Homotopy Methods for Solving Polynomial Systems tutorial at ISSAC'05, Beijing, China, 24 July 2005

Jan Verschelde*

draft of March 15, 2007

Abstract

Homotopy continuation methods provide symbolic-numeric algorithms to solve polynomial systems. We apply Newton's method to follow solution paths defined by a family of systems, a so-called homotopy. The homotopy connects the system we want to solve with an easier to solve system. While path following algorithms are numerical algorithms, the creation of the homotopy can be viewed as a symbolic operation. The performance of the homotopy continuation solver is primarily determined by its ability to exploit structure of the system. The focus of this tutorial is on linking recent algorithms in numerical algebraic geometry to the software package PHCpack.

Contents

*Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, USA. *Email:* jan@math.uic.edu or jan.verschelde@na-net.ornl.gov. *URL:* http://www.math.uic.edu/~jan. This material is based upon work supported by the National Science Foundation under Grant No. 0134611 and Grant No. 0410036.

1 Introduction

The prerequisites for this tutorial is a basic introductory course in numerical analysis. We assume familiarity with concepts as numerical stability and conditioning. Newton's method in several variables is our point of departure.

The goal of the tutorial is to sketch the main algorithms in numerical algebraic geometry. For every algorithm, we outline its input-output specifications and illustrate it by means of an exercise. Every algorithm is justified by a theorem, which defines the conditions in which the algorithm is to be applied. Every algorithm is validated by software, we show how to make use of the algorithms in PHCpack.

The tutorial is divided in three lectures. In the first lecture, we concentrate on computing approximations to all isolated solutions of a polynomial system and assume that our input is a square polynomial systems. The "square" here means having as many equations as unknowns. In the second lecture we lift this restriction, allowing more or less equations than the number of unknowns, and study numerical homotopy methods to compute all positive dimensional solution sets of a polynomial systems, in particular we compute a numerical irreducible decomposition. The third lecture is a hands-on session. After a brief introduction to `phc`, we return to the exercises corresponding to the algorithms of the first two lectures.

The focus of these tutorial notes lies on linking the algorithms in numerical algebraic geometry to the software package PHCpack. For further reading and a more detailed description, the recent book [119] is strongly recommended.

Acknowledgments. I thank the organizers of ISSAC'05 for the opportunity of presenting this tutorial. I am grateful to Kathy Piret and Yan Zhuang for their comments and for working through the exercises. Thanks to Yun Guan for carefully reading through this draft. I am also grateful to Karl Schmedders for pointing out errors.

2 Lecture I: Root Counting and Homotopy Methods

The goal of the first lecture is to sketch the main algorithms to compute all isolated solutions of a polynomial system using numerical homotopy continuation methods. This lecture is divided in four parts.

Homotopy continuation methods are globally convergent and exhaustive solvers, i.e: they find *all* isolated solutions. We show that – using complex arithmetic – with probability one the solution paths defined by the homotopy avoid the discriminant variety, except perhaps at the very end of the paths if the system to be solved has singular solutions. In the second and third part of the lecture we emphasize the analogy between root counting and homotopy methods. Generalizations of Bézout's theorem lead to linear-product start systems. The sparse structure of a polynomial system is modeled by a tuple of Newton polytopes. Polyhedral homotopies are optimal for systems with generic coefficients and have yielded fully automatic blackbox solvers. The last part of this lecture sketches recent algorithms to compute isolated singularities accurately restoring the quadratic convergence of Newton's method by deflation. Once the accurate location of the isolated singular root is known, we can compute the generators for the dual space and the multiplicity of the root.

2.1 Deformation Methods avoid the Discriminant Variety

Definition 2.1 We consider a *polynomial system* $f(\mathbf{x}) = \mathbf{0}$ of N equations $f = (f_1, f_2, \ldots, f_N)$ in n variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ with complex coefficients, $f_i(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$, for $i = 1, 2, \ldots, N$. A polynomial $p \in \mathbb{C}[\mathbf{x}]$ is a finite sum of terms $c_{\mathbf{a}}\mathbf{x}^{\mathbf{a}}$. Each term is the product of a coefficient $c_{\mathbf{a}} \in \mathbb{C}$ and a monomial $\mathbf{x}^{\mathbf{a}} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$. The *Jacobian matrix* of the system $f(\mathbf{x}) = \mathbf{0}$ is the matrix of all first partial derivatives, denoted by $\frac{\partial f}{\partial \mathbf{x}} = \left[\frac{\partial f_i}{\partial x_j}\right]$, for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, n$.

