B$ is called a *truncated normal form*, see [51] and [49, Chapter 4]. The dimension of the space V counts the number of columns of the Macaulay matrix.

Usually, one chooses the basis elements b_j of B to be monomials, and g to be a coordinate function x_i. The basis elements may arise as standard monomials from a *Gröbner basis* computation. We briefly discuss this important concept.

> Gröbner bases

Gröbner bases are powerful tools for symbolic computation in algebraic geometry. A nice way to motivate their definition is by considering *Euclidean division* as a candidate for a normal form map. In the case $n = 1$, this rewrites $g \in K[x]$ as

$$g = q \cdot f + r, \quad \text{where} \quad \deg(r) < d. \tag{16}$$

Clearly $[g] = [r]$ in $K[x]/\langle f \rangle$, and $\mathcal{N}(g) = r$ since $r \in B = \text{span}_K(1, x, \ldots, x^{d-1})$.

To generalize this to $n > 1$ variables, we fix a *monomial order* $\preceq$ on $R = K[x_1, \ldots, x_n]$ and write $\text{LT}(f)$ for the *leading term* of f with respect to $\preceq$. The reader who is unfamiliar with monomial orders can consult [16, Chapter 2, §2]. As above, let $I = \langle f_1, \ldots, f_s \rangle \subset R$ be a radical ideal such that $|V_K(I)| = \delta < \infty$. As basis elements $b_1, \ldots, b_\delta$ of $B \simeq R/I$, we use the δ $\preceq$–smallest monomials which are linearly independent modulo our ideal I. They are also called *standard monomials*. By [16, Chapter 9, §3, Theorem 3], there exists an algorithm which, for any input $g \in R$, computes $q_1, \ldots, q_s, r \in R$ such that

$$g = q_1 \cdot f_1 + \cdots + q_s \cdot f_s + r, \quad \text{where } \text{LT}(f_i) \text{ does not divide any term of } r, \forall i. \tag{17}$$

This algorithm is called *multivariate Euclidean division*. Note how the condition "$\text{LT}(f_i)$ does not divide any term of r, for all i" generalizes $\deg(r) < d$ in (16). From (17), it is clear that $[g] = [r]$. However, we do *not* have $r \in B$ in general. Hence, unfortunately, sending g to its remainder r is usually not a normal form. . . but it is when $f_1, \ldots, f_s$ is a Gröbner basis!

A set of polynomials $g_1, \ldots, g_k \in I$ forms a *Gröbner basis* of the ideal I if the leading terms $\text{LT}(g_1), \ldots, \text{LT}(g_k)$ generate the *leading term ideal* $\langle \text{LT}(g) \,:\, g \in I \rangle$. We point out that no finiteness of $V_K(I)$ or radicality of I is required for this definition. The remainder r in $g = q_1 \cdot g_1 + \cdots + q_k \cdot g_k + r$ where $\text{LT}(f_i)$ does not divide any term of r, for all i, now satisfies $[g] = [r]$ *and* $r \in B$. This justifies the following claim:

Taking remainder upon Euclidean division by a Gröbner basis is a normal form.

Computing a Gröbner basis $g_1, \ldots, g_k$ from a set of input polynomials $f_1, \ldots, f_s$ can be interpreted as Gaussian elimination on a Macaulay matrix [24]. Once this has been done, multiplication matrices are computed via taking remainder upon Euclidean division by $\{g_1, \ldots, g_k\}$.

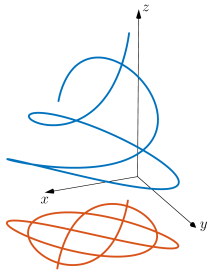
On a sidenote, we point out that Gröbner bases are often used for the *elimination of variables*. For instance, if $g_1, \ldots, g_k$ form a Gröbner basis of an ideal I with respect to a *lex* monomial order for which $x_1 \prec x_2 \prec \cdots \prec x_n$, we have for $j = 1, \ldots n$ that the *j-th elimination ideal*

$$I_j \,=\, I \cap K[x_1, \ldots, x_j] \,=\, \langle g_i \,:\, g_i \in K[x_1, \ldots, x_j] \rangle$$

is generated by those elements of our Gröbner basis which involve only the first j variables, see [16, Chapter 3, §1, Theorem 2]. In our case, a consequence is that one of the g_i is univariate in x_1, and its roots are the x_1-coordinates of $z_1, \ldots, z_\delta$. The geometric counterpart of computing the j-th elimination ideal is *projection* onto a j-dimensional coordinate space: the variety $V_K(I_j) \subset K^j$ is obtained from $V_K(I) \subset K^n$ by forgetting the final $n-j$ coordinates $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_j)$ and taking the closure of the image. Here are two examples.

