Comparison of probabilistic algorithms for analyzing the components of an affine algebraic variety

Daniel J. Bates

Colorado State University, Department of Mathematics,
Fort Collins, CO 80523-1874 – USA

Wolfram Decker

University of Kaiserslautern, Department of Mathematics,
Erwin-Schrödinger-Str., D-67663 Kaiserslautern – Germany

Jonathan D. Hauenstein

North Carolina State University, Department of Mathematics,
Raleigh, NC 27695-8205 – USA

Chris Peterson

Colorado State University, Department of Mathematics,
Fort Collins, CO 80523-1874 – USA

Gerhard Pfister

University of Kaiserslautern, Department of Mathematics,
Erwin-Schrödinger-Str., D-67663 Kaiserslautern – Germany

Frank-Olaf Schreyer

University of Saarbrücken, Mathematics and Computer Sciences
Building E2.4, D-661223 Saarbrücken – Germany

Andrew J. Sommese

University of Notre Dame, Dept. of Applied & Computational Mathematics and Statistics
Notre Dame, IN 46556 – USA

Charles W. Wampler

General Motors Research and Development

Email addresses: bates@math.colostate.edu (Daniel J. Bates), decker@mathematik.uni-kl.de (Wolfram Decker), hauenstein@ncsu.edu (Jonathan D. Hauenstein), peterson@math.colostate.edu (Chris Peterson), pfister@mathematik.uni-kl.de (Gerhard Pfister), schreyer@math.uni-sb.de (Frank-Olaf Schreyer), sommese@nd.edu (Andrew J. Sommese), Charles.W.Wampler@gm.com (Charles W. Wampler)

URL: www.math.colostate.edu/∼bates (Daniel J. Bates), www.mathematik.uni-kl.de/∼decker/de (Wolfram Decker), www.math.ncsu.edu/∼jdhauens (Jonathan D. Hauenstein), www.math.colostate.edu/∼peterson (Chris Peterson), www.mathematik.uni-kl.de/∼pfister/de (Gerhard Pfister), www.math.uni-sb.de/ag/schreyer (Frank-Olaf Schreyer), www.nd.edu/∼sommese (Andrew J. Sommese), www.nd.edu/∼cwample1 (Charles W. Wampler)

Mail Code 480-106-224, 30500 Mound Road, Warren, MI 48090 – USA

Abstract

Systems of polynomial equations arise throughout mathematics, engineering, and the sciences. It is therefore a fundamental problem both in mathematics and in application areas to find the solution sets of polynomial systems. The focus of this paper is to compare two fundamentally different approaches to computing and representing the solutions of polynomial systems: numerical homotopy continuation and symbolic computation. Several illustrative examples are considered, using the software packages Bertini and Singular.

Key words: Polynomial system, computational algebraic geometry, numerical computation, homotopy continuation, symbolic computation, primary decomposition.

1. Introduction

Systems of polynomial equations arise throughout mathematics and its application areas as a means of describing, restricting, encoding, or modeling various aspects of a given problem. The areas where polynomial systems have been utilized is incredibly diverse, touching nearly every branch of mathematics and nearly every branch of science and engineering. One is typically interested in understanding some feature of the common set of zeros of the system with the level of detail required being very much dependent on the context and source of the problem. At the most basic level, one would like to understand if the system is consistent, i.e., whether the set of solutions is empty or non-empty. If the solution set is non-empty, the next level is to decide whether the solution set is finite and, if so, to determine the number of solutions and to represent each solution (or each solution together with its multiplicity). If there are infinitely many solutions to the polynomial system, then the solution set can be decomposed into equidimensional components which can be further decomposed into irreducible components. At this level, the most basic questions range from determining the dimension of the largest component to representing each reduced, irreducible component together with a description of its degree, dimension, and multiplicity.

The focus of this paper is to compare two fundamentally different approaches to computing and representing the irreducible components of the common set of zeros of a system of polynomial equations with rational coefficients. The first approach, based on the method of homotopy continuation, is numerical in nature and leads to a representation of each irreducible component with a collection of “witness points” and basic discrete data. The witness points for an irreducible component consist of a collection of numerical approximations to generic points lying on the component while the discrete data includes the dimension and degree, along with some information about the multiplicity. The second approach is symbolic in nature and leads to a representation of each irreducible component with an ideal and basic discrete data. The ideal is described by a set of generators whose common zeros correspond to points on the irreducible component while the discrete data also includes the dimension and degree, along with some information about the multiplicity. Beyond the dimension, degree, and multiplicity, there are many other discrete and continuous invariants that can be considered useful as a description of the features of (ensembles of) irreducible components of a system of polynomial equations. While very interesting, these finer invariants go beyond the scope of this paper.

As described above, the aim of this paper is to compare two very different approaches to decomposing a polynomial system and to explore their strengths and weaknesses. In §2, we briefly discuss basic numerical algebraic geometry, leading to an explanation of the numerical irreducible decomposition algorithm in §3. In §4, we give a description of two algorithms that are used to symbolically compute a related algebraic decomposition. More specifically, this is the decomposition of the radical of the ideal I as an intersection of prime ideals which contain I. In §5, we compare the two different approaches on a collection of benchmark examples. In particular, we compare running times for homotopy-based numerical decomposition algorithms over $\mathbb{C}$ with running times for symbolic decomposition algorithms over $\mathbb{Q}$ for systems of polynomial equations with rational coefficients. For the numerical runs, we use the software package Bertini [BHSW06] on a single processor and also on a multiprocessor system with 64 cores. The symbolic runs are performed using the Singular software package [DGPS10] on a single processor.

Our findings make it clear that both symbolic and numerical based decomposition methods have strengths that can and should be combined to create improved and more flexible decomposition algorithms. Strengths of numerical based decomposition methods include parallelizability, parameter homotopies, and the ability to extract meaningful information from systems presented with floating point coefficients. Strengths of symbolic based decomposition methods include exact output, the ability to exploit certain special structure in a polynomial system, and the ability to obtain very fine invariants such as syzygies and cohomology modules. A fully integrated system that can take advantage of the strengths of each approach while avoiding the weaknesses is clearly a goal that should be pursued. However, such a hybrid system and the development of novel numeric-symbolic algorithms is beyond the scope of this paper and are the topic of future work.

2. Numerical Algebraic Geometry

Fix a system

$$f(z) = \begin{bmatrix} f_1(z) \\ \vdots \\ f_n(z) \end{bmatrix} \tag{1}$$

of n polynomials with $z = (z_1, \ldots, z_N) \in \mathbb{C}^N$.

The numerical methods of the field of *numerical algebraic geometry* may be used to compute accurate approximations of the complex solution set of $f(z)$. This includes both the set of isolated solutions (points) and a numerical analogue of the irreducible decomposition (positive-dimensional solution sets). Though these numerical methods sacrifice the certainty intrinsic to symbolic methods, they reliably produce useful solutions to problems that may be intractable with symbolic methods.

The principal notion of the field is homotopy continuation, which will produce a superset of all isolated solutions of a given system $f(z) = 0$ with $n = N$. The idea is to construct an easy to solve system $g(z) = 0$ and then glue $g(z)$ to $f(z)$ with a new parameter t, to form a homotopy function of N polynomials $H(z, t)$ in $N + 1$ variables, $(z, t) = (z_1, \ldots, z_N, t) \in \mathbb{C}^{N+1}$, such that

1. $H(z, 0) = f(z)$ and $H(z, 1) = g(z)$;
2. for each solution z^* of $g(z) = 0$ there is a real analytic path $s : (0, 1] \to \mathbb{C}^N$ such that $H(s(t), t) = 0$ for $t \in (0, 1]$ and $s(1) = z^*$;
3. for each $t^* \in (0, 1]$, the Jacobian of $H(z, t^*)$ is nonsingular; and
4. the finite set S of the limits $\lim_{t\to 0} s(t)$ of all the paths includes all isolated solutions of $f(z) = 0$.

There are many ways to construct the start systems $g(z)$ and to follow the solution paths to find the limits. See [Li03, SW05] for detailed developments of this theory. Among the various start system options, the Bertini software package provides total degree and multihomogeneous homotopies.

So far, we merely have existence of the path $s(t)$, i.e., we do not know the equations of $s(t)$. Whereas the underlying theory relies heavily on ideas from several complex variables and algebraic geometry, the computations primarily involve numerical linear algebra. Tracking the path is an exercise in numerical computation – primarily predictor-corrector methods – for which an extensive literature exists. The article [BHS11] describes how to make best use of such techniques in the context of polynomial homotopies.

Although item (3) above says that the path remains nonsingular for $t \in (0, 1]$, there is no guarantee that the conditioning of the Jacobian matrix (the matrix used in the numerical linear algebra computations fundamental to homotopy continuation) will be mild throughout the path. Ill-conditioned segments of the paths are short-lived, but they do show up, especially in larger problems. Thus, adaptive multiple-precision techniques [BHSW08, BHSW09] are essential. These methods are the key to Bertini's reliability in this regard. In fact, by using adaptive multiple precision arithmetic, the endpoint may be approximated to any desired accuracy.

