Constructive geometry / root iterations / systems of polynomial equations

The simultaneous rectangle for systems, without homotopy

All the solutions of a polynomial system, each in a certified ball, with a proof that none is missing: sweeps of the rectangle step, the solutions computed from one another by the law of the residues, and a count by mixed cells that one lifting proves

Ivan Besevic[1] · Independent research
Version S2, revised for the version 0.21.0 of the program (it replaces the version S1; companion of *The simultaneous rectangle*, G8 revision 9) · 17 November 2026

Abstract. The note [9] carried the simultaneous rectangle of [8] from one polynomial to a system of n polynomial equations in n unknowns, and concluded that the step and the certificate transpose while the global work had to be left to a homotopy. This note removes the homotopy. The points start on the solutions of a start system and are swept on F itself with the rectangle step of order three, each point deflated of its nearest neighbours; nothing is tracked. What the sweeps do not find is obtained in three ways that have no counterpart in a path method: (1) the images of the solutions found by the symmetries read on the system; (2) runs of the step of one point from random points, by batches; (3) *the law of the residues*: by the formula of Euler and Jacobi the solutions found determine the moments of the missing ones, which are then the common eigenvalues of small matrices (Proposition 4.2). Every point, wherever it comes from, passes one certificate, and the output is complete when as many disjoint balls are certified as a bound of the number of solutions. For sparse systems that bound is the mixed volume, computed by mixed cells; we show that *one* lifting, whatever it is, gives the mixed volume as soon as an exact check meets no tie (Theorem 5.4), so that the count is a proof and not a probability. On 40 000 systems of twenty-seven families the implementation returns what is expected on all of them and proves completeness on 39 998. On eighteen classical or large systems, up to 6561 solutions and 64 unknowns, it proves completeness on every one, and is faster than a reference homotopy solver with certification on fourteen of them, by a factor of 2 to 25, level on two and slower on two, by a factor of 1.5 (Figure 6). The algebra of the step, the certificate, the laws used by the residues and the argument of the lifting are checked in Lean 4: 64 theorems, no `sorry` (Section 9).

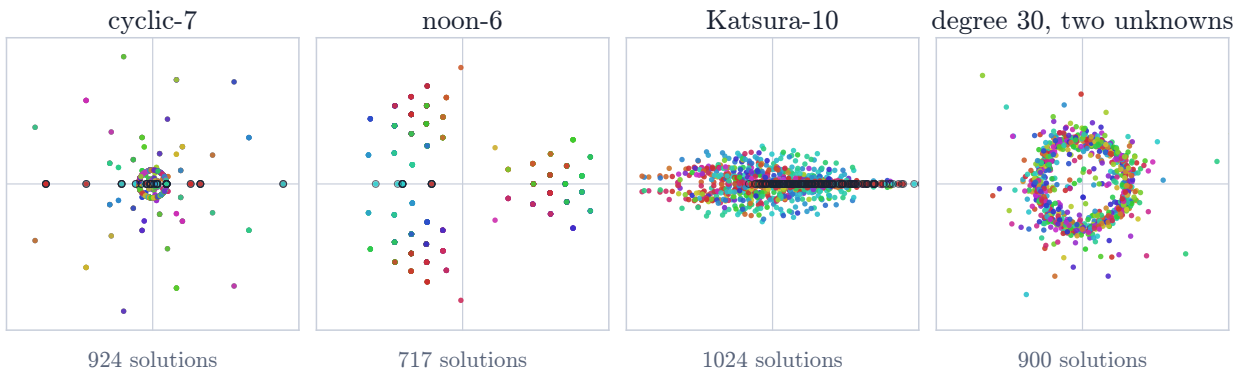


Figure 1: All the solutions of four systems, each certified and the list proved complete: the first coordinate of every solution in the complex plane (the colour is the argument of the second coordinate; a ring marks a solution proved real). The first three take 0.12 s, 0.02 s and 0.10 s on ten threads.

1 Purpose

$F = (F_1, \dots, F_n) : \mathbb{C}^n \to \mathbb{C}^n$ is a polynomial map, F_i of degree $d_i \geq 1$, $d = \max_i d_i$, and we look for all the solutions of $F(x) = 0$, each in a ball that provably contains exactly one, together with a proof that the list is

[1]Computational assistance: AI (Claude, Anthropic) was used for derivations, proofs, numerical experiments, the Lean formalisation, the programs and manuscript preparation. The author is responsible for the claims and their attribution. Section 9 states which results are formal proofs, written proofs and numerical observations.

hexagon:2611.00560v1

complete. The note [9] ended on this sentence: *for systems, the deformation does the global work, and it is not a rectangle construction.* The deformation was the total-degree homotopy [19, 15, 25, 30]. The present note keeps the statements of [9] on the step and on the certificate, which it recalls in Sections 2 and 6, and replaces the rest:

- Section 3: the search without homotopy, and the three sources of the solutions that the sweeps do not find.
- Section 4: the law of the residues, which computes the missing solutions from the others.
- Section 5: the count. Completeness is proved when the number of certified balls reaches a bound; for a sparse system the bound that is reached is the mixed volume, and its computation must itself be a proof.
- Section 6: the certificate, with two additions: a box for a double solution, and the value $F(z)$ computed with more bits inside the certificate in doubles.
- Sections 7 and 8: the algorithm, the measurements, and a comparison with a homotopy solver.

The measurements are those of a compiled implementation, the Rust crate `pandsysroots`, version 0.21.0 [10]. Vectors of $\mathbb{C}^n$ are columns, $\langle u, v\rangle = \sum_i \overline{u_i} v_i$, $\|\cdot\|$ is the Euclidean norm and $\|\cdot\|_\infty$ the maximum norm with its matrix norm. *Complete* always means: proved complete.

2 The step and the deflation

This section recalls [9], where the proofs are.

The step. A support is a function q of a cursor t with $q(1) = 1$. The three means of degree p of [6] are $q_M(t) = \mu(1+t) + \gamma\sqrt{t} - \nu\sqrt{t + \delta(t-1)^2}$, with $2\mu + \gamma - \nu = 1$. Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be analytic, x a point where $f(x) \neq 0$ and $f'(x)$ is invertible, and $D = -f'(x)^{-1} f(x)$ Newton's displacement. On the line $x + sD$ the scalar function $\varphi(s) = \langle f(x), f(x+sD)\rangle / \|f(x)\|^2$ has $\varphi(0) = 1$, $\varphi'(0) = -1$ and

$$\varphi''(0) = L := \frac{\langle f(x), f''(x)(D,D)\rangle}{\|f(x)\|^2}, \qquad t = 1 - \frac{p}{p-1} L, \qquad s = \frac{p}{1 + (p-1)\, q(t)}.$$

The line does not see all of the second-order term: $f''(x)(D,D) = L\, f(x) + R_\perp$ with $R_\perp \perp f(x)$, and $c_\perp = f'(x)^{-1} R_\perp$.

Definition 2.1 (Rectangle step of a system)**.** $x^+ = x + s\,D - \frac{1}{2}\, c_\perp$, with $p = d$ and the three means as support.

For $n = 1$ this is the step of [8]. For every support with $q'(1) = \frac{1}{2}$ it agrees with Chebyshev's step up to a term of third order, and the iteration has order three near a simple solution [9, Proposition 2.2]. The cost is one factorisation of $f'(x)$ and one second derivative along one direction. No other iteration is used anywhere in what follows: the sweeps, the runs from random points, the improvement of a candidate and the refinement with more bits are all this step.