Example: the projection of a space curve

To the right we show a blue curve in $\mathbb{R}^3$ defined by an ideal $I \subset \mathbb{R}[x, y, z]$. Its Gröbner basis with respect to the lex ordering $x \prec y \prec z$ contains $g_1 \in \mathbb{R}[x, y]$, which generates I_2. The variety $V_{\mathbb{R}}(I_2) = V_{\mathbb{R}}(g_1) \subset \mathbb{R}^2$ is the orange curve in the picture.



Example: smooth del Pezzo surfaces

In [39], the authors study del Pezzo surfaces of degree 4 in $\mathbb{P}^4$ with defining equations $x_0x_1 - x_2x_3 = a_0x_0^2 + a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 = 0$. We will substitute $x_4 = 1 - x_0 - x_1 - x_2 - x_3$ to reduce to the affine case. It is claimed that the smooth del Pezzo surfaces of this form are those for which the parameters $a_0, \ldots, a_4$ lie outside the hypersurface $H = \{a_0a_1a_2a_3a_4(a_0a_1 - a_2a_3) = 0\}$. This hypersurface is the projection of the variety

$$\left\{(a, x) \in \mathbb{Q}^5 \times \mathbb{Q}^4 \,:\, x_0x_1 - x_2x_3 = \sum_{i=0}^{3} a_i x_i^2 + a_4\big(1 - \sum_{i=0}^{3} x_i\big)^2 = 0 \text{ and } \operatorname{rank}(J) < 2\right\}$$

onto $\mathbb{Q}^5$. Here J is the 2×4 Jacobian matrix of our two equations with respect to the four variables x_0, x_1, x_2, x_3. The defining equation of H is computed in `Macaulay2` [25] as follows:

```
R = QQ[x_0..x_3,a_0..a_4]
x_4 = 1-x_0-x_1-x_2-x_3
I = ideal( x_0*x_1-x_2*x_3 , a_0*x_0^2 + a_1*x_1^2 + ... + a_4*x_4^2 )
M = submatrix( transpose jacobian I , 0..3 )
radical eliminate( I+minors(2,M) , {x_0,x_1,x_2,x_3} )
```

The work behind the final command is a Gröbner basis computation.

Remark 5 In a numerical setting, it is better to use *border bases* or more general bases to avoid amplifying rounding errors. Border bases use basis elements b_i for B whose elements satisfy a connectedness property. See, for instance, [40] for details. They do not depend on a monomial order. For a summary and comparison between Gröbner bases and border bases, see [49, Sections 3.3.1, 3.3.2]. Nowadays, bases are selected adaptively by numerical linear algebra routines, such as QR decomposition with optimal column pivoting or singular value decomposition. This often yields a significant improvement in terms of accuracy. See, for instance, Section 7.2 in [52].

4.2 Homotopy Continuation

The goal of this subsection is to briefly introduce the method of homotopy continuation for solving polynomial systems. For more details, we refer to the textbook [46].

We set $K = \mathbb{C}$ and $n = s$. We think of $F = (f_1, \ldots, f_n) \in R$ as an element of a family $\mathcal{F}$ of polynomial systems. The reader can replace $\mathcal{F}$ with any of the families seen in Section 3. A *homotopy* in $\mathcal{F}$ with *target system* $F \in \mathcal{F}$ and *start system* $G \in \mathcal{F}$ is a continuous deformation of the map $G = (g_1, \ldots, g_n) : \mathbb{C}^n \to \mathbb{C}^n$ into F, in such a way that all systems obtained throughout the deformation are contained in $\mathcal{F}$. For instance, When $F \in \mathcal{F}(d_1, \ldots, d_n)$ as in Section 3.1 and G is any other system in $\mathcal{F}(d_1, \ldots, d_n)$, a homotopy is $H(x;t) = t \cdot F(x) + (1-t) \cdot G(x)$, where t runs from 0 to 1. Indeed, for any fixed $t^* \in [0,1]$, the degrees of the equations remain bounded by $(d_1, \ldots, d_n)$, hence $H(x;t^*) \in \mathcal{F}(d_1, \ldots, d_n)$.