Upon this foundation of homotopy continuation was built a set of algorithms to decompose the solution set of $f(z) = 0$ (with n not necessarily equal to N) into irreducible components, a decomposition often used in algebraic geometry that may be thought of as a natural refinement of a decomposition into connected components. This *numerical* irreducible decomposition relies very heavily on the repeated use of very similar homotopies and will be described in some detail in the next section. Several software programs carry out the computation of a set S containing the isolated solutions of $f(z) = 0$, e.g., Bertini [BHSW06], POLSYS_GLP [SMSW06], HOM4PS [LLT08], and PHCpack [Ver99], though only Bertini automatically computes the numerical irreducible decomposition with a single command.

3. A numerical algorithm for computing the components of an affine algebraic variety

Let $f(z)$ be a polynomial system of n polynomials on $\mathbb{C}^N$, as in (1), and let $V(f)$ denote the set of zeroes of $f(z)$, also known as an *algebraic set.* The irreducible decomposition is the breakup of $V(f)$ into irreducible components:

$$V(f) := \bigcup_{i=0}^{\dim V(f)} Z_i = \bigcup_{i=0}^{\dim V(f)} \bigcup_{j \in \mathcal{I}_i} Z_{i,j} \tag{2}$$

where the algebraic set Z_i is the pure i-dimensional part of $V(f)$, $\mathcal{I}_i$ is a finite possibly empty set of indices for the irreducible components in dimension i, and the $Z_{i,j}$ are irreducible algebraic sets with no $Z_{i,j}$ contained in any $Z_{i',j'}$ with $(i,j) \neq (i',j')$. These irreducible components $Z_{i,j}$ are the closures of the connected components of $V(f) \setminus Sing(f)$, where $Sing(f)$ is the set of singular points of $V(f)$. As will be explained in §4, there is a bijection between the set of irreducible components of the (geometric) solution set and the set of (algebraic) prime ideals in the prime decomposition of the radical of the ideal generated by the polynomials of $f(z)$.

For each i with Z_i nonempty, we know that there is a nonempty Zariski open subset U of the Grassmannian of $(N-i)$-dimensional affine linear subspaces of $\mathbb{C}^N$, such that for each subspace $L_i \in U$, $L_i \cap Z_i$ consists of $\deg Z_i$ points; similarly each $L_i \cap Z_{i,j}$ consists of $\deg Z_{i,j}$ points.

Given a system of polynomials $f(z)$, the *numerical* irreducible decomposition of $V(f)$ consists of a *witness set* $\mathcal{W}_{i,j}$ for each irreducible component $\mathcal{Z}_{i,j}$. A witness set for an irreducible component is a tuple $(f(z), L_i, W_{i,j})$ providing exactly the polynomial system and linear subspace that were used to find the witness point set $W_{i,j}$, which is simply the set of $\deg Z_{i,j}$ points of $L_i \cap Z_{i,j}$.

How is the computation of the numerical irreducible decomposition carried out? In practice, there are several variations on a basic theme. Conceptually, the main two steps are:

1. Computation of a witness set of each nonempty Z_i, i.e., computation of L_i and the $\deg Z_i$ points $\mathcal{Z}_i = Z_i \cap L_i$.
2. Computation of the breakup of $\mathcal{Z}_i$ into the sets $W_{i,j}$.

To carry out Step 1, we choose L_i at random by forming a system of i linear equations, $\mathcal{L}_i$, whose coefficients are chosen at random from $\mathbb{C}$, so $L_i = V(\mathcal{L}_i)$. Then we compute a finite set $\widehat{\mathcal{Z}}_i \subset V(f) \cap L_i$ containing the set $\mathcal{Z}_i$. In the case that system f has more than $N - i$ equations, it is randomized down to a system of $N - i$ equations as

$$A \cdot f(z),$$

where A is a random $(N-i) \times n$ matrix. Here it should be noted that if $k \leq N$ equations vanish on a nonempty set, then all components of the set are at least $N - k$ dimensional. For sufficiently general A, all i-dimensional components of $V(f)$ are also components of $V(A \cdot f(z))$, though the randomized system may have new components as well (which can easily be recognized since the points on these new components will not satisfy the original system).

The major task of Step 1 is to solve the system

$$\begin{bmatrix} \mathcal{L}_i(z) \\ A \cdot f(z) \end{bmatrix} = 0$$

of N equations in N unknowns where $\mathcal{L}_i$ are linear equations with $V(\mathcal{L}_i) = L_i$. Although this exact approach was done in [SW95], a more efficient algorithm, known as the *cascade* approach, was introduced in [SV00]. A further improvement is the regenerative cascade of [HSW11], though this technique will only produce nonsingular solutions and witness sets for reduced positive-dimensional components.

With the $\widehat{\mathcal{Z}}_i$ sets in hand, Step 1 is completed by extracting $\mathcal{Z}_i$ from $\widehat{\mathcal{Z}}_i$. That is, we have $\widehat{\mathcal{Z}}_i = \mathcal{Z}_i \cup J_i$, where the subset J_i (the i^{th} junk set) needs to be identified and culled out of $\widehat{\mathcal{Z}}_i$ to yield $\mathcal{Z}_i$. The set J_i is contained in

$$\bigcup_{k>i} Z_k.$$

From this is it clear that $J_{\dim V(f)} = \emptyset$, so a by-product of computing the $\widehat{\mathcal{Z}}_i$ is the determination of the dimension of $V(f)$ as the largest i for which $\widehat{\mathcal{Z}}_i$ is not empty.

To see how we identify J_i, fix i, let z^* be a point of $\widehat{\mathcal{Z}}_i$, and assume that we have computed $\mathcal{Z}_k$ for $k > i$. For each $k > i$ find a family of general linear spaces of codimension k starting at L_k and ending at a general linear space containing z^*. If $z^* \in Z_k$, then the limits of the paths starting at the $\mathcal{Z}_k$ will contain the point z^* in the limit. This was introduced in [SVW01]. A newer approach for computing J_i, which is often more efficient (particularly if you are only interested in the sets $\mathcal{Z}_i$ for small i) is presented in [BHPS09].

Having computed the $\mathcal{Z}_i$, Step 2 consists of breaking these up according to irreducible components. This step has developed over the past 20 years so that the current two-phase approach is typically quite efficient. In the first phase, the point on the Grassmannian corresponding to L_i is moved in a loop on the Grassmannian. Some such loops may pick up a monodromy action, i.e., the same points will be present before and after the loop, but their order may change. Points of $\mathcal{Z}_i$ only pass to one another if they lie on the same irreducible component, so these *monodromy loops* will provide a partial decomposition of the points of $\mathcal{Z}_i$ into subsets corresponding to irreducible components.

It is typically not clear when or if the monodromy method is finished (since these monodromy loops can result in no changes in the ordering of the points, even if the decomposition is incomplete). Thus, it is necessary to rely on the second phase of Step 2 to complete the equidimensional decomposition. This second phase is called the *trace test.* We assign to each point of $\mathcal{Z}_i$ a complex number called the trace, denoted t_k for point $z_k \in \mathcal{Z}_i$, with the property that

$$\sum_{k \in \Lambda} t_k = 0$$

if and only if the points $\{z_k, k \in \Lambda\}$ form complete irreducible components. Thus, the trace test may be used to certify whether the groupings produced by monodromy are complete. Full details may be found in [SW05].

We need two final comments regarding the numerical irreducible decomposition of an algebraic set, since both deflation and regeneration play a role in the runs later in the article. When an irreducible component $Z_{i,j}$ has generic multiplicity greater than 1, then the corresponding points $W_{i,j}$ in the witness set $\mathcal{Z}_i$ are singular. That is, the Jacobian matrix of partial derivatives of f evaluated at such points has rank less than $N - i$. A preparatory step to computing the break-up is to first regularize the singular points.

Singular points can be regularized through a process called *deflation.* Deflation for isolated solutions was introduced in [OWM83, Oji87] with a proof of termination provided in [LVZ06] (see also [DZ05, LVZ08, HSW10]). Deflation was extended from isolated solutions to irreducible components in [SW05, §13.3.2,§15.2.2]. Though deflation is generally quite expensive, it is a necessary evil to allow for the numerical handling of singular solutions.

Finally, there are also equation-by-equation methods for solving polynomial systems, i.e, we can compute the isolated solutions of $f(z)$ as the intersection of the solution sets of the sets $V(f_i)$. Regeneration [HSW10] is currently the most efficient equation-by-equation algorithm, though it cannot be used to find singular solutions. For large systems, it is often more efficient than other algorithms for finding all isolated nonsingular solutions.