Evaluation. A step needs F, F' and $F''(D,D)$ at the point, and on a large dense system this is most of its cost. The terms of each F_i are kept as a tree of factors: a term is the list of its factors x_v^e, and the terms that begin with the same factors share them. With $S(w)$ the sum of the terms below a node w, without the factors that lead to w, and $L(u)$ the product of the factors from the root to u,

$$S(u) = c(u) + \sum_{w \text{ child of } u} x_{v(w)}^{e(w)}\, S(w), \qquad F_i = S(\text{root}), \qquad \frac{\partial F_i}{\partial x_v} = \sum_u L(u) \sum_{\substack{w \text{ child of } u \\ v(w) = v}} e(w)\, x_v^{e(w)-1}\, S(w),$$

since a term has at most one factor of each unknown. $F''(D,D)$ is twice the coefficient of s^2 of $S(\text{root})$ when each factor is replaced by its expansion along D to the second order. This is about two products per node. A dense polynomial of degree 16 in three unknowns has 969 terms, 2907 factors and 1140 nodes, and the evaluation of one step on three such equations takes 27 microseconds instead of 63 term by term. Far from the origin the values exceed 10^{154} and their squares overflow: F and $F''(D,D)$ are divided by one power of two before L is computed, which leaves it unchanged.

The deflation. Let $z_1, \ldots, z_m$ be the points. For the point z_k and each equation i, let $V_i(k)$ be the $d_i - 1$ points nearest to z_k among the others. The map deflated for the point k is $G_k = (F_i / g_i)_i$, where g_i is the

product over $j \in V_i(k)$ of the linear forms $\langle z_k - z_j, x - z_j\rangle / \|z_k - z_j\|^2$, equal to one at z_k and to zero on the hyperplane through z_j orthogonal to $z_k - z_j$. The step of Definition 2.1 is applied to G_k; it needs F, F' and $F''(D, D)$ at z_k and sums over the neighbours. For $n = 1$ the neighbours are all the other points and the deflated Newton step is that of Ehrlich and Aberth [16, 1]. The number $d_i - 1$ comes from the law of the weights [9, Proposition 3.3]: with all the other points as neighbours the points are sent to infinity, with small equal weights they meet, and with $d - 1$ neighbours of weight one a point that is far comes back in one step and two points cannot settle on one solution.

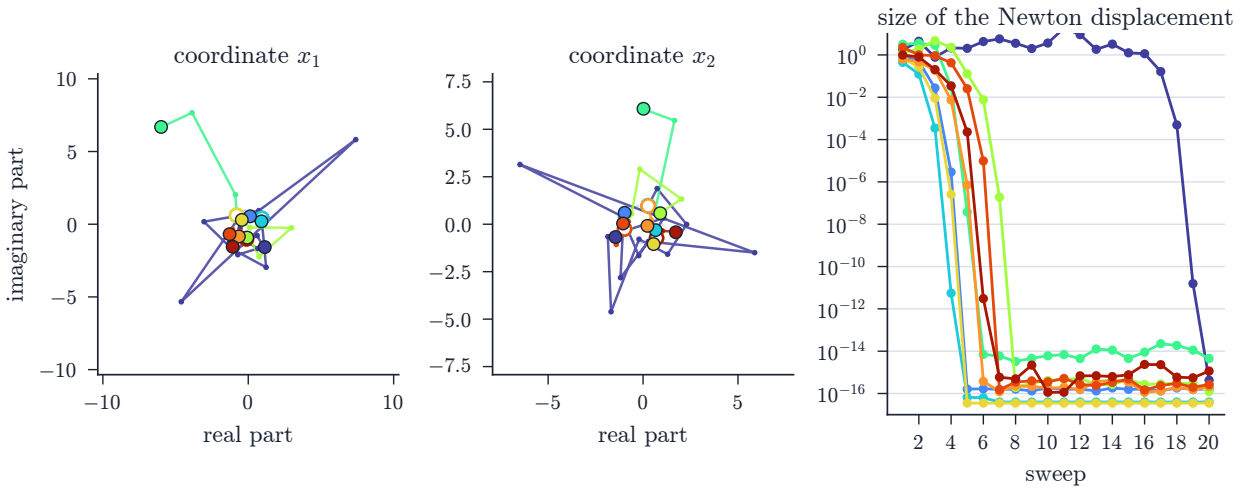


Figure 2: Two cubics with complex Gaussian coefficients, nine points swept on F from the binomial start, all together. Left and centre: the two coordinates of each point after each sweep, from the start to the nine solutions (large dots). Right: the size of the Newton displacement of each point. Eight points settle in five to eight sweeps; the ninth one wanders until the nineteenth sweep before it finds the solution that is left. One example.

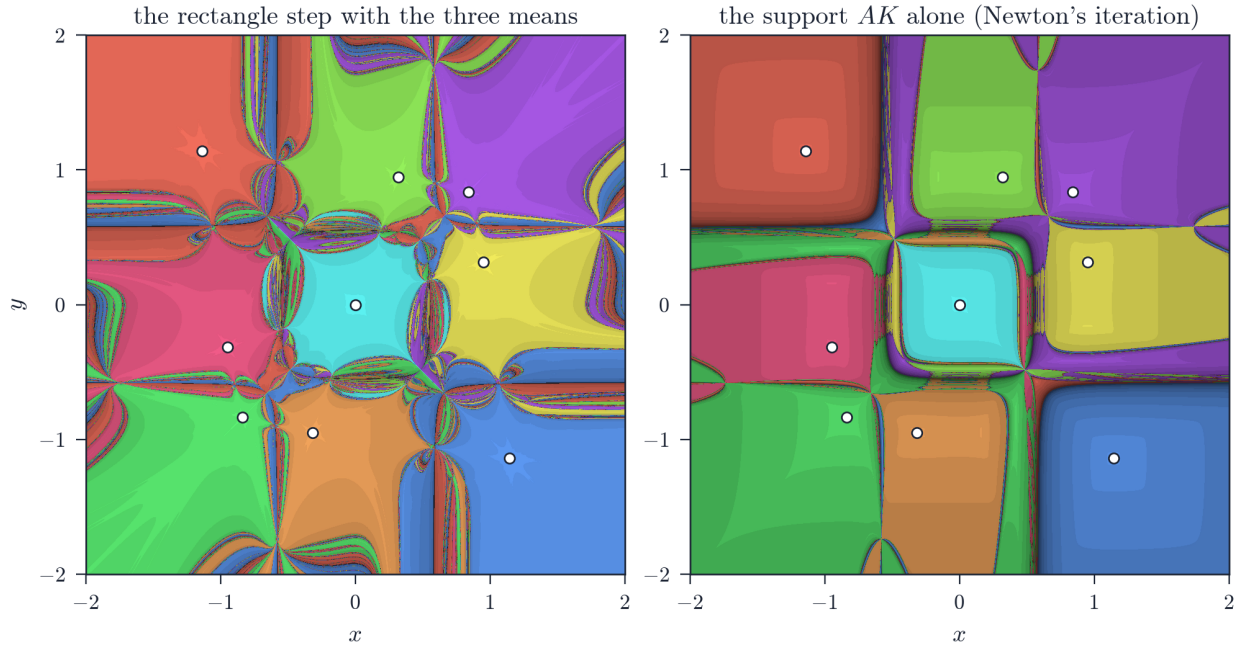


Figure 3: The step of one point alone, without neighbours, on $x^3 - x + 0.3\,y = 0$, $y^3 - y + 0.3\,x = 0$, from each point of a grid of 900×900 real starting points: the colour is the solution reached (the nine solutions are real, white dots), darker when more steps were needed, black when no solution is reached in sixty steps. Left: the rectangle step of Definition 2.1, 5.4 steps on average; 0.4 % of the starts reach no solution. Right: the support AK without the term $c_\perp$, which is Newton's iteration, 7.2 steps on average; every start reaches a solution. The step of order three is faster and its basins are more intricate; in the method the starts that fail cost one run among thousands.