The method of homotopy continuation for solving the target system $f_1 = \cdots = f_n = 0$ assumes that a start system $g_1 = \cdots = g_n = 0$ can easily be solved. The idea is that transforming G continuously into F via a homotopy $H(x;t)$ in $\mathcal{F}$ transforms the solutions of G continuously into those of F. Here is an example with $n = 1$.

Example: $n = s = 1, \mathcal{F} = \mathcal{F}(3)$

Let $f = -6 + 11x - 6x^2 + x^3 = (x-1)(x-2)(x-3)$ be the *Wilkinson polynomial* of degree 3. We view $f = 0$ as a member of $\mathcal{F}(3)$ and choose the start system $g = x^3 - 1 = 0$. The solutions of $g = 0$, are the third roots of unity. The solutions of $H(x;t) = \gamma \cdot t \cdot f(x) + (1-t) \cdot g(x)$ travel from these roots to the integers $1, 2, 3$ as t moves from 0 to 1. This is illustrated in Fig. 5. The random complex constant γ is needed to *avoid the discriminant*, see below. This is known as the *gamma trick*.

More formally, if $H(x;t) = (h_1(x;t), \ldots, h_n(x;t))$ is a homotopy with $t \in [0,1]$, the solutions describe continuous paths $x(t)$ satisfying $H(x(t);t) = 0$. Taking the derivative with respect to t gives the *Davidenko differential equation*

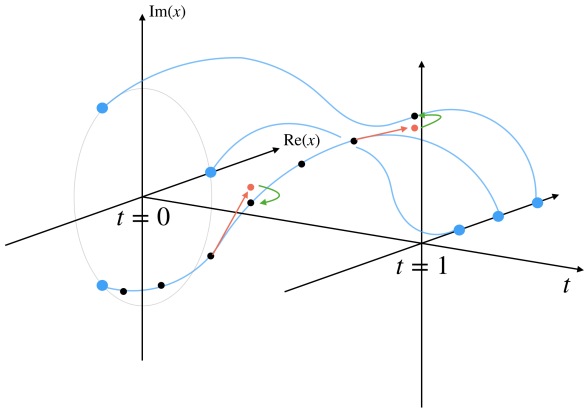


Fig. 5: The third roots of unity travel to $1, 2, 3$ along continuous paths.

$$\frac{\mathrm{d}H(x(t),t)}{\mathrm{d}t} = J_x \cdot \dot{x}(t) + \frac{\partial H}{\partial t}(x(t),t) = 0, \quad \text{with } J_x = \left(\frac{\partial h_i}{\partial x_j}\right)_{i,j}. \tag{18}$$

Each start solution x^* of $g_1 = \cdots = g_n = 0$ gives an initial value problem with $x(0) = x^*$, and the corresponding solution path $x(t)$ can be approximated using any numerical ODE method. This leads to a discretization of the solution path, see the black dots in Fig. 5. The solutions of $f_1 = \cdots = f_n = 0$ are obtained by evaluating the solution paths at $t = 1$. The following are important practical remarks.

> Predict and correct

Naively applying ODE methods for solving the Davidenko equation (18) is not the best we can do. Indeed, we have the extra information that the solution paths $x(t)$ satisfy the implicit equation $H(x(t), t) = 0$. This is used to improve the accuracy of the ODE solver in each step. Given an approximation of $x(t^*)$ at any fixed $t^* \in [0, 1)$ and a step size $0 < \Delta t \ll 1$, one approximates $x(t^* + \Delta t)$ by $\tilde{x}$ using, for instance, Euler's method. This is called the *predictor* step. Then, one refines $\tilde{x}$ to a satisfactory approximation of $x(t^* + \Delta t)$ by using $\tilde{x}$ as a starting point for Newton iteration on $H(x, t^* + \Delta t) = 0$. This is the *corrector step*. The two-step process is illustrated in Fig. 5 (*predict* in orange, *correct* in green, solution paths $x(t)$ in blue). In the right part of the figure, the predict-correct procedure fails: because of a too-large step size Δt in the predictor step, the Newton correction converges to a *different* path. This phenomenon is called *path jumping*, and to avoid it one must choose the stepsize Δt adaptively. Recent work in this direction uses Padé approximants, see [53, 54].