Algorithm 2.1 Newton's method
Input: $f(\mathbf{x}) = \mathbf{0}$, a system of equations;
$\mathbf{x}^0$ is an initial guess for the solution.
Output: $\mathbf{x}^*$ is a more accurate approximation for the solution.
Implementation: `phc -v`.

Newton's method is also called the method of Newton-Raphson. When the system has more equations than unknowns, the method becomes known as the Gauss-Newton method, using least squares to solve the overdetermined linear systems.

Example 2.1 Consider the system (from [95]) and its Jacobian matrix:

$$f(x_1, x_2) = \begin{cases} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125x_2^2 - 1.5 = 0 \end{cases} \qquad \frac{\partial f}{\partial \mathbf{x}}(x_1, x_2) = \begin{bmatrix} 2x_1 & 1 \\ 1 & 0.25x_2 \end{bmatrix}. \tag{1}$$

One step in Newton's method starting at $\mathbf{x}^0 = (a, b)$ is computed by solving a linear system for the update vector $\Delta\mathbf{x}$, which is then added to $\mathbf{x}^0$ to obtain a new approximation $\mathbf{x}^1$:

$$\frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}^0)\Delta\mathbf{x} = -f(\mathbf{x}^0) \qquad \mathbf{x}^1 = \mathbf{x}^0 + \Delta\mathbf{x}. \tag{2}$$

For some norm $||\cdot||$, $||f(\mathbf{x}^0)||$ (also called the residual) measures the backward error, i.e.: how much one should change the coefficients of f to have $\mathbf{x}^0$ as the exact solution. The condition number C of the Jacobian matrix $\frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}^0)$ measures the forward error, i.e.: if the coefficients are given with m digits precision, then the error on $\mathbf{x}^0$ can be as large as $C10^{-m}$. ◇

Theorem 2.1 (quadratic convergence of Newton's method) *If the Jacobian matrix of f is regular at the root $\mathbf{x}^*$ and the initial guess $\mathbf{x}^0$ is sufficiently close to $\mathbf{x}^*$, then the sequence of approximations $\mathbf{x}^k$ produced by Newton's method converges quadratically to the root $\mathbf{x}^*$.*

Exercise 2.1 Consider again the system (1). The system has two solutions (counted with multiplicity): $(-3, -6)$ and $(1, 2)$. Which one of the two roots is regular? Starting at $(-2.999, -5.999)$ and $(0.999, 1.999)$ respectively, how many iterations does Newton's method take to get 32 decimal places correctly? ◇

Definition 2.2 To solve a *target system* $f(\mathbf{x}) = \mathbf{0}$, we construct a *start system* $g(\mathbf{x}) = \mathbf{0}$ and define the *artificial-parameter homotopy*

$$h(\mathbf{x}, t) = \gamma(1 - t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0}, \quad \gamma \in \mathbb{C}, \quad t \in [0, 1]. \tag{3}$$

The constant γ is a random constant number, also called the *accessibility constant*. The t is the *continuation parameter* and varies from 0 to 1, deforming the start system g into the target system f.

Algorithm 2.2 predictor-corrector method
Input: $h(\mathbf{x}, t) = \mathbf{0}$, a homotopy;
$\mathbf{x}^0$: $h(\mathbf{x}^0, 0) = \mathbf{0}$, a start solution.
Output: $\mathbf{x}^1$: $h(\mathbf{x}^1, 1) = \mathbf{0}$, target solution.
Implementation: `phc -p`.