4. Symbolic methods

Given a system $f(z)$ of n polynomials

$$f_i \in \mathbb{C}[z] = \mathbb{C}[z_1, \dots, z_N]$$

as in (1), algebraic geometry allows us to study the set of zeros $Z = V(f)$ by algebraic means (we refer to [GP08] for details on what follows). On the algebraic side, rather than considering a specific set of polynomials defining Z, we consider the collection $I(Z)$ of all polynomials vanishing on Z. This collection contains the zero polynomial and is closed under taking linear combinations of its elements, with polynomials in $\mathbb{C}[z]$ as coefficients. In the terminology of commutative algebra, this means that $I(Z)$ is an *ideal*. With regard to the ideal

$$I = \langle f_1, \dots, f_n \rangle := \left\{ \sum_{i=1}^{n} g_i f_i \;\middle|\; g_i \in \mathbb{C}[z] \right\}$$

generated by the original polynomials f_i, Hilbert's celebrated Nullstellensatz tells us that $I(Z)$ is obtained from I by taking the *radical*:

$$I(Z) = \sqrt{I}, \;\text{ where }\; \sqrt{I} := \{ f \in \mathbb{C}[z] \mid f^m \in I \;\text{ for some }\; m \geq 1\}.$$

Algebraic geometers make use of the fact that geometric properties of Z correspond to algebraic properties of $I(Z)$. For example, the dimension of Z equals the *Krull dimension* of $I(Z)$. Or, Z is irreducible if and only if $I(Z)$ is a *prime ideal*; furthermore, if Z is reducible, its break-up into irreducible components corresponds to a decomposition of $I(Z)$ as an intersection of prime ideals. On the algebraic side, solving the decomposition problem of a set of zeros hence means to solve the following problem: Given a proper ideal I of the polynomial ring $\mathbb{C}[z]$, find prime ideals $P_1, \dots, P_s \in \mathbb{C}[z]$ such that

$$\sqrt{I} = P_1 \cap \dots \cap P_s, \;\text{ with }\; P_i \not\subset P_j \;\text{ for }\; i \neq j; \tag{3}$$

here, the P_i are uniquely determined (up to order) and are called the *minimal associated primes* of I. In addition to finding the P_i, we may wish to compute their dimensions and degrees by algebraic means.

The decomposition (3) is called the *prime decomposition* of $\sqrt{I}$. More generally, rather then decomposing $\sqrt{I}$, we may write the ideal I as an intersection of ideals which are not necessarily prime, but whose radical is prime: A *primary decomposition* of I is a decomposition $I = Q_1 \cap \dots \cap Q_t$ into *primary ideals* Q_i, where $P_i = \sqrt{Q_i} \neq P_j = \sqrt{Q_j}$ for all $i \neq j$, and such that no Q_i can be omitted. Then, though this is not necessarily true for the Q_i, the P_i are uniquely determined and are called the *associated primes* of I. The minimal associated primes of I are precisely the associated primes of I which are minimal with respect to inclusion.

Algebraically, the workhorse behind almost every problem arising in computational algebraic geometry is Buchberger's algorithm for computing Gröbner bases. These bases are special sets of generators for ideals in polynomial rings which depend on the choice of a well-ordering on the semigroup of monomials in that ring, and which are well-suited for computational purposes. Note that choosing a monomial ordering $>$ such as the *lexicographical ordering* allows us to speak of the *leading monomial* or *leading coefficient* of any given polynomial, and of the *leading ideal* of any given ideal I. A Gröbner basis of I is a finite set $g_1, \dots, g_r$ of generators for I such that the leading monomials of the g_i generate the leading ideal of I. By considering leading ideals, problems regarding arbitrary ideals can be reduced to problems regarding ideals generated by monomials which are usually much easier. Once the ideal and the monomial ordering are fixed, there is a uniquely determined *reduced Gröbner basis*, where reduced means that certain minimality conditions are fulfilled.

Gröbner basis techniques can be used, for example, to compute the intersection $I \cap J$ of two ideals in a polynomial ring R with coefficients in a field, the *ideal quotient*

$$I : J := \{f \in R \mid fg \in I \text{ for all } g \in J\}$$

of I by J, or the *saturation*

$$I : J^\infty := \{f \in R \mid fJ^m \subset I \text{ for some } m \geq 1\} = \bigcup_{m=1}^{\infty} (I : J^m)$$

of I with respect to J. By the Noetherian property of R, and since $I : J^{k+1} = (I : J^k) : J$ for each k, the computation of $I : J^\infty$ just means iterating the computation of ideal quotients until it stabilizes, i.e., $I : J^m = I : J^{m+1}$ for some m.

In our context here, Gröbner basis techniques settle the problem of finding the dimension and the degree. To compute the (minimal) associated primes, however, these techniques (or other means of manipulating ideals in polynomial rings) have to be combined with algorithms for polynomial factorization. Before discussing this in more detail, we need to make a remark on the ground field. Exact computer algebra calculations require that the arithmetic operations of the field under consideration can be implemented in an exact way. Thus, these calculations are carried through over fields such as the field of rational numbers, finite prime fields, or algebraic extensions thereof. Note that if we are given a system of polynomial equations f with rational coefficients, then the decomposition of $V(f)$ over $\mathbb{Q}$ may differ from that over $\mathbb{C}$ in that polynomials (algebraic sets) which are irreducible over $\mathbb{Q}$ may be reducible over $\mathbb{C}$. Factoring a polynomial (decomposing an algebraic set) over $\mathbb{C}$ then means to find an appropriate algebraic field extension of $\mathbb{Q}$ over which the *absolute factorization* (*absolute decomposition*) occurs, and to compute the irreducible factors (components) over that field. On the other hand, in many examples of interest, calculations over $\mathbb{Q}$ provide a reliable picture of what is happening over $\mathbb{C}$. In fact, it is often even sufficient to work over a finite prime field. In the latter way, the well-known problem of intermediate coefficient swell over the rationals can be avoided.

In the following two subsections, given a field K and a proper ideal I in the polynomial ring $K[z] = K[z_1, \dots, z_N]$, we sketch two algorithms for computing the minimal associated primes of I. Either algorithm proceeds by appropriately "splitting" $\sqrt{I}$ as an intersection of "simpler" ideals which are either prime or can be handled recursively. See [DGP99, GP08] for details and proofs.

4.1. The algorithm of Gianni, Trager, and Zacharias

To split $\sqrt{I}$, the algorithm of Gianni, Trager, and Zacharias (see [GTZ88]) first makes use of Gröbner bases to reduce to the zero-dimensional case, and then relies on properties of *zero-dimensional ideals in general position* to reduce to polynomial factorization. If a zero-dimensional ideal occurring in the process is not in general position, a generic coordinate transformation is required.

Before discussing some details, we regard the notion of Krull dimension from a computational point of view. For this, we think of $z = \{z_1, \ldots, z_N\}$ as a set of variables, and consider subsets u of z.

Definition 4.1.1. *A* maximal independent set of variables *with respect to a given ideal $I \subset K[z]$ is a subset of variables $u \subset z$ which is of maximal cardinality with the property that $I \cap K[u] = \{0\}$.*

In the situation above, the Krull dimension of I is precisely the cardinality of a maximal independent set of variables with respect to I. In particular, I is zero-dimensional if and only if $I \cap K[z_i] \neq \langle 0 \rangle$ for all i. In view of Gröbner bases, we can go one step further: The Krull dimension of I is the cardinality of a maximal independent set of variables with respect to the leading ideal of I. This reduces the computation of dimension to a purely combinatorial problem. The zero-dimensional case is particularly easy:

Remark 4.1.2. *To detect whether a given ideal $I \subset K[z]$ is zero-dimensional, compute a Gröbner basis of I. Then I is zero-dimensional if and only if for each i, $1 \leq i \leq N$, there is some $m_i \geq 0$ such that $z_i^{m_i}$ occurs as the leading monomial of a Gröbner basis element.*

We are now ready to describe the decomposition of zero-dimensional ideals. If I is such an ideal, by the very definition of Krull dimension, each associated prime of I is a maximal ideal. In particular, every associated prime of I is a minimal associated prime. We start with the following simple fact:

Lemma 4.1.3. *Let $I \subset K[z]$ be a zero-dimensional ideal, let g be a monic generator of the principal ideal $I \cap K[z_N]$, and let $g = g_1^{\nu_1} \cdots g_s^{\nu_s}$ be the factorization of g, where the g_i are monic and irreducible, with $g_i \neq g_j$ for $i \neq j$. Then*

$$I = \bigcap_{i=1}^{s} \langle I, g_i^{\nu_i} \rangle. \tag{4}$$

Next, it turns out that (4) is a primary decomposition whenever I is in *general position.* Essentially, this means that each associated prime P of I is in general position, which in turn means that P is of type $\langle z_1 + g_1(z_N), \ldots, z_{N-1} + g_{N-1}(z_N), g_N(z_N) \rangle$, with polynomials $g_1, \ldots, g_N \in K[z_N]$. Furthermore, the condition that I is in general position can be achieved by a coordinate transformation. In making this precise, we use the following notation:

Definition 4.1.4. *For $a = (a_1, \ldots, a_{N-1}) \in K^{N-1}$, let the coordinate transformation $\varphi_a : K[z] \to K[z]$ be defined by*

$$\varphi_a(z_i) = z_i \;\; \textit{if} \;\; i < N \;\; \textit{and} \;\; \varphi_a(z_N) = z_N + \sum_{i=1}^{N-1} a_i z_i.$$

In Proposition 4.1.5 below, we use the superscript (a) to indicate that the polynomials under consideration depend on a.