> Avoid the discriminant

Each of the families seen in Section 3 has a subvariety $\nabla_{\mathcal{F}} \subset \mathcal{F}$ consisting of systems with non-generic behavior in terms of their number of solutions. This subvariety is sometimes referred to as the *discriminant* of $\mathcal{F}$, for reasons alluded to at the beginning of Section 3. When the homotopy $H(x;t)$ crosses $\nabla_{\mathcal{F}}$, i.e. $H(x;t^*) \in \nabla_{\mathcal{F}}$ for some $t^* \in [0,1)$, two or more solution paths collide at t^*, or some solution paths diverge. This is not allowed for the numerical solution of (18). Fortunately, crossing $\nabla_{\mathcal{F}}$ can be avoided. The discriminant $\nabla_{\mathcal{F}}$ has complex codimension at least one, hence *real* codimension at least two. Since the homotopy $t \mapsto H(x;t)$ describes a one-real-dimensional path in $\mathcal{F}$, it is always possible to go *around* the discriminant. See for instance [46, Section 7]. When the target system F belongs to the discriminant, *end games* are used to deal with colliding/diverging paths at $t = 1$ [46, Section 10]. This story implies that the number of paths tracked in a homotopy algorithm is the generic number of solutions of the family $\mathcal{F}$. In that sense, results like Theorems 1, 2 and 3 characterize the complexity of homotopy continuation in the respective families.

> Start systems in practice

There are recipes for start systems in the families $\mathcal{F}$ from Section 3. For instance, we use $G = (x_1^{d_1}-1, \ldots, x_n^{d_n}-1)$ for $\mathcal{F}(d_1, \ldots, d_n)$. The $d_1 \cdots d_n$ solutions can easily be written down. Note that $G \notin \nabla_{d_1,\ldots,d_n}$. For the families $\mathcal{F}(\mathcal{A})$ and $\mathcal{F}(\mathcal{A}_1, \ldots, \mathcal{A}_n)$, an algorithm to solve start systems was developed in [27]. For solving start systems of other families, one may use *monodromy loops* [21].

5 Case study: 27 lines on the Clebsch surface

A classical result from intersection theory states that every smooth cubic surface in complex three-space contains exactly 27 lines. In this final section, we use Gröbner bases and homotopy continuation to compute lines on the Clebsch surface defined by (9). This particular surface is famous for the fact that all its 27 lines are real. Let $f(x,y,z)$ be as in (9). A line in $\mathbb{R}^3$ parameterized by $(a_1+t\cdot b_1, a_2+t\cdot b_2, a_3+t\cdot b_3)$ is contained in our Clebsch surface if and only if $f(a_1+t\cdot b_1, a_2+t\cdot b_2, a_3+t\cdot b_3) \equiv 0$. The left-hand side evaluates to a cubic polynomial in t with coefficients in the ring $\mathbb{Z}[a_1,a_2,a_3,b_1,b_2,b_3] = \mathbb{Z}[a,b]$:

$$f\,(\,a_1+t\cdot b_1, a_2+t\cdot b_2, a_3+t\cdot b_3\,) \;=\; f_1(a,b)\cdot t^3 + f_2(a,b)\cdot t^2 + f_3(a,b)\cdot t + f_4(a,b).$$

The lines contained in the Clebsch surface satisfy

$$f_1(a,b) \;=\; f_2(a,b) \;=\; f_3(a,b) \;=\; f_4(a,b) \;=\; 0. \tag{19}$$

We further reduce this to a system of $s = 4$ equations in $n = 4$ unknowns by removing the redundancy in our parameterization of the line: the space of lines in three-space (i.e. the Grassmannian $\mathbb{G}(1,3)$) has dimension four, not six. We may impose a random affine-linear relation among the a_i and b_i. We choose to substitute