Example 2.2 Consider again the system (1). Since the two equations are both quadrics, a natural choice for a homotopy is

$$h(\mathbf{x}, t) = \begin{pmatrix} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{pmatrix}(1 - t) + \begin{pmatrix} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125x_2^2 - 1.5 = 0 \end{pmatrix} t = \mathbf{0}. \tag{4}$$

At $t = 0$, we can easily compute the four solutions of the start system $h(\mathbf{x}, 0) = \mathbf{0}$. As we move t away from zero, we use Newton's method to extend the corresponding solutions of $h(\mathbf{x}, t) = \mathbf{0}$ until we reach $t = 1$. However, at $t \approx 0.92$, there are singular solutions and the path tracker will fail. This example is small enough so we compute all values for t for which the system has singular solutions, with the Maple 9.5 instructions below:

```
[> f := [x^2 + y - 3, x + 1/8*y^2 - 3/2];              # target system
[> g := [x^2 - 1, y^2 - 1];                            # start system
[> h := [g[1]*(1-t) + f[1]*t,g[2]*(1-t) + f[2]*t];     # homotopy
[> dh := Matrix(2,2,[[diff(h[1],x),diff(h[1],y)],
                     [diff(h[2],x),diff(h[2],y)]]);    # Jacobian matrix
[> d := LinearAlgebra[Determinant](dh);
[> gb := Groebner[gbasis]([h[1],h[2],d],plex(x,y,t));  # eliminate x,y
[> fsolve(gb[1],t,complex);                            # solve the equation in t
```

Since at $t = 0$, the start system has only regular solutions, after eliminating x and y we must end up with a nonzero polynomial in t, so there are only finitely many singularities along the solution paths defined by $h(\mathbf{x}, t) = \mathbf{0}$. The homotopy is not good if those singularities occur for t between 0 and 1. $\diamond$

The homotopy (4) is a bad homotopy because we took γ to be one, which is far from being a random complex number. The Maple calculations in Example 2.2 illustrate the main theorem of elimination theory, which is applied in the proof of the next theorem.

Theorem 2.2 (the gamma trick) *Denote the start system in the homotopy by $g(\mathbf{x}) = h(\mathbf{x}, 0) = \mathbf{0}$ and the target system by $f(\mathbf{x}) = h(\mathbf{x}, 1) = \mathbf{0}$. For almost all choices of a complex constant γ, all solution paths defined by the homotopy $h(\mathbf{x}, t) = \gamma(1-t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0}$ are regular, i.e.: for all $t \in [0, 1)$, the Jacobian matrix of $h(\mathbf{x}, t)$ is regular and no path diverges.*

Exercise 2.2 Consider again the system (1). Use `phc -b` to solve it. Look at the output file for `#iter` to see the number of iterations with Newton's method. You will see a difference between the number of iterations for paths converging to the solutions $(-3, -6)$ and $(1, 2)$. Can you relate these differences with your findings in Exercise 2.1? In the output file of `phc -b`, look which value for γ has been generated. As in Example 2.2, calculate the discriminant for the homotopy used by `phc -b`. $\diamond$

Definition 2.3 In many practical applications we consider a system as $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$. In addition to the n variables $\mathbf{x}$, the system $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ has m parameters $\mathbf{y}$, which are meaningful to the application. The system $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ then naturally defines a *natural-parameter homotopy*. In addition to the equations in f, often we encounter *side conditions*, which may not be explicitly, but implicitly defined by polynomial equations. The union of all algebraic sets defined by those side conditions is denoted by U.

Example 2.3 In [29], the authors study an application from molecular chemistry

$$f(x_1, x_2, x_3) = \begin{cases} 0.5(x_2^2 + 4x_2x_3 + x_3^2) + a(x_2^2x_3^2 - 1) = 0 \\ 0.5(x_3^2 + 4x_3x_1 + x_1^2) + a(x_3^2x_1^2 - 1) = 0 \\ 0.5(x_1^2 + 4x_1x_2 + x_2^2) + a(x_1^2x_2^2 - 1) = 0 \end{cases} \tag{5}$$

where a is a parameter. The system is also discussed on page 391 and following in [121]. For generic choices of the parameter a, there are 16 regular isolated solutions. $\diamond$

Algorithm 2.3 coefficient-parameter polynomial continuation
Input: $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, system with parameters $\mathbf{y}$;

$\mathbf{x}^0$: $f(\mathbf{x}^0, \mathbf{y}^0) = \mathbf{0}$, a start solution;
$\mathbf{y}^0$, target values for the parameters.
Output: $\mathbf{x}^1$: $f(\mathbf{x}^1, \mathbf{y}^1) = \mathbf{0}$, target solution.
Implementation: `phc -p`.