Proposition 4.1.5 (Shape Lemma). *Assume that K has characteristic zero. Let $I \subset K[z]$ be a zero-dimensional ideal. Then there is a non-empty Zariski open subset $U \subset K^{N-1}$ such that the following holds for each $a \in U$: There are $g_1^{(a)}, \ldots, g_N^{(a)} \in K[z_N]$, with $g_N^{(a)}$ square-free, and such that $\varphi_a(\sqrt{I}) = \langle z_1 + g_1^{(a)}, \ldots, z_{N-1} + g_{N-1}^{(a)}, g_N^{(a)} \rangle$. Furthermore, if $g_N^{(a)} = f_1^{(a)} \cdots f_{r_a}^{(a)}$ is the decomposition into irreducible factors, then $\{\varphi_a^{-1}(\langle z_1 + g_1^{(a)}, \ldots, z_{N-1} + g_{N-1}^{(a)}, f_i^{(a)} \rangle) \mid i = 1, \ldots, r_a\}$ is the set of minimal associated primes of I.*

Remark 4.1.6. *The proposition is also true in characteristic $p > 0$ if p is large compared to the degree of the given generators for I. In fact, if Algorithm 1 below terminates in characteristic p, then it returns a correct result.*

To compute the associated primes of a zero-dimensional ideal $I \subset K[z]$, the above discussion suggests to apply a coordinate transformation φ_a chosen at random, if necessary, and then use polynomial factorization. However, to make this work, we need to be able to decide whether the random choice of a made by the computer was indeed general enough. Here, we use a criterion which allows us to check wether a given ideal is a zero-dimensional primary ideal in general position. Rather than stating this criterion, we exemplify its usage in Algorithm 1 below, where we will just return the associated primes and not the primary ideals. In formulating the algorithm (and in subsequent statements), given an ideal $I \subset K[z]$ and a polynomial $f \in K[z]$, we write $\langle I, f \rangle$ for the ideal generated by the elements of I and f.

It remains to explain how to reduce the higher-dimensional case to the zero-dimensional case. The basis for this is Proposition 4.1.7 below. In stating the proposition, given a subset of variables $u \subset z$ and its complement $z \smallsetminus u$, we write $K(u)$ for the field of rational functions in the u variables; furthermore, given an ideal I of $K[z]$, we write $IK(u)[z \smallsetminus u]$ for the ideal generated by I in the polynomial ring with coefficients in $K(u)$ and variables in $z \smallsetminus u$.

Proposition 4.1.7. *Let $I \subset K[z]$ be an ideal, and let $u \subsetneq z$ be a maximal independent set of variables with respect to I. Then there exists $h \in K[u]$ such that*

1. *$I = (I : \langle h \rangle) \cap \langle I, h \rangle$, and*
2. *$IK(u)[z \smallsetminus u] \cap K[z] = I : \langle h \rangle = I : \langle h \rangle^\infty$.*

Algorithm 1 zeroMinAssGTZ (I)

```
Input: A zero-dimensional ideal $I \subset K[z]$ given by a finite set of generators.
Output: The associated primes of $I$.
compute $S$, the reduced Gröbner basis of $I$ with respect to the lexicographical ordering, taking $z_1 > \ldots > z_N$
factor the element $g \in S$ with smallest leading monomial: $g = g_1^{\nu_1} \cdots g_s^{\nu_s}$
if ($s > 1$) then
    return $\bigcup_{i=1}^{s}$ zeroMinAssGTZ ($\langle I, g_i \rangle$)
Prime := $\langle g_1 \rangle$
$i := N$
while ($i > 1$) do
    $i := i - 1$
    choose $f \in S$ with leading monomial of type $z_i^m$
    $b :=$ the coefficient of $z_i^{m-1}$ in $f$ considered as a polynomial in $z_i$
    $q := z_i + \frac{b}{m}$
    if ($q^m \equiv f \mod$ Prime) then
        Prime := Prime $+\langle q \rangle$
    else
        choose $a \in K^{N-1}$ at random
        return $\varphi_a^{-1}$(zeroMinAssGTZ ($\varphi_a(I)$))
return Prime
```

Furthermore, $IK(u)[z \smallsetminus u]$ is zero-dimensional, and if $IK(u)[z \smallsetminus u] = \bigcap_{i=1}^{s} Q_i$ is a primary decomposition, with associated primes $P_1, \ldots, P_s$, then $I : \langle h \rangle = \bigcap_{i=1}^{s}(Q_i \cap K[z])$ is a primary decomposition, with associated primes $P_1 \cap K[z]$, $\ldots, P_s \cap K[z]$.

In fact, the proof of the proposition gives a constructive way of finding the polynomial h. This will be exemplified in Algorithm 2. Combining Algorithms 1 and 2, we get Algorithm 3 which computes the minimal associated primes for ideals which are not necessarily zero-dimensional.

Remark 4.1.8. *The inclusion $I \subset \langle I, h \rangle$ in Step 4 of Algorithm 3 is proper since $I \cap K[u] = \{0\}$ but $h \in K[u]$. Hence, by the* ascending chain condition *in the* Noetherian ring *$K[z]$, the algorithm must terminate, provided that Algorithm 1 terminates. Algorithm 1, in turn, terminates by a similar argument, provided we can find coordinate transformations which are defined over K and are general enough. This is guaranteed to work in characteristic zero but may fail in positive characteristic.*

4.2. Minimal associated primes via characteristic sets

The concept of characteristic sets goes back to Ritt [Rit32, Rit50] and Wu [Wu84]. In our context, it yields an algorithm which first applies successive pseudo-divisions to construct a characteristic set $\mathcal{F}$, and then makes use of $\mathcal{F}$ to split $\sqrt{I}$. There are two types of splitting, depending on whether $\mathcal{F}$ is

Algorithm 2 reductionToZero (I)

```
Input: $I := \langle f_1, \ldots, f_k \rangle \subset K[z]$.
Output: A list $(u, G, h)$, where
    - $u \subset z$ is a maximal independent set of variables with respect to $I$,
    - $G = \{g_1, \ldots, g_s\} \subset I$ is a Gröbner basis of $IK(u)[z \smallsetminus u]$,
    - $h \in K[u]$ such that $IK(u)[z \smallsetminus u] \cap K[z] = I : \langle h \rangle = I : \langle h \rangle^\infty$.
compute a maximal independent set $u \subset z$ with respect to $I$
compute a Gröbner basis $G = \{g_1, \ldots, g_s\}$ of $I$ with respect to the lexicographical ordering, with any variable in $z \smallsetminus u$ taken greater than any variable in $u$
$h :=$ product of the leading coefficients of the $g_i$ considered as polynomials in $z \smallsetminus u$ with coefficients in $K(u)$
compute $m$ such that $\langle g_1, \ldots, g_s \rangle : \langle h \rangle^m = \langle g_1, \ldots, g_s \rangle : \langle h \rangle^{m+1}$
return $u, \{g_1, \ldots, g_s\}, h^m$
```

Algorithm 3 minAssGTZ (I)

```
Input: $I := \langle f_1, \ldots, f_k \rangle \subset K[z]$.
Output: The list of minimal associated primes of $I$.
$(u, G, h) =$ reductionToZero ($I$)
change ring to $K(u)[z \smallsetminus u]$ and compute
    PrimesZero := zeroMinAssGTZ ($\langle G \rangle_{K(u)[z \smallsetminus u]}$)
change ring to $K[z]$ and compute
    Primes:= $\{P \cap K[z] \mid P \in$ PrimesZero$\}$
return Primes $\cup$ minAssGTZ ($\langle I, h \rangle$)
```

irreducible or not. Computational tools involved are polynomial factorization over an appropriate extension field of K and, in the irreducible case, Gröbner bases.

To present some details, we need the following notation:

Definition 4.2.1. *Let f be a polynomial in $K[z] = K[z_1, \dots, z_N]$.*

1. *The* class *of f, written* $\mathrm{cl}(f)$*, is the maximal k such that f actually depends on z_k. This number is zero if f is constant.*
2. *If* $\mathrm{cl}(f) \neq 0$*, the* initial *of f, written* $\mathrm{in}(f)$*, is the leading coefficient of f considered as a polynomial in $z_{\mathrm{cl}(f)}$.*
3. *If $f \neq 0$, a polynomial $g \in K[z]$ is* Ritt-Wu reduced with respect to f *if* $\deg_{z_{\mathrm{cl}(f)}}(g) < \deg_{z_{\mathrm{cl}(f)}}(f)$.