$$a_3 = -(7 + a_1 + 3a_2)/5, \quad b_3 = -(11 + 3b_1 + 5b_2)/7.$$

Implementations of Gröbner bases are available, for instance, in `Maple` [37] and in `msolve` [7], which can be used in `julia` via the package `msolve.jl`. This is also available in the package `Oscar.jl` [43]. The following snippet of `Maple` code constructs our system (19) and computes a Gröbner basis with respect to the graded lexicographic monomial ordering with $a_1 \succ a_2 \succ b_1 \succ b_2$. This basis consists of 23 polynomials $g_1, \ldots, g_{23}$.

```
> f := 81*(x^3 + y^3 + z^3) - 189*(x^2*y + x^2*z + x*y^2 + x*z^2 + y^2*z + y*z^2)
       + 54*x*y*z + 126*(x*y + x*z + y*z) - 9*(x^2 + y^2 + z^2) - 9*(x + y + z) + 1:
> f := expand(subs({x = t*b[1] + a[1], y = t*b[2] + a[2], z = t*b[3] + a[3]}, f)):
> f := subs({a[3] = -(7 + a[1] + 3*a[2])/5, b[3] = -(11 + 3*b[1] + 5*b[2])/7}, f):
> ff := coeffs(f, t):
> with(Groebner):
> GB := Basis({ff}, grlex(a[1], a[2], b[1], b[2]));
> nops(GB);                                                    ----> output: 23
```

The set of standard monomials is the first output of the command `NormalSet`. It consists of 27 elements, and the multiplication matrix with respect to a_1 in this basis is constructed using `MultiplicationMatrix`:

```
> ns, rv := NormalSet(GB, grlex(a[1], a[2], b[1], b[2])):
> nops(ns);                                                    ----> output: 27
> Ma1 := MultiplicationMatrix(a[1], ns, rv, GB, grlex(a[1], a[2], b[1], b[2])):
```

This is a matrix of size 27×27 whose eigenvectors reveal the solutions (Theorem 4). We now turn to `julia` and use `msolve` to compute the 27 lines on $\{f = 0\}$ as follows:

```
using Oscar
R,(a1,a2,b1,b2) = PolynomialRing(QQ,["a1","a2","b1","b2"])
I = ideal(R, [-189*b2*b1^2 - 189*b2^2*b1 + 27*(11 + 3*b1 + 5*b2)*b1^2 + ...
A, B = msolve(I)
```

The output `B` contains 4 rational coordinates (a_1, a_2, b_1, b_2) of 27 lines which approximate the solutions. To see them in floating point format, use for instance

```
[convert.(Float64,convert.(Rational{BigInt},b)) for b in B]
```

We have drawn three of these lines on the Clebsch surface in Fig. 6 as an illustration. Other software systems supporting Gröbner bases are `Macaulay2` [25], `Magma` [8], `Mathematica` [**?**] and `Singular` [26].

Homotopy continuation methods provide an alternative way to compute our 27 lines. Here we use the `julia` package `HomotopyContinuation.jl` [12].

```
using HomotopyContinuation
@var x y z t a[1:3] b[1:3]
f = 81*(x^3 + y^3 + z^3) - 189*(x^2*y + x^2*z + x*y^2 + x*z^2 + y^2*z + y*z^2)
    + 54*x*y*z + 126*(x*y + x*z + y*z) - 9*(x^2 + y^2 + z^2) - 9*(x + y + z) + 1
fab = subs(f, [x;y;z] => a+t*b)
```

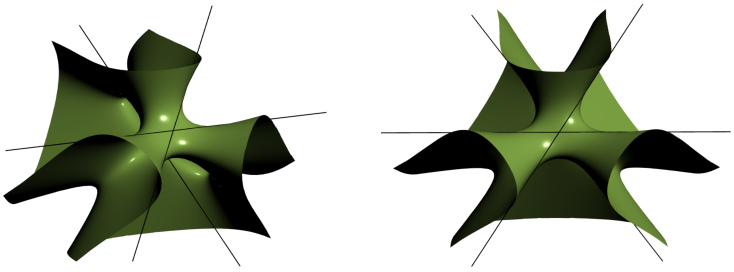


Fig. 6: Two views on the Clebsch surface with three of its 27 lines.

```
E, C = exponents_coefficients(fab,[t])
F = subs(C,[a[3];b[3]] => [-(7+a[1]+3*a[2])/5; -(11+3*b[1]+5*b[2])/7])
R = solve(F)
```