Theorem 2.3 (generic root count) *For almost all choices of the parameters* $\mathbf{y}$ *of the system* $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, *the number of isolated solutions in* $\mathbb{C}^n \setminus U$ *is the same constant, denote it by* $\mathcal{N}$. *For any choice of the parameters* $\mathbf{y}$, *the system* $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ *has at most* $\mathcal{N}$ *isolated solutions in* $\mathbb{C}^n \setminus U$.

Exercise 2.3 Consider again the system in (5). Pick a random complex number for the parameter a and verify with `phc -b` that there are 16 regular isolated solutions in general. Now choose a particular value for a, anything you like, and use the system you just solved as start system to compute the solutions for the particular instance of a. ◇

Bibliographic Notes. The second part of [11] gives a geometric perspective on the theory for finding zeros of polynomial systems, starting with Newton's method and covering condition numbers of homotopies. Path following methods are covered in [2, 3, 4, 5], see also [83] and [141, 142, 143]. The idea of coefficient-parameter polynomial continuation appeared in [88, 89], generalizing cheater homotopies [75], see also [78].

2.2 Product Structures generalize Bézout's Theorem

One can view the theorem of Bézout as the fundamental theorem of algebraic geometry. In this section we show how apply this theorem using multidegrees. Consider for example the eigenvalue problem $A\mathbf{x} = \lambda\mathbf{x}$. Instead of viewing this as a system of quadratic equations, one better separates the eigenvalue λ from the eigenvector $\mathbf{x}$ to exploit the linearity (degree one) in $\mathbf{x}$ and λ.

Definition 2.4 Consider a partition of the variables $\mathbf{x}$,

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = ((x_{1,1}, x_{1,2}, \ldots, x_{1,n_1}), (x_{2,1}, x_{2,2}, \ldots, x_{2,n_2}), \ldots, (x_{k,1}, x_{k,2}, \ldots, x_{k,n_k})). \quad (6)$$

For a polynomial p in $\mathbf{x}$, we define the *degree of* p *in* $\mathbf{x}_i$ (also called the *multidegree*) as

$$\deg(p, \mathbf{x}_i) = \deg(p(\mathbf{x}_1 = \mathbf{c}_1, \ldots, \mathbf{x}_{i-1} = \mathbf{c}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1} = \mathbf{c}_{i+1}, \ldots, \mathbf{x}_n = \mathbf{c}_n)), \quad (7)$$

for random choices of $\mathbf{c}_1 \in \mathbb{C}^{n_1}$, $\mathbf{c}_2 \in \mathbb{C}^{n_2}$, …, $\mathbf{c}_n \in \mathbb{C}^{n_k}$. For the system $f(\mathbf{x}) = \mathbf{0}$, $f = (f_1, f_2, \ldots, f_n)$, and partition $n = n_1 + n_2 + \cdots + n_k$, define the n-by-k *degree matrix* A as $A_{ij} = \deg(f_i, \mathbf{x}_j)$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, k$. The *permanent* $\mathrm{per}(A, (n_1, n_2, \ldots, n_k))$ is the sum of all possible products of n_1 elements of column 1 with n_2 elements of column 2, …, with n_k elements of column k, each time taking exactly one element from each row.

Algorithm 2.4 compute the permanent of a degree matrix
Input: a partition of n: $n = n_1 + n_2 + \cdots + n_k$, $n_i > 0$;
A is an n-by-k matrix.
Output: $\mathrm{per}(A, (n_1, n_2, \ldots, n_k))$ is the permanent of A.
Implementation: `phc -r`.

Theorem 2.4 (multihomogeneous Bézout bound) *Let* $n = n_1 + n_2 + \cdots + n_k$ *be a partition of* $\mathbf{x}$ *and* A *be the corresponding degree matrix of a system* $f(\mathbf{x}) = \mathbf{0}$. *The permanent* $\mathrm{per}(A, (n_1, n_2, \ldots, n_k))$ *is an upper bound for the number of isolated solutions of* $f(\mathbf{x}) = \mathbf{0}$.