Definition 4.2.2. *A set of polynomials $\mathcal{F} = \{f_1, \dots, f_r\} \subset K[z]$ is called an* ascending set *if either*

1. *$r = 1$ and $f_1 \neq 0$, or*
2. *$r > 1$, $0 < \mathrm{cl}(f_1) < \cdots < \mathrm{cl}(f_r)$, and each f_i, $i = 2, \dots, r$, is Ritt-Wu reduced with respect to $f_1, \dots, f_{i-1}$.*

The basic computational tool here is *pseudo-division*: Given $f, g \in K[z]$ with $k := \mathrm{cl}(f) > 0$, this yields a unique expression

$$\mathrm{in}(f)^\alpha g = qf + h\,, \text{ with } \deg_{z_k}(h) < \deg_{z_k}(f)$$

and $\alpha := \max\{0, \deg_{z_k}(g) - \deg_{z_k}(f) + 1\}$. Then $\mathrm{prem}(g|f) := h$ is called the *pseudo-remainder* of g with respect to f. The *pseudo-remainder* of g with respect to an ascending set $\mathcal{F} = \{f_1, \dots, f_r\} \subset K[z]$ is inductively defined by $\mathrm{prem}(g|\mathcal{F}) := \mathrm{prem}(g|f_1, \dots, f_r) := \mathrm{prem}(\mathrm{prem}(g|f_2, \dots, f_r)|f_1)$. Note that g is Ritt-Wu reduced with respect to each f_i if and only if $\mathrm{prem}(g|\mathcal{F}) = g$.

Definition 4.2.3. *Let $X \subset K[z] \setminus \{0\}$ be a finite set of generators for I. An ascending set $\mathcal{F} = \{f_1, \dots, f_r\} \subset I$ is called a* characteristic set *for X, if either*

1. *$r = 1$ and f_1 is constant, or*
2. $\mathrm{cl}(f_1) > 0$ *and* $\mathrm{prem}(g|\mathcal{F}) = 0$ *for all $g \in X$.*

Remark 4.2.4. *With notation as above, there is a natural way of defining a well-founded ordering on the set of all ascending sets contained in X. Based on pseudo-division, there is an efficient algorithm* CharSet *which first computes a minimal element with respect to this ordering, and then extends the minimal element to a characteristic set.*

To explain the two types of splitting via a characteristic set, we need:

Definition 4.2.5. *Let $\mathcal{F} = \{f_1, \dots, f_r\} \subset K[z] = K[z_1, \dots, z_N]$ be an ascending set, and let $m = N - r$. After renaming the variables, we may assume that* $\mathrm{cl}(f_i) = z_{m+i}$*, $i = 1, \dots, r$. With this assumption, $\mathcal{F}$ is called* irreducible *if each f_i is irreducible in $K_i[z_{m+i}]$, where K_i is inductively defined by $K_1 := K(z_1, \dots, z_m)$, and $K_i := K_{i-1}[z_{m+i-1}]/\langle f_{i-1}\rangle$.*

Proposition 4.2.6. *Let $\mathcal{F} = \{f_1, \ldots, f_r\} \subset K[z]$ be an irreducible ascending set. Then*

$$P_{\mathcal{F}} := \{g \in K[z] \mid \mathrm{prem}(g|\mathcal{F}) = 0\}$$

is a prime ideal. Furthermore, if $X \subset K[z] \setminus \{0\}$ is a finite set of generators for I, and $\mathcal{F}$ is an irreducible characteristic set for X, then

$$\sqrt{I} = P_{\mathcal{F}} \cap \sqrt{\langle X \cup \{\mathrm{in}(f_1)\}\rangle} \cap \cdots \cap \sqrt{\langle X \cup \{\mathrm{in}(f_r)\}\rangle}\,.$$

Remark 4.2.7. *If $\mathcal{F} = \{f_1, \ldots, f_r\} \subset K[z]$ is an irreducible ascending set, then*

$$P_{\mathcal{F}} = (\ldots((\langle \mathcal{F}\rangle : \langle \mathrm{in}(f_1)\rangle^{\infty}) : \langle \mathrm{in}(f_2)\rangle^{\infty}) : \cdots) : \langle \mathrm{in}(f_r)\rangle^{\infty}\,.$$

Hence $P_{\mathcal{F}}$ can be computed via Gröbner bases.

Remark 4.2.8. *Let $\mathcal{F} = \{f_1, \ldots, f_r\} \subset K[z]$ be an ascending set. With notation and assumption as in Definition 4.2.5, suppose that $\mathcal{F}$ is reducible. Choose i minimal with $\{f_1, \ldots, f_i\}$ reducible. Let $f_i = h_1^{\rho_1} \cdots h_s^{\rho_s}$ be the factorization of f_i into irreducible factors in $K_i[z_{m+i}]$. Then, by clearing denominators in the h_j and subsequently computing pseudo-remainders with respect to $\{f_1, \ldots, f_{i-1}\}$, we get polynomials $g_j \in K[z]$ which are Ritt-Wu reduced with respect to $\mathcal{F}$ and satisfy $\mathrm{cl}(g_j) = \mathrm{cl}(f_i)$. Furthermore, there is an expression of type $\mathrm{in}(f_1)^{s_1} \cdots \mathrm{in}(f_{i-1})^{s_{i-1}} g_1^{p_1} \cdots g_s^{p_s} \in \langle f_1, \ldots, f_i\rangle$.*

Summing up, we get Algorithm 4.

5. Comparison and timings

We have chosen a series of five examples that are easy to create automatically. All but the n-point problem can be made arbitrarily large.

The problems are:

1. the problem of adjacent minors (§5.1):
   (a) 2×2 minors in a $3 \times n$ matrix;
   (b) 3×3 minors in a $4 \times n$ matrix;
2. the KSS-Wright systems (§5.2);
3. the Sudoku problem (§5.3);
4. the n-point problem for $n = 3$ and $n = 9$ (§5.4); and
5. the Stevenson-pattern planar structures (§5.5).

The timings reported below used a cluster of Xeon 5410 cores (2.33 GHz) running Linux. We used Bertini version 1.3.0 and Singular version 3-1-3. The timings are reported in seconds as returned by Singular or by the Linux `time` command, rounded to the nearest second, for Bertini. The Singular timings are reported under the headings GTZ (Gianni-Trager-Zacharias [GTZ88]) and SY (Shimoyama-Yokoyama [SY96]). The Bertini timings are averages over five runs since different choices of random numbers will result in slightly different run times. The Bertini runs used regeneration in all cases except the Stephenson-pattern structures, where a classic 2-homogeneous homotopy was used instead.

Some discussion of these results may be found in §6.

Algorithm 4 MinAssPrimesCharSets (I)

Input: An ideal $I \subsetneq K[z]$ given by a finite set of generators $X \subset K[z] \setminus \{0\}$.
Output: The minimal associated primes of I.

```
Result := $\emptyset$
Rest := $\{X\}$
**while** Rest $\neq \emptyset$ **do**
  choose $X \in$ Rest
  Rest:= Rest $\setminus\{X\}$
  $\mathcal{F}$ := CharSet$(X)$
  **if** $\mathcal{F} = \{f\}$ with $f \in K$ **then**
    **return** $\langle 1 \rangle$
  **else**
    **if** $\mathcal{F}$ is irreducible **then**
      Result := Result $\cup \{\mathcal{F}\}$
      Rest := Rest $\cup \{X \cup \mathcal{F} \cup \{\mathrm{in}(f)\} \mid f \in \mathcal{F}, \mathrm{cl}(\mathrm{in}(f)) > 0\}$
    **else**
      find $f_1, \ldots, f_{i-1} \in \mathcal{F}$ and $g_1, \ldots, g_s$ as in Remark 4.2.8
      Rest := Rest $\cup \{X \cup \mathcal{F} \cup \mathrm{in}(f_j) \mid j = 1, \ldots, i-1\}$
                   $\cup \{X \cup \mathcal{F} \cup \{g_j\} \mid j = 1, \ldots, s, \mathrm{cl}(\mathrm{in}(f_j)) > 0\}$
write Result $= \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$
**for** $i = 1$ to $k$ **do**
  $J := \langle \mathcal{F}_i \rangle$
  **for** $f \in \mathcal{F}_i$ **do**
    $J := J : \langle \mathrm{in}(f) \rangle^\infty$
  Result := $\big($Result $\setminus\{\mathcal{F}_i\}\big) \cup \{J\}$
discard redundant prime ideals in Result
**return** Result
```

5.1. Adjacent Minors

Our first family of problems is the adjacent $k \times k$ minors of an $m \times n$ matrix of indeterminants. Here, adjacent means that the k column and k row indices are consecutive sequences of numbers. The advantage of the system is that it is easy to count the number of components of a given degree and dimension *a priori*, so we know in advance what output to expect.

For example, consider the 2×2 adjacent minors of a $2 \times n$ matrix For each tuple $d = (d_1, \ldots, d_s)$ of positive integers with $d_1 + \cdots + d_s + s = n+1$, there exists a prime component of degree $d_1 \cdots d_s$ defined by the following sum of ideals. For each $t = 1, \ldots, s$, consider the indices $\ell_t = d_1 + \cdots + d_t + t$. Define the ideal $I_{d,t}$ as the ideal generated by the 2×2 minors of the matrix consisting of the $\ell_{t-1}+1$ to $\ell_t - 1$ columns of the original matrix. For $t = 1, \ldots, s-1$, define the ideals $L_{d,t}$ as the ideal generated by entries of the ℓ_t. Then, $I_d = \sum_{t=1}^{s} I_{d,t} + \sum_{t=1}^{s-1} L_{d,t}$ is a prime ideal of codimension $n - 1 = 2(s-1) + \sum_{t=1}^{s}(d_t - 1)$ and degree $\prod_{t=1}^{s} d_t$ of codimension $n-1$. We leave it to the reader to check the primary decomposition of the ideal of adjacent minors $A_{2,2,n} = \cap_d I_d$. See [Stu02, HS04] (and others) for more details.