The output is shown in Fig. 7. There are 27 solutions, as expected. The last line indicates that a `:polyhedral` start system was used. In our terminology, this means that the system was solved using a homotopy in the family $\mathcal{F}(\mathcal{A}_1,\dots,\mathcal{A}_4)$ from Section 3.3. The number of tracked paths is 45, which is the mixed volume $\mathrm{MV}(\mathcal{A}_1,\dots,\mathcal{A}_4)$ of this family. The discrepancy $27 < 45$ means that our system F lies in the discriminant $\nabla_{\mathcal{A}_1,\dots,\mathcal{A}_4}$. The 18 'missing' solutions are explained in [4, Section 3.3]. The output also tells us that all solutions have multiplicity one (this is the meaning of `non-singular`) and all of them are real.

```
Result with 27 solutions
========================
• 45 paths tracked
• 27 non-singular solutions (27 real)
• random_seed: 0xb8b6077c
• start_system: :polyhedral
```

Fig. 7: The `julia` output when computing 27 lines on the Clebsch surface.

Other software implementing homotopy continuation are `Bertini` [2] and `PHCpack` [55]. Numerical normal form methods are used in `EigenvalueSolver.jl` [5].

References

1. L. Baldi and B. Mourrain. Computing real radicals by moment optimization. In *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation*, pages 43–50, 2021.
2. D. J. Bates, A. J. Sommese, J. D. Hauenstein, and C. W. Wampler. *Numerically solving polynomial systems with Bertini*. SIAM, 2013.
3. K. Batselier. *A numerical linear algebra framework for solving problems with multivariate polynomials*. PhD thesis, Faculty of Engineering, KU Leuven (Leuven, Belgium), 2013.
4. M. R. Bender and S. Telen. Toric eigenvalue methods for solving sparse polynomial systems. *Mathematics of Computation*, 91(337):2397–2429, 2022.

5. M. R. Bender and S. Telen. Yet another eigenvalue algorithm for solving polynomial systems. *arXiv preprint arXiv:2105.08472*, 2021.
6. D. N. Bernstein. The number of roots of a system of equations. *Functional Analysis and its applications*, 9(3):183–185, 1975.
7. J. Berthomieu, C. Eder, and M. Safey El Din. msolve: A library for solving polynomial systems. In *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*, pages 51–58, 2021.
8. W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).
9. P. Breiding. An algebraic geometry perspective on topological data analysis. *arXiv preprint arXiv:2001.02098*, 2020.
10. P. Breiding, T. Ö. Çelik, T. Duff, A. Heaton, A. Maraj, A.-L. Sattelberger, L. Venturello, and O. Yürük. Nonlinear algebra and applications. *arXiv preprint arXiv:2103.16300*, 2021.
11. P. Breiding, K. Rose, and S. Timme. Certifying zeros of polynomial systems using interval arithmetic. *arXiv preprint arXiv:2011.05000*, 2020.
12. P. Breiding and S. Timme. HomotopyContinuation.jl: A package for homotopy continuation in Julia. In *International Congress on Mathematical Software*, pages 458–465. Springer, 2018.
13. C. Conradi, E. Feliu, M. Mincheva, and C. Wiuf. Identifying parameter regions for multistationarity. *PLoS computational biology*, 13(10):e1005751, 2017.
14. D. A. Cox. *Applications of polynomial systems*, volume 134. American Mathematical Soc., 2020.
15. D. A. Cox, J. B. Little, and D. O'Shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.
16. D. A. Cox, J. B. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, corrected fourth edition, 2018.
17. H. Darmon, F. Diamond, and R. Taylor. Fermat's last theorem. *Current developments in mathematics*, 1995(1):1–154, 1995.
18. J. Desjardins and R. Winter. Density of rational points on a family of del Pezzo surfaces of degree one. *Advances in Mathematics*, 405:108489, 2022.
19. A. Dickenstein. Biochemical reaction networks: An invitation for algebraic geometers. In *Mathematical Congress of the Americas*, volume 656, pages 65–83. Contemp. Math, 2016.
20. J. Draisma, E. Horobeţ, G. Ottaviani, B. Sturmfels, and R. R. Thomas. The euclidean distance degree of an algebraic variety. *Foundations of computational mathematics*, 16(1):99–149, 2016.
21. T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 39(3):1421–1446, 2019.
22. D. Eisenbud and J. Harris. *The geometry of schemes*, volume 197. Springer Science & Business Media, 2006.
23. I. Z. Emiris and B. Mourrain. Computer algebra methods for studying and computing molecular conformations. *Algorithmica*, 25(2):372–402, 1999.
24. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.
25. D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.
26. G.-M. Greuel, G. Pfister, and H. Schönemann. Singular—a computer algebra system for polynomial computations. In *Symbolic computation and automated reasoning*, pages 227–233. AK Peters/CRC Press, 2001.
27. B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Mathematics of computation*, 64(212):1541–1555, 1995.
28. B. Huber and B. Sturmfels. Bernstein's theorem in affine space. *Discrete & Computational Geometry*, 17(2):137–141, 1997.