Example 2.4 Polynomial systems occur frequently in mechanisms design. To determine the hand position and orientation of a PUMA robot (PUMA = Programmable Universal Machine for Assembly), one has to solve the following polynomial system (from [84]):

$$f(\mathbf{x}) = \begin{cases} x_1^2 + x_2^2 - 1 = 0 \\ x_3^2 + x_4^2 - 1 = 0 \\ x_5^2 + x_6^2 - 1 = 0 \\ x_7^2 + x_8^2 - 1 = 0 \\ 0.004731x_1x_3 - 0.3578x_2x_3 - 0.1238x_1 - 0.001637x_2 - 0.9338x_4 + x_7 - 0.3571 = 0 \\ 0.2238x_1x_3 + 0.7623x_2x_3 + 0.2638x_1 - 0.07745x_2 - 0.6734x_4 - 0.6022 = 0 \\ x_6x_8 + 0.3578x_1 + 0.004731x_2 = 0 \\ -0.7623x_1 + 0.2238x_2 + 0.3461 = 0 \end{cases} \tag{8}$$

The total degree (product of the degrees of the polynomials in the system) equals 128. Partitioning $\mathbf{x}$ into $((x_1x_2),(x_3x_4x_7x_8),(x_5x_6))$ leads to the degree matrix and permanent

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad \begin{aligned} \mathrm{per}(A,(2,4,2)) &= 2_{1,1} \times 1_{8,1} \\ &\quad \times 2_{2,2} \times 2_{4,2} \times 1_{5,2} \times 1_{6,2} \\ &\quad \times 2_{4,3} \times 1_{7,3} \\ &= 16. \end{aligned} \tag{9}$$

The double subscripts in the products above refer to the coordinates of the entries in the degree matrix. Notice that the product is the only possible nonzero choice of exactly one element of each row, choosing exactly 2, 4, and 2 elements of columns 1, 2, and 3 respectively. The permanent $\mathrm{per}(A,(2,4,2))$ is a 3-homogeneous Bézout number which drastically improves the total degree. For the choices of the coefficients in (8), there are exactly 16 solutions. ◇

Exercise 2.4 Use `phc -r` to verify the calculation done in Example 2.4. Finding a good partition is very hard, fortunately, even for $n = 8$, enumerating all partitions of $\mathbf{x}$ goes fast. Let `phc -r` enumerate all partitions of $\mathbf{x}$ in this case and check how fast this goes. Also try the heuristic methods for finding a partition. ◇

Definition 2.5 The system $g(\mathbf{x}) = \mathbf{0}$, $g = (g_1, g_2, \ldots, g_n)$, is a *linear-product start system* if for all $i = 1, 2, \ldots, n$: $g_i(\mathbf{x})$ is the product of linear equations with randomly chosen coefficients.

Algorithm 2.5 solve a linear-product start system

Input: $g(\mathbf{x}) = \mathbf{0}$, g_i is a product of linear equations.
Output: all solutions of $g(\mathbf{x}) = \mathbf{0}$.
Implementation: `phc -r`.

Example 2.5 For the system of Example 2.4, the random linear-product start system corresponding to