3 The search, without homotopy

Start. The points start on the solutions of a start system; only its solutions are used, the system itself is never deformed. By default it is the binomial system $x_i^{d_i} = r^{d_i}$, whose $N = d_1 \cdots d_n$ solutions are the points $(r\omega_1, \ldots, r\omega_n)$, $\omega_i^{d_i} = 1$, each coordinate being a root extraction of [6, 7]; r is the geometric mean over i of $(|F_i(0)|/c_i)^{1/d_i}$, c_i the largest modulus of a coefficient of degree d_i of F_i. When a partition of the unknowns gives a smaller multihomogeneous Bézout number (Section 5), the start is a product of linear forms that follows the partition, with as many solutions as that number. When the mixed volume is smaller still and is known before the search, as many points as the mixed volume are drawn among those of the start: a point in excess can only leave for infinity, and costs as much as a point that finds a solution. When the system has symmetries beyond the conjugation (below), a solution found gives its images, and the search does not need a point per solution: with $g \leq 8$ the size of the orbit of a point in general position, counted up to eight, it starts from $2/g$ of the count: 1637 points for the 6545 solutions of noon-8, 231 for the 924 of cyclic-7.

Sweeps. All the points are moved together by the step of Section 2 on F, each one deflated of its nearest neighbours (Figure 2). A point whose Newton displacement is at the level of the rounding has settled: it no longer moves and stays a neighbour of the others, so that a sweep costs only the points still looking for a solution. A point that leaves the range of the start by a factor 10^8 is at infinity. The sweeps stop when every point has settled or left, or after a number of sweeps without progress. Nothing guarantees that the sweeps find every solution, and in general they do not: on the large systems of Section 8 they bring 4 % to 88 % of the solutions (Table 1). In [9] this was the reason for the homotopy. Here the rest comes from three sources.

(1) Symmetries. Three symmetries are read on the terms of the system: the conjugation when the coefficients are real; a permutation of the unknowns that sends the set of the equations onto itself up to constants; the multiplication of every unknown by a root of unity of order g when the degrees of the terms of each equation are congruent modulo g. The image of a solution is a solution; it is treated as a candidate and certified like any other point, each image once. For cyclic-7 the sweeps of 231 points find 39 solutions, their images are 759 others, and the 126 that are left come from the runs below and from their images.

(2) Runs from random points. The step of one point alone, from a point drawn at random at the scale of the solutions found, ends on a solution drawn more or less at random (Figure 3); it is cheap, and a batch of several thousand runs is spread over the threads. While many solutions are missing a good part of the runs find a new one. The batches go on as long as they bring something.

(3) The law of the residues, Section 4: the solutions found give the missing ones by linear algebra. Its cost grows as the cube of the number of missing solutions and that of a batch as their number, so, at degree eight at most, the batches come first, while more than two hundred solutions are missing and a batch finds one in sixteen of them, and the residues close the count.

Last, and only when solutions are still missing: restarts of the step deflated of every solution found, from points near the solutions and at several scales; and a second search from another start. A search for solutions that the count announces and that do not exist (a system with a curve of solutions) is given up after two batches and three hundred restarts without result; the output is then not complete, and says so.

system	unknowns	solutions	points	sweeps	symmetries	random runs	residues
cyclic-7	7	924	231	39	759	126	
noon-7	7	2173	544	344	1829		
noon-8	8	6545	1637	872	5673		
Katsura-12	12	4096	4096	3259	474	178	185
two unknowns, degree 40	2	1600	1600	1049			551
three unknowns, degree 16	3	4096	4096	3607			489
8 cubics	8	6561	6561	5343		1018	200
12 quadrics	12	4096	4096	3184		803	109
bilinear $7 + 7$	14	3432	3432	1676		1563	193
eigenvalues, order 64	64	64	64	33	10		21
sparse, four unknowns	4	185	185	121			64

Table 1: Where the solutions come from, on eleven systems of Section 8: the number of points swept, the solutions certified at the end of the sweeps, and those added by each of the three sources (the runs from random points with their images; the last column includes the few partners of close pairs found by the step of a cluster of two). Every system is complete.

4 The law of the residues

Theorem 4.1 (Euler, Jacobi [21]). *Let F have $N = d_1 \cdots d_n$ solutions in $\mathbb{C}^n$, all simple. For every polynomial g of degree at most $E = d_1 + \cdots + d_n - n - 1$,*

$$\sum_{F(\xi)=0} \frac{g(\xi)}{\det F'(\xi)} = 0.$$

In one unknown this is $\sum_\xi \xi^k / f'(\xi) = 0$ for $k \le d-2$. The theorem says that the solutions are not independent: a few of them are determined by the others.

Let $\xi_1, \dots, \xi_K$ be the solutions found and $\zeta_1, \dots, \zeta_r$ the missing ones, $K + r = N$, and $\omega_j = 1/\det F'(\zeta_j) \neq 0$. For an exponent c with $|c| \le E$ the *moment*

$$m(c) := -\sum_{k=1}^{K} \frac{\xi_k^{\,c}}{\det F'(\xi_k)} = \sum_{j=1}^{r} \omega_j \, \zeta_j^{\,c} \tag{4.1}$$

is computed from the solutions found, and is the moment of a measure with one atom per missing solution. Let B be a set of exponents and $\ell(x) = \sum_c \ell_c x_c$ a linear form, such that $|a| + |b| + 1 \le E$ for all $a, b \in B$, and

$$M_0 = \big(m(a+b)\big)_{a,b\in B}, \qquad M_\ell = \Big(\sum_c \ell_c \, m(a+b+e_c)\Big)_{a,b\in B}.$$

Proposition 4.2. *Let V be the $r \times |B|$ matrix $(\zeta_j^{\,a})$, $W = \operatorname{diag}(\omega_j)$ and $\Lambda = \operatorname{diag}(\ell(\zeta_j))$. Then $M_0 = V^T W V$ and $M_\ell = V^T W \Lambda V$. If V has rank r and v is a vector with $Vv = e_j$, then*

$$M_\ell \, v = \ell(\zeta_j)\, M_0 \, v, \qquad M_0 \, v = \omega_j \, \big(\zeta_j^{\,a}\big)_{a\in B}.$$

So the numbers $\ell(\zeta_j)$ are the generalised eigenvalues of the pair (M_ℓ, M_0), and for an eigenvector v the vector $u = M_0 v$ lists the monomials at the missing solution up to a factor: if a and $a + e_c$ are in B, the coordinate c of ζ_j is u_{a+e_c}/u_a.

Proof. By (4.1), $m(a+b) = \sum_j \zeta_j^{\,a} \, \omega_j \, \zeta_j^{\,b}$ and $\sum_c \ell_c m(a+b+e_c) = \sum_j \zeta_j^{\,a} \, \omega_j \, \ell(\zeta_j) \, \zeta_j^{\,b}$. With $Vv = e_j$: $M_0 v = V^T W e_j = \omega_j$ times the row j of V, and $M_\ell v = V^T W \Lambda e_j = \ell(\zeta_j)\, M_0 v$. □

The matrices are those of the multiplication by ℓ in the quotient algebra, read on the missing solutions alone; that the solutions of a system are eigenvalues of such matrices is the method of [3]. What is specific here is where the matrices come from: no Gröbner basis and no resultant, only the values of monomials and of $\det F'$ at the solutions already certified. The size of the matrices is the number of *missing* solutions, not N.