29. E. Hubert and E. Rodriguez Bazan. Algorithms for fundamental invariants and equivariants. *Mathematics of Computation*, 91(337):2459–2488, 2022.
30. K. Kaveh and A. G. Khovanskii. Newton-Okounkov bodies, semigroups of integral points, graded algebras and intersection theory. *Annals of Mathematics*, pages 925–978, 2012.
31. A. G. Khovanskii. Newton polyhedra and the genus of complete intersections. *Functional Analysis and its applications*, 12(1):38–46, 1978.
32. Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision*, pages 302–315. Springer, 2008.
33. A. G. Kushnirenko. Newton polytopes and the Bézout theorem. *Functional analysis and its applications*, 10(3):233–235, 1976.
34. J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.
35. M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.
36. D. Maclagan and B. Sturmfels. *Introduction to tropical geometry*, volume 161. American Mathematical Soc., 2021.
37. Maplesoft, a division of Waterloo Maple Inc.. Maple.
38. M. Michałek and B. Sturmfels. *Invitation to nonlinear algebra*, volume 211. American Mathematical Soc., 2021.
39. V. Mitankin and C. Salgado. Rational points on del Pezzo surfaces of degree four. *arXiv preprint arXiv:2002.11539*, 2020.
40. B. Mourrain. A new criterion for normal form algorithms. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 430–442, 1999.
41. B. Mourrain and J. P. Pavone. Subdivision methods for solving polynomial equations. *Journal of Symbolic Computation*, 44(3):292–306, 2009.
42. S. Müller, E. Feliu, G. Regensburger, C. Conradi, A. Shiu, and A. Dickenstein. Sign conditions for injectivity of generalized polynomial maps with applications to chemical reaction networks and real algebraic geometry. *Foundations of Computational Mathematics*, 16(1):69–97, 2016.
43. Oscar – open source computer algebra research system, version 0.9.0, 2022.
44. M. Sala. Gröbner bases, coding, and cryptography: a guide to the state-of-art. In *Gröbner Bases, Coding, and Cryptography*, pages 1–8. Springer, 2009.
45. A. J. Sommese, J. Verschelde, and C. W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM Journal on Numerical Analysis*, 38(6):2022–2046, 2001.
46. A. J. Sommese, C. W. Wampler, et al. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005.
47. B. Sturmfels. *Solving systems of polynomial equations*. Number 97. American Mathematical Soc., 2002.
48. B. Sturmfels. Beyond linear algebra. *arXiv preprint arXiv:2108.09494*, 2021.
49. S. Telen. *Solving Systems of Polynomial Equations*. PhD thesis, KU Leuven, Leuven, Belgium, 2020. Available at https://simontelen.webnode.page/publications/.
50. S. Telen. Introduction to toric geometry. *arXiv preprint arXiv:2203.01690*, 2022.
51. S. Telen, B. Mourrain, and M. Van Barel. Solving polynomial systems via truncated normal forms. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1421–1447, 2018.
52. 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.
53. S. Telen, M. Van Barel, and J. Verschelde. A robust numerical path tracking algorithm for polynomial homotopy continuation. *SIAM Journal on Scientific Computing*, 42(6):A3610–A3637, 2020.
54. S. Timme. Mixed precision path tracking for polynomial homotopy continuation. *Advances in Computational Mathematics*, 47(5):1–23, 2021.
55. 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.
56. C. W. Wampler and A. J. Sommese. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica*, 20:469–567, 2011.