$A_{2,m,n}$, for $3 \leq m \leq n$, are radical ideals with primary components of various dimensions which have a combinatorial description similar to, though more complicated than, the one above. For example, $A_{2,3,5}$ has 11 components of codimension and degree as follows: 1 codimension 5 component of degree 1, 3 codimension 6 components all of degree 1, 6 codimension 7 components, two of degree 6, one of degree 4, two of degree 2 and one of degree 1, 1 codimension 8 component of degree 15.

We conjecture that 2×2 adjacent minors always define radical ideals.

n	GTZ	SY	Bertini	Bertini with 64 cores
3	0	0	0	3
4	0	0	2	4
5	1	0	13	6
6	3	1	71	11
7	19	3	370	23
8	154	22	1935	63
9	1430	170	9871	248
10	14294	1429	51098	1132
11	> 24 hours	12709	> 24 hours	6823

Table 1: Comparison of Singular GTZ and SY methods, Bertini on 1 core, and Bertini on 64 cores on adjacent minors systems with $k = 2, m = 3$: in seconds unless otherwise marked.

5.2. The Wright-KSS systems

Our next family is an example of a zero-dimensional ideal with a high-multiplicity solution [KSS98, Wri85]. Wright constructed the first example to test his early version of a continuation algorithm. His example is case $n = 5$ in

n	GTZ	SY	Bertini	Bertini with 64 cores
4	0	70	4	6
5	111	> 24 hours	49	9
6	> 24 hours	> 24 hours	527	39
7	> 24 hours	> 24 hours	6548	157
8	> 24 hours	> 24 hours	78362	1958

Table 2: Comparison of Singular GTZ and SY methods, Bertini on 1 core, and Bertini on 64 cores on adjacent minors systems with $k = 3, m = 4$: in seconds unless otherwise marked.

the sequence of systems defined in [KSS98], a straightforward generalization of Wright's problem. The systems are defined by n equations in n unknowns:

$$z_k^2 + \sum_{i=1}^{n} z_i - 2z_k - (n-1) = 0, \qquad k = 1, 2, \ldots, n.$$

Due to the high-multiplicity, numerical methods must use adaptive-precision tracking [BHSW08, BHSW09] which increases the cost. Again this example can be analyzed without a computer by rewriting the system (by subtracting two equations and factoring).

n	GTZ	SY	Bertini
5	0	0	1
6	0	0	1
7	1	1	1
8	1	3	9
9	7	37	18
10	71	203	21
11	1095	3488	94
12	18173	18892	160
13	> 24 hours	> 24 hours	972
14	> 24 hours	> 24 hours	1682
15	> 24 hours	> 24 hours	5892
16	> 24 hours	> 24 hours	8831
17	> 24 hours	> 24 hours	26925
18	> 24 hours	> 24 hours	42574
19	> 24 hours	> 24 hours	> 24 hours

Table 3: Comparison of Singular GTZ and SY methods, and Bertini on 1 core on Wright-KSS systems: in seconds unless otherwise marked.

5.3. Sudoku puzzles

Our next examples arise from Sudoku puzzles. A well-posed such puzzle has a unique solution. Once completed, a Sudoku is a 9×9 square grid consisting of nine distinguished 3×3 blocks whose entries are taken from the digits 1 through 9. Furthermore, the Sudoku is subject to the condition that each digit from 1 to 9 appears exactly once in each row, column, and distinguished 3×3 block.

To model the Sudoku conditions by polynomial equations, represent the 81 cells of the Sudoku by 81 variables, say $z_1, \ldots, z_{81}$. Then the entry a_i in the ith cell of the completed Sudoku satisfies $a_i \in \{1, \ldots, 9\}$ if and only if a_i is a root of the univariate polynomial $F_i = \prod_{k=1}^{9}(z_i - k) \in \mathbb{Q}[z_i]$. The polynomial $F_i(z_i) - F_j(z_j)$ vanishes on $V(z_i - z_j)$, so that $z_i - z_j$ is a factor of $F_i(z_i) - F_j(z_j)$, for $i \neq j$. We hence have well-defined polynomials

$$G_{ij}(z_i, z_j) = \frac{F_i - F_j}{z_i - z_j} \in \mathbb{Q}[z_i, z_j],\ i \neq j.$$

Now set

$$\begin{aligned} E = \{(i,j) \mid 1 \leq i < j \leq 81,\ &\text{and the } i\text{th and } j\text{th cell are in the} \\ &\text{same row, column, or distinguished } 3 \times 3 \text{ block}\}, \end{aligned}$$

and suppose that we are given a well-posed Sudoku puzzle with pre-assigned values $\{a_i\}_{i\in L}$, for some subset $L \subset \{1, \ldots, 81\}$. Consider the system defined by the polynomials F_i, $i = 1, \ldots, 81$, G_{ij}, $(i,j) \in E$, and $z_i - a_i$, $i \in L$. Then, with respect to any monomial well-ordering, the reduced Gröbner basis of I_S has the shape $z_1 - a_1, \ldots, z_{81} - a_{81}$, where $(a_1, \ldots, a_{81})$ is the solution of the Sudoku. See [DP12] for details.

In this way, Sudokus provide excellent examples of polynomial systems with a structure very well-suited to symbolic computation but poorly-suited to numerical computation. In fact, some of the equations in our setting are well-known examples of poorly conditioned equations of Wilkinson type [Wil59]. Furthermore, the system is overdetermined, so Bertini must "square" the system, thereby destroying the special structure. One instance of a Sudoku problem, for example, ran in 1274 seconds in Singular. Bertini required over 24 hours.

Let us point out, however, that attacking a Sudoku puzzle can be regarded as a graph coloring problem, with one color for each of the digits from 1 to 9, and that compared to what we discussed here, graph theory provides much more efficient methods for solving a given puzzle. See, for example, [HM07].

5.4. n-point four-bar design

These problems concern finding four-bar linkages whose coupler curve interpolates a number of given points. The maximum number of general points that can be interpolated exactly is nine, as this equals the number of design parameters in a four-bar linkage. The associated nine-point problem was first posed by Alt in 1923 [Alt23] and complete solution lists (for various sets of nine given points) were first computed in 1992 [WMS92, WMS97]. One can also consider interpolating fewer than nine points, in which case the solution sets are positive-dimensional, giving the designer the ability to choose a design that satisfies other concerns that may affect the suitability of the four-bar. We consider n points for $3 \leq n \leq 9$. In each case, we are only interested in the solution components of dimension $9-n$. Higher dimensional components exist, but for general points these are degenerate sets that contain no useful four-bars.

The definition of the n-point problem is as follows. We are given points in isotropic coordinates as $(p_i, \bar{p}_i) \in \mathbb{C}^2$, $i = 0, \ldots, n-1$. Kinematicians call these *precision points*, because we ask that the coupler curve pass exactly through them. Without loss of generality, we may translate coordinates to make $(p_0, \bar{p}_0) = (0,0)$. The variables of the problem are the linkage parameters $q = (a_1, a_2, b_1, b_2, \bar{a}_1, \bar{a}_2, \bar{b}_1, \bar{b}_2) \in \mathbb{C}^8$. Define the 2×1 column vectors $v, u, \bar{u}$ as

$$v = \begin{bmatrix} (p-a_1)(\bar{p}-\bar{a}_1) + b_1\bar{b}_1 - (b_1-a_1)(\bar{b}_1-\bar{a}_1) \\ (p-a_2)(\bar{p}-\bar{a}_2) + b_2\bar{b}_2 - (b_2-a_2)(\bar{b}_2-\bar{a}_2) \end{bmatrix}, \tag{5}$$

$$u = \begin{bmatrix} b_1(\bar{p}-\bar{a}_1) \\ b_2(\bar{p}-\bar{a}_2) \end{bmatrix}, \quad \text{and} \quad \bar{u} = \begin{bmatrix} \bar{b}_1(p-a_1) \\ \bar{b}_2(p-a_2) \end{bmatrix}. \tag{6}$$

Then the coupler curve equation is

$$f_{cc}(p, \bar{p}; q) = \det[v \;\; \bar{u}] \cdot \det[v \;\; u] + (\det[u \;\; \bar{u}])^2 = 0. \tag{7}$$

The n-point system is

$$f_{cc}(p_i, \bar{p}_i; q) = 0, \qquad i = 1, \ldots, n-1. \tag{8}$$

One may check that $f_{cc}(0,0;q) = 0$ for any q, as expected, so the coupler curve passes through the first point $(p_0, \bar{p}_0) = (0,0)$. It is easy to rotate any coupler curve around this point to make it pass through the second precision point. Thus the interesting cases begin at $n = 3$ and progress to $n = 9$. Each equation in the system is degree seven in the variables q, so the n-point problem has total degree $7^{(n-1)}$.