In practice. B is taken a quarter larger than r, so that the elimination can choose well-conditioned monomials, and the rank r of M_0 is read during the elimination. One random form ℓ separates the eigenvalues; they are computed by a Hessenberg reduction and the shifted QR iteration, the eigenvectors by inverse iteration, and the coordinates as the quotients of the proposition. The law needs $2 \max_{a\in B} |a| + 1 \le E$: for two equations of degree 40, $E = 77$, the monomials of degree at most 38 are 780, and the 551 missing solutions of Table 1 are computed from the 1049 others.

Three forms of the law. Theorem 4.1 asks for as many solutions as Bézout's number. Two extensions are used, each tied to a count of Section 5. *(a) Groups of unknowns.* If the system has as many simple solutions as the multihomogeneous Bézout number of a partition of the unknowns into groups of sizes $k_1, \dots, k_s$, the sum vanishes for every g whose degree in the unknowns of the group j is at most $E_j = \sum_i d_{ij} - k_j - 1$, d_{ij} being the degree of F_i in that group. This covers the bilinear systems and the eigenvalue problems, where Bézout's number is far above the number of solutions. *(b) Newton polytopes.* If the system has as many simple solutions as the mixed volume of its Newton polytopes, the sum of $\xi^{\,c} / \det F'(\xi)$ vanishes for every exponent c such that $c + (1, \dots, 1)$ is in the interior of the sum of the polytopes (Khovanskii [22]). Whether an exponent is inside the sum is decided by a linear program.

What rests on the law. Nothing. A candidate computed by the law is improved by the step of Definition 2.1 and must pass the certificate of Section 6; if the hypotheses of the law fail (a solution that is not simple, a count that is not reached), the candidates are close to nothing and are refused. The law is a way to find points, not an argument of the proof.



Figure 4: Two equations of degree 12 with complex Gaussian coefficients, 144 solutions. The sweeps alone find 134 of them (grey). The ten others (rings) are computed from the 134 by Proposition 4.2 (crosses): before any step of improvement the candidates are within $4 \cdot 10^{-15}$ to 10^{-12} of the solutions (right). One example.

5 The count

The output is complete when the number of disjoint certified balls is a bound of the number of isolated solutions in $\mathbb{C}^n$, counted with multiplicity. The bounds used, the smallest being kept:

- Bézout's number $N = d_1 \cdots d_n$;
- the multihomogeneous Bézout number of a partition of the unknowns into groups of sizes $k_1, \ldots, k_s$: the coefficient of $a_1^{k_1} \cdots a_s^{k_s}$ in $\prod_i (d_{i1}a_1 + \cdots + d_{is}a_s)$ (Morgan and Sommese [26]); all the partitions up to nine unknowns, searched once per system;
- the mixed volume $\mathrm{MV}(P_1, \ldots, P_n)$ of the Newton polytopes P_i of the supports A_i with the origin added (Bernstein [5] in the torus; Li and Wang [24] for $\mathbb{C}^n$).

Remark 5.1. When every P_i is the whole product of simplices $\prod_j d_{ij}\Delta_{k_j}$ of a partition, the mixed volume is the multihomogeneous Bézout number of the partition, and nothing has to be computed; it is enough to look for the vertices of the product in the support. This is the case of the dense and of the bilinear systems.

Mixed cells. Up to three unknowns the mixed volume is computed from the volumes of the sums of the polytopes, exactly, in integers. Beyond, by the mixed cells of Huber and Sturmfels [20]. Give each point a of each support A_i an integer $w_i(a)$, the *lifting*. For $g \in \mathbb{R}^n$ the lifted function of the support i is $a \mapsto \langle a, g\rangle + w_i(a)$.

Definition 5.2. A *choice of edges* is a pair $\{p_i, q_i\} \subset A_i$ for each i, with $\Delta = \det(p_1 - q_1, \ldots, p_n - q_n) \neq 0$; its vector g is the solution of the n equations $\langle p_i - q_i, g\rangle = w_i(q_i) - w_i(p_i)$. It is a *mixed cell* of w if, for every i, the lifted function of A_i at g is smallest at p_i and q_i and nowhere else. It is a *tie* of w if it is smallest at p_i and q_i for every i, and also at a third point for some i.

Theorem 5.3 (Huber, Sturmfels [20])**.** *For a lifting in general position,* $\mathrm{MV}(P_1, \ldots, P_n) = \sum |\Delta|$ *over the mixed cells.*

The liftings that are not in general position are contained in a finite union of hyperplanes of the space of liftings (the walls of the regular subdivisions of the Cayley configuration of the supports). A lifting drawn at random among real numbers is in general position with probability one; a lifting drawn among the integers below 2^{36} is so with a probability that is close to one and is not one. A count that proves completeness cannot rest on it. Computing the cells with two liftings and comparing is evidence; the next statement is a proof.

Theorem 5.4 (One lifting)**.** *Let w be any lifting. If w has no tie, then* $\mathrm{MV}(P_1, \ldots, P_n) = \sum |\Delta|$ *over the mixed cells of w.*

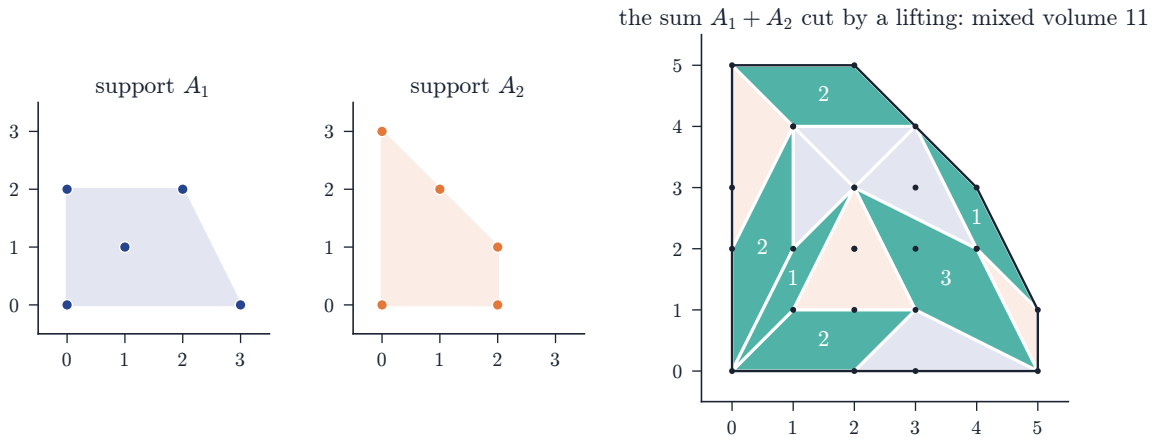


Figure 5: Two supports in the plane (left, centre) and the subdivision of $A_1 + A_2$ that a lifting induces (right). The mixed cells (green) are the sums of an edge of A_1 and an edge of A_2; the number in each is $|\Delta|$, and their sum, 11, is the mixed volume: two equations with these supports have at most 11 isolated solutions. The other cells are the sum of a triangle of one support and a point of the other.

Proof. Let η be a direction of the space of liftings contained in none of the hyperplanes of the paragraph above. Then $w_\varepsilon = w + \varepsilon\eta$ is in general position for every small $\varepsilon > 0$: a hyperplane that contains w does not contain w_ε, and the others are at a positive distance from w. Fix a choice of edges. Its vector for w_ε is $g_\varepsilon = g + \varepsilon h$, an affine function of ε, since the matrix of the n equations does not depend on the lifting. Each condition "the lifted function is not smaller at a than at p_i" therefore reads $A + \varepsilon B \le 0$, with A its value for w.

A mixed cell of w is a mixed cell of w_ε for small ε: its conditions hold strictly at $\varepsilon = 0$, that is $A < 0$, hence $A + \varepsilon B < 0$ for small ε.

A choice of edges that is a mixed cell of w_ε for arbitrarily small $\varepsilon > 0$ is a mixed cell of w: every condition satisfies $A + \varepsilon B \le 0$ for arbitrarily small $\varepsilon > 0$, hence $A \le 0$; the lifted functions are smallest at p_i and q_i for w, and since w has no tie, nowhere else.

The choices of edges are finitely many, so for ε small enough w and w_ε have the same mixed cells, with the same Δ, and Theorem 5.3 applies to w_ε. □

The search of the cells enumerates the choices of edges support by support. A partial choice puts linear conditions on g: equalities for the edges, inequalities for the other points. It is given up when these conditions have no solution, which is decided by the simplex method in doubles on the dual program: weights $y \ge 0$ on the conditions, of sum one, whose combination of the rows vanishes, and that make the same combination of the right sides smallest. A new edge adds columns to this program, so the basis of the parent choice is still a basis and a few pivots are enough. A negative value comes with the $n + 1$ conditions of the basis, and the doubles are *never trusted*: the choice is given up only if an exact elimination on these $n + 1$ conditions, in integers, produces a combination with a zero left side and a negative right side. At the last support the $n - 1$ edges chosen leave a line of vectors g, computed once in integers, and every edge of the last support is tested on it by signs of products of integers; a condition met with equality by a choice whose other conditions hold is a tie, and the lifting is then drawn again. So for the lifting that is kept, every mixed cell is found and counted exactly, no tie exists, and Theorem 5.4 applies. A float error can only make the search longer.

The mixed volume of cyclic-7 (924) takes 0.05 s, that of cyclic-9 (11 016) 1.7 s, that of noon-9 (19 665) 0.7 s. It is computed before the search when the other counts would start a hundred points or more.

6 The certificate

Let $z \in \mathbb{C}^n$ and let A be any matrix, in practice an approximate inverse of $F'(z)$. Write $F(z+h) = \sum_\alpha c_\alpha h^\alpha$ and $\eta = \|AF(z)\|_\infty$, $\delta = \|I - AF'(z)\|_\infty$, $T_k = \max_i \sum_{|\alpha|=k} |(Ac_\alpha)_i|$.

Theorem 6.1 (Certificate [9, Theorem 6.1])**.** *If $\eta + \delta\rho + \sum_{k\ge2} T_k\rho^k \le \rho$ and $\delta + \sum_{k\ge2} kT_k\rho^{k-1} < 1$, the closed ball $B(z,\rho)$ of the maximum norm contains exactly one solution of $F = 0$, and it is simple.*

The proof is a contraction of the ball by $x \mapsto x - AF(x)$. If M certified balls are pairwise disjoint, F has at least M solutions; if M is a bound of Section 5, it has no other isolated solution. For a real system with a complete output, a solution whose mirror ball meets no other ball is real [9, Proposition 6.3].

A cheaper bound. The sums T_k need all the Taylor coefficients at z, which is most of the cost at high degree. Let $|F_i|$ be the polynomial with the moduli of the coefficients of F_i, and $H_i(\rho)$ the sum of all its second derivatives at the point $(|z_1| + \rho, \dots, |z_n| + \rho)$.

Proposition 6.2. *Let $S(\rho) = \max_i \sum_j |A_{ij}|\, H_j(\rho)$. In Theorem 6.1, $\sum_{k\geq 2} T_k \rho^k$ may be replaced by $\frac{1}{2}\rho^2 S(\rho)$ and $\sum_{k\geq 2} kT_k\rho^{k-1}$ by $\rho\, S(\rho)$. In particular, with $\rho = 2\eta$, the ball is certified as soon as $2\delta + \rho\, S(\rho) \leq 1$ and $\delta + \rho\, S(\rho) < 1$.*

Proof. For $\|h\|_\infty \leq \rho$ and $0 \leq \theta \leq 1$, each second derivative of F_i at $z + \theta h$ is bounded in modulus by the same derivative of $|F_i|$ at $(|z_c| + \rho)_c$, a polynomial with nonnegative coefficients. Taylor's formula bounds the terms of order two and more of $F_i(z + h)$ by $\frac{1}{2}\rho^2 H_i(\rho)$ and their derivative along a direction of norm one by $\rho\, H_i(\rho)$; these are the two quantities that the proof of Theorem 6.1 uses. □

One evaluation instead of the whole expansion; the sums T_k are computed only when this test does not conclude.

The value with more bits. The radius of the ball is 2η, and η is computed from an enclosure of $F(z)$. At high degree the value of $F(z)$ in interval arithmetic of doubles is an interval far wider than the number: the ball is certified, and wide, though the point is as good as a double can be. On two equations of degree 40, this happened at 743 of the 1600 solutions. Refining the point with 128 bits does not address the cause. Instead, $F(z)$ alone is computed with fixed-point balls of 128 bits and rounded outward to doubles, and the certificate in doubles is run again with this enclosure; A, δ and S are unchanged. Then, if the ball is still wide, the point is moved by the step computed from that value, while the radius decreases: where the square of the displacement is below the last bit of the point, the terms of order two of Definition 2.1 are below the rounding and the step is its first term.

Beyond doubles. A point that doubles do not certify is improved by the step of Definition 2.1 on centres of 128, 256, ... bits and certified at that precision; the ball returned around the centre rounded to doubles is certified with the slack of that rounding. A second entry returns centres and radii at any precision asked for.

Double solutions. Theorem 6.1 proves simplicity, and cannot hold at a solution of multiplicity two, nor in doubles at two solutions closer than the rounding allows to separate. At a point c where $F'(c)$ is nearly singular, with v and u its right and left vectors of smallest singular value, take coordinates $x = c + Th$ whose first axis is v and a matrix M whose first row is u^*, such that $G(h) = MF(c + Th)$ is close to $P(h) = (a\, h_1^2, h_2, \dots, h_n)$.

Proposition 6.3 (Double box)**.** *If $|G_1 - a\, h_1^2| < |a|\, r_1^2$ and $|G_i - h_i| < r_2$ for $i \geq 2$ on the polydisc $D = \{|h_1| \leq r_1,\ |h_i| \leq r_2\}$, then F has exactly two solutions in $c + TD$, counted with multiplicity.*

Proof. P has one zero in D, of multiplicity two. For the norm $\nu(u) = \max\big(|u_1|/(|a|r_1^2), \max_{i\geq 2} |u_i|/r_2\big)$, on the boundary of D, $\nu(P)$ is at least one and $\nu(G - P)$ is below one: by Rouché's theorem for maps of $\mathbb{C}^n$ [2], G and P have the same number of zeros in D. □

The two inequalities are checked on the Taylor coefficients of G in interval arithmetic, and a double box counts for two in the count. A point that doubles do not certify is tried with 128 bits, which tell two close solutions apart, then with a double box, and only then with more bits: on a double solution more bits only stall, and the refinement stops by itself after fourteen rounds in a row that take less than three quarters off the correction. The box is tried wherever points have ended, whatever their number, and "nearly singular" is judged against the size of the terms of the Jacobian read a little away from the axes, so that a double solution with a zero coordinate ($x^2 q(x)$ at the origin) is seen. A place known not to hold a simple solution (a double box, a point that no number of bits certified) is not certified again when a later run ends there.

7 The algorithm

1. *Units.* The unknowns and the equations are scaled by powers of two that bring the coefficients close to one; powers of two change no mantissa, and the radii are returned in the units given.

2. *Count.* The bounds of Section 5; the mixed volume first when the others would start a hundred points or more.
3. *Sweeps* from the start (Section 3), then the certificate at every point.
4. *The other sources*, while solutions are missing: the symmetries; batches of runs from random points; the law of the residues; the partner of a close pair by the step of a cluster of two; double boxes; restarts deflated of the solutions found; a second search.
5. *Output:* the certified balls, the double boxes, whether the list is complete, for a real system which solutions are real, and the points that were not certified.