From [WMS92], for $n = 9$, we expect 8652 isolated roots that appear in a six-way symmetry group, meaning that the solutions appear in 1442 distinct orbits. Notice that 8652 is much smaller than the total degree of 7^8, so the problem has a very special structure compared to systems of general seventh-degree equations.

Singular did not terminate in under 24 hours for $n = 3$ or $n = 9$. On 1 core, Bertini took 24 seconds for $n = 3$ and 28018 seconds for $n = 9$. On 64 cores, the $n = 9$ case required 929 seconds with Bertini.

5.5. Stephenson-pattern planar structures

The Stephenson six-bar linkage is obtained by adding a dyad (two links in series) to a four-bar linkage, with one end of the dyad connected to the four-bar's coupler point and the other end connected to ground. The ungrounded link of this dyad becomes the new coupler link. The Stephenson-pattern $2n$-bar linkages continue this pattern by sequentially adding yet another dyad going from the previous coupler link to ground [Wun63]. Any of these one-degree-of-freedom (1DOF) linkages can be converted to a structure by adding a single final link from the last coupler link to ground. Thus, the n-th Stephenson-pattern planar structure is a mechanism having $2n+1$ links. Structures such as these, which do not contain any sub-mechanism that is also a structure are called Baranov trusses. The solution of any complicated structure can be broken down into solving a sequence of submechanisms that are each Baranov trusses [Man73]. The Baranov trusses are precisely those structures whose solution cannot be subdivided in this way.

The polynomial system for the n-th Stephenson-pattern structure written in isotropic coordinates is:

$$x_i\bar{x}_i = \ell_i^2, \qquad i = 0,\dots,n, \tag{9}$$

$$\theta_i\bar{\theta}_i = 1, \qquad i = 1,\dots,n-1, \tag{10}$$

$$a_0 + x_0 = a_1 + x_1 + b_1\theta_1, \tag{11}$$

$$a_i + x_i + c_i\theta_i = a_{i+1} + x_{i+1} + b_{i+1}\theta_{i+1}, \qquad i = 1,\dots,n-2, \tag{12}$$

$$a_{n-1} + x_{n-1} + c_{n-1}\theta_{n-1} = a_n + x_n, \tag{13}$$

$$\bar{a}_0 + \bar{x}_0 = \bar{a}_1 + \bar{x}_1 + \bar{b}_1\bar{\theta}_1, \tag{14}$$

$$\bar{a}_i + \bar{x}_i + \bar{c}_i\bar{\theta}_i = \bar{a}_{i+1} + \bar{x}_{i+1} + \bar{b}_{i+1}\bar{\theta}_{i+1}, \qquad i = 1,\dots,n-2, \tag{15}$$

$$\bar{a}_{n-1} + \bar{x}_{n-1} + \bar{c}_{n-1}\bar{\theta}_{n-1} = \bar{a}_n + \bar{x}_n. \tag{16}$$

Here, the parameters are $(\ell_i, a_i, \bar{a}_i)$, $i = 0,\dots,n$, and $(b_i, \bar{b}_i, c_i, \bar{c}_i)$, $i = 1,\dots,n-1$. The variables are $(x_i, \bar{x}_i)$, $i = 0,\dots,n$ and $(\theta_i\bar{\theta}_i)$, $i = 1,\dots,n-1$. One sees that we have $4n$ equations in $4n$ unknowns. Of these equations, $2n$ are quadratic and the other $2n$ are linear. Notice that (11–13) are of similar form to (14–16) except each symbol without an overbar is swapped for the same symbol with an overbar. The quadratics are all self-similar under this operation. Considering a partition of the variables into two groups, (x_i, θ_i) and $(\bar{x}_i, \bar{\theta}_i)$, all valid i, one easily sees that the linear equations respect the groups and the quadratic ones are bilinear. The total degree of the n-th system is 2^{2n}, while its 2-homogeneous root count is $\binom{2n}{n}$ and its actual root count is $2 \cdot 3^{n-1}$ [Wun63]. Case $n = 2$ is the pentad, having 6 solutions, case $n = 3$ is one of three possible Baranov septads, having 18 solutions [Inn95].

Since the problem is algebraic in its parameters, after solving the problem for case n for one set of random complex parameters, the solution for any subsequent example of that case can be found by numerically tracking the solutions as the parameters move along a path in $\mathbb{C}^{7n-1}$ from the first instance to the new set of parameters. The performance of this *parameter homotopy* is documented in Table 4 along with the symbolic runs and the numerical runs for solving the system from scratch. For this problem only, since we are looking for only isolated solutions, the numerical runs used a 2-homogeneous homotopy instead of regeneration.

6. Discussion

The timings of §5 clearly indicate that neither the symbolic methods nor the numerical methods considered in this article are always superior.

N	`min ass primes`	`lex GB`	`Bertini`	param	64 cores
4	2	676	1	0	2
8	> 24 hours	> 24 hours	1145	21	6
12	> 24 hours	> 24 hours	> 24 hours	3676	266

Table 4: Comparison of Singular **`min ass primes`** and **`lex GB`** methods, Bertini on 1 core with random parameter values, a parameter run with Bertini on 1 core, and the same on 64 cores, for the Stephenson-N problems with $N = 4, 8, 12$.

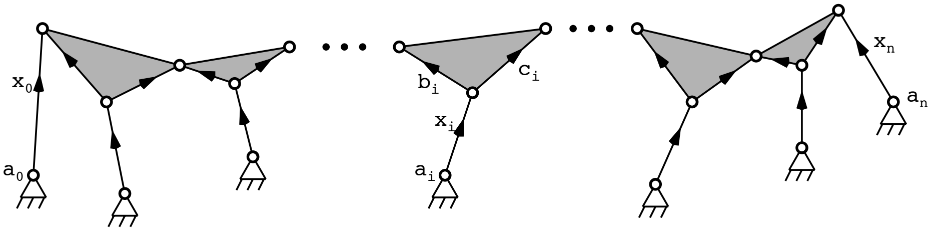

Figure 1: Stephenson-pattern structures. Label a_i means point $(a_i, \bar{a}_i)$, etc.

It may seem that certain classes of users might want to use one set of methods rather than the other. For example, for users interested in certified, exact solutions, symbolic methods are likely the preferred choice, though recent work on certification of numerical methods, e.g., [HS12], may soon provide a viable alternative.[1] Conversely, those satisfied with numerical output might tend to use numerical methods, but the Sudoku example indicates that such methods are not nearly as effective as symbolic methods for problems with a certain structure.

So what conclusions *can* the reader draw from this article. Here are a few:

1. Both numerical and symbolic methods have value, and a collaboration between the researchers in these two communities could be beneficial to the mathematicians, scientists, and engineers who need to solve polynomial systems.
2. One difference between the two sets of methods is clear. Symbolic methods produce exact output whereas numerical methods produce numerical approximations. Of course, as discussed above, if symbolic methods are not run over $\mathbb{Q}$, then one could argue that the exact computations of symbolic methods are not necessarily producing the output expected by the user.
3. Numerical methods tend to scale more nicely than symbolic methods for the examples of this paper, though this may not always be the case (see

[1] In fact, there may be an opportunity to combine numerical and symbolic methods to develop symbolically certified numerical methods. In particular, one could use the numerical methods of this paper to find solutions, convert them to exact solutions via exactness recovery techniques [BHMPS12], then certify these exact results with symbolic methods such as those in the paper.

the 2×2 adjacent minors example). Even if numerical methods do scale better, that does not necessarily mean that they will be faster for smaller problems (see the Sudoku example). Similarly, numerical methods are parallelizable, though there has been recent work on parallelizing symbolic methods, as well.

4. Problems with special structure, particularly those that are already (or are nearly) Gröbner bases, as with the 2×2 adjacent minors (each being a binomial polynomial) of a $3 \times n$ matrix, are particularly well-suited to symbolic computation. However, the timings for the 2×2 adjacent minors problems indicate that even this advantage does not always mean that such methods will necessarily be significantly faster.
5. For problems that are parameterized and need to be solved repeatedly for many different parameter values, parameter homotopies can be particularly effective, as demonstrated on the Stephenson-pattern problems. Of course, parameterized Gröbner bases could be a viable alternative.

As for future work in this vein, there are several clear directions. First and foremost, it is likely that there are currently undiscovered symbolic-numeric methods that will take advantage of the best of both worlds. For example, some symbolic computation may help to condition polynomial systems for numerical runs. Conversely, numerical runs might help to pinpoint the underlying geometric structure of the solution set, which could help the user to improve the efficiency of symbolic methods. Also, it would be interesting to compare polyhedral homotopy methods (see [Li03]), which seem particularly efficient for highly-structured polynomial systems, to the symbolic methods of this article.

Finally, it is clear that potential users should try all symbolic and numerical methods, perhaps one method on each on several computers, to see which are particularly effective for their class of problems. With this in mind, it seems important to have software packages that seamlessly connect symbolic, numerical, and symbolic-numeric methods. Currently, users wishing to use Bertini and Singular (or any other pairing of numerical and symbolic software packages) must create input files or scripts in significantly different syntaxes and also must become accustomed to the various forms of output produced by each. This is a significant obstacle for non-experts that can be overcome only by collaborations between experts.