Theorem 7.1. *The algorithm stops. If the bounds computed in interval arithmetic are valid, every ball of the output contains exactly one solution of $F = 0$, a simple one, every double box contains exactly two solutions counted with multiplicity, and distinct balls and boxes are disjoint. If the output says that it is complete, F has no other isolated solution in $\mathbb{C}^n$, and a solution flagged real is real.*

Proof. Every loop is bounded. A ball of the output passed Theorem 6.1, in the form of Proposition 6.2 or with the sums T_k; a double box passed Proposition 6.3; disjointness is checked on lower bounds of the distances. The output says that it is complete when the number of balls plus twice the number of double boxes is one of the bounds of Section 5, each of which bounds the number of isolated solutions counted with multiplicity; when that bound is a mixed volume computed by cells, Theorem 5.4 gives it. □

As in [8, 9] the theorem guarantees the truth of the output, and its completeness only when the output claims it. What is new is how often it can be claimed.

8 Measurements

All the times are wall times on one machine (Apple M5, ten cores, 16 GB), with the crate `pandsysroots` 0.21.0.

family	systems	as expected	proved complete	certified	ms
dense, complex	4 000	4 000	4 000	77 352	2.36
dense, real	4 000	4 000	4 000	80 384	2.19
one unknown, degree 2 to 30	1 500	1 500	1 500	24 287	0.22
integer coefficients	1 500	1 500	1 500	13 542 + 1 d.	0.27
coefficients from 10^{-4} to 10^{4}	1 500	1 500	1 500	12 883	1.18
no constant term (origin)	1 000	1 000	1 000	9 237	0.37
sparse, with the pure powers	2 500	2 500	2 500	74 666	2.20
sparse, random supports	2 500	2 500	2 500	25 326	2.30
sparse, four unknowns	500	500	498	10 893	13.56
triangular	2 000	2 000	2 000	39 448	2.15
products of linear forms	1 000	1 000	1 000	13 562	0.92
symmetric functions	500	500	500	3 000	0.11
eigenvalues as a system	1 500	1 500	1 500	6 849	1.40
eigenvalues, symmetric matrix	1 000	1 000	1 000	4 007	0.97
bilinear in two groups	1 500	1 500	1 500	7 474	2.44
coupled powers	1 000	1 000	1 000	31 373	0.71
Katsura, perturbed	1 000	1 000	1 000	15 108	1.33
other units for the unknowns	2 000	2 000	2 000	17 025	0.31
other units for the equations	1 000	1 000	1 000	8 599	0.29
spheres	1 500	1 500	1 500	3 000	1.72
conic and nearly tangent line	1 500	1 500	1 500	3 000	0.89
close pair, three unknowns	500	500	500	1 000	0.60
conic and tangent line, exact	1 000	1 000	1 000	1 000 d.	0.83
double solutions	500	500	500	1 616 d.	13.71
overdetermined, consistent	1 000	1 000	1 000	8 987	0.30
overdetermined, inconsistent	1 000	1 000	1 000	9 311	0.31
underdetermined	1 500	1 500	1 500	6 661	0.18
total	**40 000**	**40 000**	**39 998**	**506 974 + 2 617 d.**	

Table 2: Forty thousand systems of twenty-seven families, two to five unknowns for most of them, each solved on one thread (ten at a time). "As expected": proved complete, except for the overdetermined systems (every solution kept, or the absence of solution proved), the underdetermined ones (complete, with the degree of the solution set) and the sparse systems of four unknowns (nothing left uncertified). "d.": double boxes. ms: mean time for one system.

Forty thousand systems (Table 2). The corpus is drawn with fixed seeds: dense and sparse systems, integer and badly scaled coefficients, triangular systems, products of linear forms, eigenvalue problems, bilinear systems, close pairs and exact double solutions, overdetermined and underdetermined systems. Every system gives what is expected of it, and 39 998 outputs are complete; the two others are sparse systems of four unknowns that have fewer solutions than their mixed volume, where every solution found is certified and nothing proves that the list is complete. The whole corpus takes 8.1 s on ten threads: 506 974 simple solutions and 2617 double boxes.

Classical systems, and a homotopy solver (Figure 6). The same systems, written to text files, were given to HomotopyContinuation.jl 2.23 [12], with `solve` then `certify` [13], on the same machine with ten threads; its time is that of a second run, without the compilation of the code of each system, which takes it 1 to 185 s more the first time. The systems are cyclic-n [11], noon-n [27] and Katsura-n (as in the collection of [31]), dense systems of high degree, systems of quadrics and of cubics, bilinear systems $\sum_{a,b} c_{ab} x_a y_b + \cdots = 0$ in two groups of unknowns, the eigenvalue problem $Mv = \lambda v$ as a system, and a sparse system with random supports.

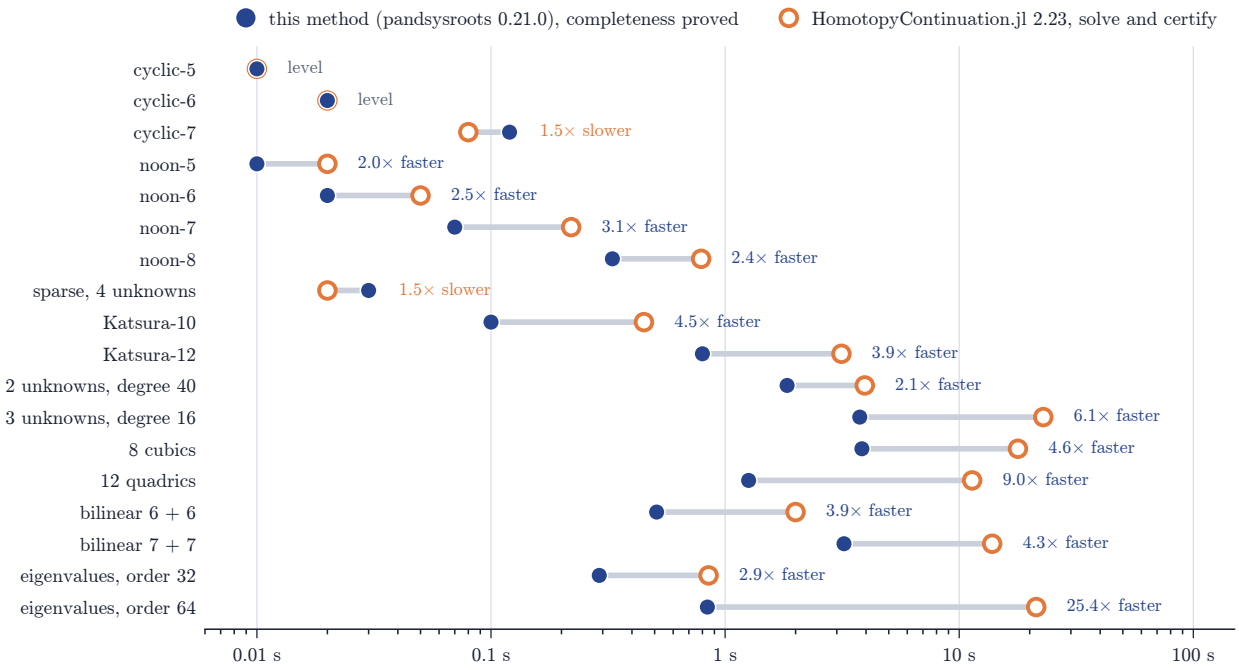


Figure 6: Eighteen systems, 32 to 6561 solutions, 2 to 64 unknowns: the time to find and certify all the solutions. Both programs return all of them on every line; the method of this note also proves that there is no other. Times below 0.01 s are shown as 0.01 s.