References

[Alt23] H. Alt, Über die Erzeugung gegebener ebener Kurven mit Hilfe des Gelenkvierecks. *Zeitschrift für Angewandte Mathematik und Mechanik*, 3(1):13–19, 1923.

[BHMPS12] D.J. Bates, J.D. Hauenstein, T. McCoy, C. Peterson, and A.J. Sommese, Recovering exact results from inexact numerical data in algebraic geometry. To appear in *Experimental Mathematics*, 2012.

[BHPS09] D.J. Bates, J.D. Hauenstein, C. Peterson, and A.J. Sommese, A numerical local dimension test for points on the solution set of a system of polynomial equations. *SIAM J. on Numer. Anal.*, 47:3608–3623, 2009.

[BHS11] D.J. Bates, J.D. Hauenstein, and A.J. Sommese, Efficient path tracking methods. *Numerical Algorithms*, 58(4):451–459, 2011.

[BHSW06] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, *Bertini: Software for numerical algebraic geometry.* Available at `www.nd.edu/∼sommese/bertini`, 2006.

[BHSW08] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Adaptive multiprecision path tracking. *SIAM J. Numer. Anal.*, 46:722–746, 2008.

[BHSW09] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Stepsize control for adaptive multiprecision path tracking. *Contemporary Mathematics*, 496:21–31, 2009.

[BW96] E. Becker and T. Wörmann, Radical computations of zero-dimensional ideals and real root counting. *Math. Comput. Simul.*, 42:561–569, 1996.

[DZ05] B. Dayton and Z. Zeng, Computing the multiplicity structure in solving polynomial systems, In *Proceedings of ISSAC 2005*, pp. 116–123, ACM, New York, 2005.

[DGP99] W. Decker, G.-M. Greuel, and G. Pfister, Primary decomposition: algorithms and comparisons. In *Algorithmic algebra and number theory*, pp. 187–220, Springer, Berlin, 1999.

[DGPS10] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, SINGULAR *3-1-3 — A computer algebra system for polynomial computations.* Available at `www.singular.uni-kl.de`, 2011.

[DP12] W. Decker and G. Pfister, A First Course in Computational Algebraic Geometry. Cambridge University Press, 2012.

[EHV92] D. Eisenbud, C. Huneke, and W. Vasconcelos, Direct methods for primary decomposition. *Invent. Math.*, 110:207–235, 1992.

[GP08] G.-M. Greuel and G. Pfister, *A* SINGULAR *Introduction to Commutative Algebra.* Second edition, Springer, Berlin, 2008.

[GTZ88] P. Gianni, B. Trager, and G. Zacharias, Gröbner bases and primary decomposition of polynomial ideals. *J. Symb. Comput.*, 6:149–167, 1988.

[HM07] A. Herzberg and R. Murty, Sudoku squares and chromatic polynomials. *Notices of the AMS*, 54:708–717, 2007.

[HSW10] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Regeneration homotopies for solving systems of polynomials, *Math. Comput.*, 80:345–377, 2010.

[HSW11] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Regenerative cascade homotopies for solving polynomial systems, *Appl. Math. Comput.*, 218:1240–1246, 2011.

[HS12] J.D. Hauenstein and F. Sottile, Algorithm 921: alphaCertified: Certifying solutions to polynomial systems, *ACM Trans. Math. Software* 38(4):28, 2012.

[HS04] S. Hosten and S. Sullivant, Ideals of adjacent minors. *J. Algebra*, 277:615–642, 2004.

[Inn95] C. Innocenti, Polynomial solution to the position analysis of the 7-link Assur kinematic chain with one quaternary link. *Mech. Mach. Theory*, 30:1295–1303, 1995.

[KSS98] H. Kobayashi, H. Suzuki, and Y. Sakai, Numerical calculation of the multiplicity of a solution to algebraic equations. *Math. Comp.*, 67:257-270, 1998.

[KL91] T. Krick and A. Logar, An algorithm for the computation of the radical of an ideal in the ring of polynomials. *Applied algebra, algebraic algorithms and error-correcting codes*, Vol. 539 of Lecture Notes in Comput. Sci., pp. 195–205, Springer, Berlin, 1991.

[LVZ06] A. Leykin, J. Verschelde, and A. Zhao, Newton's method with deflation for isolated singularities of polynomial systems. *Theor. Comp. Sci.* 359:111–122, 2006.

[LVZ08] A. Leykin, J. Verschelde, and A. Zhao, Higher-order deflation for polynomial systems with isolated singular solutions. In *Algorithms in algebraic geometry*, Vol. 146 of IMA Vol. Math. Appl., pp. 79–97, Springer, New York, 2008.

[Li03] T.-Y. Li, Numerical solution of polynomial systems by homotopy continuation methods. *Handbook of numerical analysis, Vol. XI*, pp. 209–304, North-Holland, Amsterdam, 2003.

[LLT08] T.L. Lee, T.Y. Li, and C.H. Tsai, HOM4PS–2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83:109–133, 2008. Available at `hom4ps.math.msu.edu/HOM4PS_soft.htm`.

[Man73] N.I. Manolescu, A method based on Baranov Trusses, and using graph theory to find the set of planar jointed kinematic chains and mechanisms. *Mech. Mach. Theory*, 8:3–22, 1973.

[Mon02] C. Monico, Computing the primary decomposition of zero-dimensional ideals. *J. Symb. Comput.*, 34:451–459, 2002.

[Oji87] T. Ojika, Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations. *J. Math. Anal. Appl.* 123:199–221, 1987.

[OWM83] T. Ojika, S. Watanabe, and T. Mitsui, Deflation algorithm for the multiple roots of a system of nonlinear equations. *J. Math. Anal. Appl.* 96:463–479, 1983.

[Rit32] J.F. Ritt, *Differential equations from the algebraic standpoint.* Amer. Math. Soc. Colloq. Publ., Vol. 14, AMS, Providence, RI, 1932.

[Rit50] J.F. Ritt, *Differential algebra.* Amer. Math. Soc. Colloq. Publ., Vol. 33, AMS, Providence, RI, 1950.

[SY96] T. Shimoyama, and K. Yokoyama, Localization and primary decomposition of polynomial ideals. *J. Symb. Comput.*, 22:247–277, 1996.

[SV00] A.J. Sommese and J. Verschelde, Numerical homotopies to compute generic points on positive dimensional algebraic sets, *J. Complexity*, 16:572–602, 2000.

[SVW01] A.J. Sommese, J. Verschelde and C.W. Wampler, Using monodromy to decompose solution sets of polynomial systems into irreducible components. In *Applications of algebraic geometry to coding theory, physics and computation*, Vol. 36 of NATO Sci. Ser. II Math. Phys. Chem., pp. 297–315, Kluwer Acad. Publ., Dordrecht, 2001.

[SW95] A.J. Sommese and C.W. Wampler, Numerical algebraic geometry. In *The mathematics of numerical analysis*, Vol. 32 of Lectures in Appl. Math., pp. 749–763, AMS, Providence, RI, 1996.

[SW05] A.J. Sommese and C.W. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science.* World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[Stu02] B. Sturmfels, *Solving systems of polynomial equations.* CBMS Regional Conference Series in Mathematics, Vol. 97, 2002.

[SMSW06] H.-J. Su, J. M. McCarthy, M. Sosonkina, and L. T. Watson, Algorithm 857: POLSYS_GLP – a parallel general linear product homotopy code for solving polynomial systems of equations. *ACM Trans. on Math. Software*, 32:561–579, 2006.

[Ver99] J. Verschelde, Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. on Math. Software*, 25:251–276, 1999. Available at `www.math.uic.edu/∼jan`.

[Wan92] D.M. Wang, Irreducible decomposition of algebraic varieties via characteristic sets and Gröbner bases. *Comput. Aided Geom. Design*, 9:471–484, 1992.

[WMS92] C.W. Wampler, A.P. Morgan, and A.J. Sommese, Complete solution of the nine-point path synthesis problem for four-bar linkages. *ASME J. Mech. Design*, 114:153–159, 1992.

[WMS97] C.W. Wampler, A.P. Morgan, and A.J. Sommese, Complete solution of the nine-point path synthesis problem for four-bar linkages - Closure. *ASME J. Mech. Design*, 119:150–152, 1997.

[Wil59] J.H. Wilkinson, The evaluation of the zeros of ill-conditioned polynomials. *Numer. Math.*, 1:150–180, 1959.

[Wri85] A.H. Wright, Finding all solutions to a system of polynomial equations. *Math. Comp.*, 44:125–133, 1985.

[Wu84] W.J. Wu, Basic principles of mechanical theorem proving in elementary geometries. *J. Systems Sci. Math. Sci.*, 4:207–235, 1984.

[Wun63] W. Wunderlich, Höhere koppelkurven. *Ost. Ing. Arch.*, 17:162-165, 1963.