Both find and certify every solution of every system. The homotopy solver is ahead on cyclic-7 and on the sparse system, by a factor of 1.5; the two are level on cyclic-5 and cyclic-6; the method of this note is ahead on the fourteen others: by 2 to 3 on the noon systems, on the system of degree 40 and on the eigenvalue problem of order 32, by 4 to 9 on the Katsura systems, the system of degree 16, the quadrics, the cubics and the bilinear systems, and by 25 on the eigenvalue problem of order 64. Two differences of nature should be kept in mind. The homotopy solver follows as many paths as its start system has solutions and certifies what it finds; it does not prove that no solution is missing. And its strength is where this method was weakest: seven versions of the crate ago, it was 100 to 5000 times faster on the sparse systems, and this method missed solutions there (Figure 7).

A system that is not zero-dimensional. cyclic-8 has 1152 isolated solutions and a curve of solutions [4]; its mixed volume is 2560. The method returns the 1152 isolated solutions, certified, in 2 s, then looks for the others for 7 s and stops: the output is not complete, and says so. (The homotopy solver stopped with an error of its linear algebra on this file, twice, with its default settings.) A count that sees the curve is not implemented.

One unknown (Table 3). For $n = 1$ the method is a variant of that of [8], without what is specific to one polynomial: the square-free decomposition, the exact multiplicities, the orders four to six. The comparison measures what these are worth. The solver of [8] is complete on the 50 000 polynomials in 1.9 s of processor time; the method of this note is complete on 49 995 in 67 s: ten to twenty times slower on the simple cases,

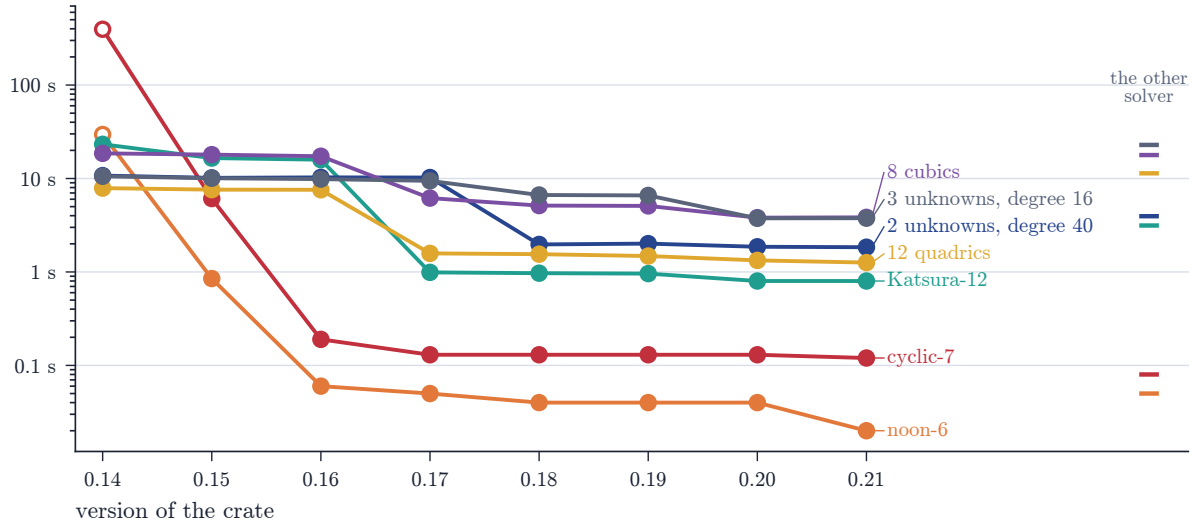


Figure 7: Seven of the systems across eight versions of the crate, and the homotopy solver (ticks on the right). A hollow marker is a run that was not complete: 916 of the 924 solutions of cyclic-7, 714 of the 717 of noon-6. Each fall has one cause: the count (0.15: the mixed volume beyond five unknowns, without which the search looked for solutions that do not exist; 0.16: the pruning of the search of the cells, and one lifting), what follows the sweeps (0.17: the random runs before the residues, the partners in parallel), the certificate at high degree (0.18: $F(z)$ with more bits), the evaluation on the tree of factors (0.20), the start of a symmetric system (0.21). The version 0.19 changed the double solutions and the large values, which these systems do not have.

		the solver of [8]		this method, $n = 1$	
family	polynomials	complete	ms	complete	ms
complex coefficients, degree 2 to 40	8 000	8 000	0.031	8 000	0.46
real coefficients, degree 2 to 40	8 000	8 000	0.029	8 000	0.42
integers from -9 to 9, degree 2 to 30	6 000	6 000	0.021	6 000	0.24
roots drawn in the disc, degree 2 to 30	6 000	6 000	0.053	6 000	0.60
real roots in $[-2, 2]$, degree 2 to 20	5 000	5 000	0.048	4 999	0.74
trinomials, degree 3 to 40	5 000	5 000	0.020	5 000	0.27
coefficients from 10^{-4} to 10^{4}	4 000	4 000	0.019	4 000	0.35
complex coefficients, degree 60 to 100	2 000	2 000	0.166	2 000	21.79
two roots 10^{-6} apart, degree 3 to 15	3 000	3 000	0.052	3 000	0.61
one double root, exact	3 000	3 000	0.024	2 996	0.94
total	**50 000**	**50 000**		**49 995**	

Table 3: Fifty thousand polynomials in one unknown (960 386 roots), each solved on one thread by the solver of [8] (`pandroots` 0.14.0) and by the method of this note as a system of one equation. On the 49 995 polynomials where both are complete, every ball of one meets the discs of the other for the same number of roots.

forty times on the polynomials with an exact double root, a hundred and thirty times at degree 60 to 100. Of the five polynomials that are not proved complete, four have a root of multiplicity three or four, for which this note has no certificate, and one has real roots close to one another. Where both are complete, the two certificates agree on every root. For one polynomial the solver of [8] is the one to use. This comparison is also what showed two defects of an earlier version, since corrected: a double solution at the origin was not seen, and a root of modulus 10^6 at degree 28 was lost to the overflow of a square (with them, 49 497 polynomials were complete, in 1183 s).

9 Evidence and limits

Formalised in Lean 4 (Lean 4.35.0, Mathlib v4.35.0; fourteen files, 64 theorems, no `sorry`, the standard axioms only; folder `lean`):

- `ThreeMeans`, `DeflatedStep`, `Obstruction`, `Certificate` (31 theorems): the files of [9]. In particular Theorem 6.1 in a Banach space, from Mathlib's contraction theorem, and the count: as many disjoint certified balls as a bound, no other solution.
- `Residues`, `Residues2`, `Residues3` (12): the formula of Theorem 4.1 in one unknown and for a decoupled system in two; that its degree bound is sharp; Proposition 4.2: a moment matrix applied to a vector that selects one missing solution gives the monomials there (`moment_mulVec`), a coordinate is a quotient of two entries (`coordinate_from_two_rows`), and the matrix of a linear form has the same vectors (`linear_form_moment_mulVec`).
- `Sparse` (5): a combination of linear conditions with nonnegative weights on the inequalities, zero on the left and negative on the right, proves that they have no solution (the test by which a branch of the search of the cells is given up); the two symmetries of Section 3 that are not permutations.
- `Cells` (5): the two steps of the proof of Theorem 5.4, on one condition (`strict_persists`, `limit_weak`, `cell_of_perturbed`), and the weak duality of the linear program of the search.
- `Apart` (3): two balls are disjoint as soon as the real parts of one coordinate are far enough, and the test made in doubles implies the exact inequality.
- `Sharp` (3): the product of two fixed-point balls, rounded down, with the radius that the program gives it.
- `Scale` (1), `Tree` (2), `Symmetry` (2): the quotient L is unchanged when F and $F''(D, D)$ are multiplied by the same number; the two identities of the evaluation on the tree of factors; the image of a solution by a permutation of the unknowns that sends the equations onto themselves up to constants is a solution, and distinct solutions have distinct images.

Not formalised: Theorem 5.3 and the statement on the hyperplanes that precedes Theorem 5.4, which are quoted; Theorem 4.1 in several unknowns and its two extensions, which are quoted and on which no claim of the output rests; the bounds of Section 5 (Bézout, Morgan and Sommese, Bernstein, Li and Wang), which are hypotheses of the counting statement; Rouché's theorem in Proposition 6.3; that the interval and fixed-point arithmetic of the program produce the bounds the theorems ask for; the program itself.

Written proofs: Propositions 4.2, 6.2, 6.3; Theorems 5.4 and 7.1; Remark 5.1 is a known fact stated without proof.

Checked numerically. That the mixed cells give the same sum with six different liftings, and the same number as the volumes of the sums of the polytopes in two to four unknowns (tests of the crate); the counts 70, 156, 924, 2560 for cyclic-5 to 8. The agreement of the two certificates on 49 995 polynomials (Table 3). The tables and figures of Section 8.

Not new. The total-degree and multihomogeneous counts [26], the mixed volume as a count [5, 24], mixed cells and their enumeration with linear programs [20, 18]; the formula of Euler and Jacobi and its form on Newton polytopes [21, 22]; solutions as eigenvalues of multiplication matrices [3]; deflation by solutions already found [14, 17]; certificates by contraction and by interval arithmetic [23, 28, 29, 13]. What is specific to this note: a search for all the solutions of a system that follows no path, with the rectangle step as its only iteration; the use of the law of the residues to compute the solutions that a search has missed from those it has found (Proposition 4.2 and its use), in the dense, multihomogeneous and polytope cases; Theorem 5.4, which turns the count by mixed cells into a proof for the lifting actually drawn; the certificate in doubles fed

with the value in fixed point; the start of a symmetric system from fewer points than solutions; and the measurements.

Limits.

- *No theorem of convergence.* Nothing proves that the search finds every solution; when it does not, the output is correct and not complete. On the corpus and the systems reported it found them all whenever the count was reached by the system, which is an observation.
- *The count must be reached.* Completeness is proved only when the system has as many isolated solutions, with multiplicity, as one of the bounds. Two sparse systems of four unknowns of Table 2 have fewer; a system with a curve of solutions (cyclic-8) is not proved complete, and time is lost looking for what does not exist.
- *Multiplicity.* Double solutions have their box; a solution of multiplicity three or more is not certified, and there is no counterpart of the square-free decomposition of [8] (four polynomials of Table 3).
- *The comparison* is on one machine, with systems chosen by the author and the other solver with its default settings; on other families the order may differ. The cost of a sweep is one factorisation of an $n \times n$ matrix per point, and the sweeps are most of the time on the large dense systems. The result depends on the rounding: a change in the order of the sums of the evaluation changed which solutions the sweeps found, and one system took ten times longer until the runs from random points were lengthened.
- *The orders* four and six of [8] are lost for systems, for the structural reason given in [9, Proposition 4.1]; the order is three.
- *The search of the mixed cells* is given up beyond eight million branches or forty terms in an equation; the count then falls back on the other bounds.

References

[1] O. Aberth. Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comp.* 27 (1973), 339–344.

[2] L. A. Aĭzenberg, A. P. Yuzhakov. *Integral Representations and Residues in Multidimensional Complex Analysis*. Amer. Math. Soc., 1983.

[3] W. Auzinger, H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Numerical Mathematics, Singapore 1988*, Birkhäuser, 1988, 11–30.

[4] J. Backelin. Square multiples n give infinitely many cyclic n-roots. Reports, Matematiska Institutionen, Stockholms Universitet, 1989.

[5] D. N. Bernstein. The number of roots of a system of equations. *Funct. Anal. Appl.* 9 (1975), 183–185.

[6] I. Besevic. *Geometric Root Calculators*, version V29, 2026.

[7] I. Besevic. *Complex Geometric Root Calculators*, version C5, 2026.

[8] I. Besevic. *The simultaneous rectangle*, version G8, revision 9, 2026. `https://doi.org/10.5281/zenodo.24257672`.

[9] I. Besevic. *The simultaneous rectangle for systems*, version S1, 2026.

[10] I. Besevic. `pandsysroots`, version 0.21.0, 2026. `https://crates.io/crates/pandsysroots`.

[11] G. Björck, R. Fröberg. A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots. *J. Symbolic Comput.* 12 (1991), 329–336.

[12] P. Breiding, S. Timme. HomotopyContinuation.jl: a package for homotopy continuation in Julia. In *Mathematical Software – ICMS 2018*, Springer, 2018, 458–465.

[13] P. Breiding, K. Rose, S. Timme. Certifying zeros of polynomial systems using interval arithmetic. *ACM Trans. Math. Software* 49 (2023), no. 1.

[14] K. M. Brown, W. B. Gearhart. Deflation techniques for the calculation of further solutions of a nonlinear system. *Numer. Math.* 16 (1971), 334–342.

[15] F. J. Drexler. Eine Methode zur Berechnung sämtlicher Lösungen von Polynomgleichungssystemen. *Numer. Math.* 29 (1977), 45–58.

[16] L. W. Ehrlich. A modified Newton method for polynomials. *Comm. ACM* 10 (1967), 107–108.

[17] P. E. Farrell, Á. Birkisson, S. W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM J. Sci. Comput.* 37 (2015), A2026–A2045.

[18] T. Gao, T. Y. Li, M. Wu. Algorithm 846: MixedVol: a software package for mixed-volume computation. *ACM Trans. Math. Software* 31 (2005), 555–560.

[19] C. B. Garcia, W. I. Zangwill. Finding all solutions to polynomial systems and other systems of equations. *Math. Programming* 16 (1979), 159–176.

[20] B. Huber, B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comp.* 64 (1995), 1541–1555.

[21] C. G. J. Jacobi. Theoremata nova algebraica circa systema duarum aequationum inter duas variabiles propositarum. *J. Reine Angew. Math.* 14 (1835), 281–288.

[22] A. G. Khovanskii. Newton polyhedra and the Euler–Jacobi formula. *Russian Math. Surveys* 33 (1978), no. 6, 237–238.

[23] R. Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing* 4 (1969), 187–201.

[24] T. Y. Li, X. Wang. The BKK root count in $\mathbb{C}^n$. *Math. Comp.* 65 (1996), 1477–1484.

[25] A. P. Morgan, A. J. Sommese. Computing all solutions to polynomial systems using homotopy continuation. *Appl. Math. Comput.* 24 (1987), 115–138.

[26] A. P. Morgan, A. J. Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Appl. Math. Comput.* 24 (1987), 101–113.

[27] V. W. Noonburg. A neural network modeled by an adaptive Lotka–Volterra system. *SIAM J. Appl. Math.* 49 (1989), 1779–1792.

[28] S. M. Rump. Verification methods: rigorous results using floating-point arithmetic. *Acta Numer.* 19 (2010), 287–449.

[29] S. Smale. Newton's method estimates from data at one point. In *The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics*, Springer, 1986, 185–196.

[30] A. J. Sommese, C. W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science.* World Scientific, 2005.

[31] J. Verschelde. Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Software* 25 (1999), 251–276.