the degree matrix is $g(\mathbf{x}) =$

$$\begin{cases} \left(c_{10}^{(1)} + c_{11}^{(1)}x_1 + c_{12}^{(1)}x_2\right)\left(c_{10}^{(2)} + c_{11}^{(2)}x_1 + c_{12}^{(2)}x_2\right) = 0 \\ \left(c_{20}^{(1)} + c_{21}^{(1)}x_3 + c_{22}^{(1)}x_4 + c_{23}^{(1)}x_7 + c_{24}^{(1)}x_8\right)\left(c_{20}^{(2)} + c_{21}^{(2)}x_3 + c_{22}^{(2)}x_4 + c_{23}^{(2)}x_7 + c_{24}^{(2)}x_8\right) = 0 \\ \left(c_{30}^{(1)} + c_{31}^{(1)}x_5 + c_{32}^{(1)}x_6\right)\left(c_{30}^{(2)} + c_{31}^{(2)}x_5 + c_{32}^{(2)}x_6\right) = 0 \\ \left(c_{40}^{(1)} + c_{41}^{(1)}x_3 + c_{42}^{(1)}x_4 + c_{43}^{(1)}x_7 + c_{44}^{(1)}x_8\right)\left(c_{40}^{(2)} + c_{41}^{(2)}x_3 + c_{42}^{(2)}x_4 + c_{43}^{(2)}x_7 + c_{44}^{(2)}x_8\right) = 0 \\ \left(c_{50}^{(1)} + c_{51}^{(1)}x_1 + c_{52}^{(1)}x_2\right)\left(c_{50}^{(2)} + c_{51}^{(2)}x_3 + c_{52}^{(2)}x_4 + c_{53}^{(2)}x_5 + c_{54}^{(2)}x_6\right)\left(c_{50}^{(3)} + c_{51}^{(3)}x_7 + c_{52}^{(3)}x_8\right) = 0 \\ \left(c_{60}^{(1)} + c_{61}^{(1)}x_1 + c_{62}^{(1)}x_2\right)\left(c_{60}^{(2)} + c_{61}^{(2)}x_3 + c_{62}^{(2)}x_4 + c_{63}^{(2)}x_5 + c_{64}^{(2)}x_6\right) = 0 \\ \left(c_{70}^{(1)} + c_{71}^{(1)}x_1 + c_{72}^{(1)}x_2\right)\left(c_{70}^{(2)} + c_{71}^{(2)}x_3 + c_{72}^{(2)}x_4 + c_{73}^{(2)}x_5 + c_{74}^{(2)}x_6\right)\left(c_{70}^{(3)} + c_{71}^{(3)}x_7 + c_{72}^{(3)}x_8\right) = 0 \\ c_{80}^{(1)} + c_{81}^{(1)}x_1 + c_{82}^{(1)}x_2 = 0 \end{cases} \tag{10}$$

The coefficients of the equations are complex numbers, chosen at random. Observe that the algorithm to solve $g(\mathbf{x}) = \mathbf{0}$ makes the same moves as the algorithm to compute the permanent of the degree matrix. $\diamond$

Theorem 2.5 (linear-product root count) *If all coefficients of $g(\mathbf{x}) = \mathbf{0}$ are chosen at random, then all solutions of $g(\mathbf{x}) = \mathbf{0}$ are isolated and regular. Moreover, the number of isolated solutions of $g(\mathbf{x}) = \mathbf{0}$ bounds the number of isolated solutions of any system $f(\mathbf{x}) = \mathbf{0}$ with the same degree structure as g.*

Exercise 2.5 Use `phc -r` to construct a linear-product start system for the system (8) of Example 2.4. Verify that the constructed start system (10 has indeed 16 regular isolated solutions and use `phc -p` to solve the system (8). Alternatively, and less tediously, one could solve the system with the black-box option `phc -b` and study the output file. $\diamond$

Definition 2.6 For a system $f(\mathbf{x}) = \mathbf{0}$, a *linear-product structure* is an array S of n arrays, $S = (S_1, S_2, \ldots, S_n)$. The jth array of S_i is a list of variables which will occur with nonzero coefficient in the corresponding linear-product start system $g(\mathbf{x}) = \mathbf{0}$, $g = (g_1, g_2, \ldots, g_n)$. Formally, we have

$$g_i(\mathbf{x}) = \prod_{j=1}^{\deg(g_i)} \left(c_{i,j,0} + \sum_{x \in S_{ij}} c_{i,j,k}x \right), \quad c_{i,j,k} \in \mathbb{C}, \tag{11}$$

where the coefficients $c_{i,j,k}$ are chosen at random. We say that S is a *supporting* linear-product structure for the system $f(\mathbf{x}) = \mathbf{0}$ if every monomial in the system f also occurs in the system g.

Example 2.6 A neural network modeled by an adaptive Lotka-Volterra system gives rise to the following system:

$$f(x_1, x_2, x_3) = \begin{cases} x_1x_2^2 + x_1x_3^2 - \lambda x_1 + 1 = 0 & (x_1)\ (x_2\ \ x_3)\ (x_2\ \ x_3) \\ x_2x_1^2 + x_2x_3^2 - \lambda x_2 + 1 = 0 & (x_2)\ (x_1\ \ x_3)\ (x_1\ \ x_3) \\ x_3x_1^2 + x_3x_2^2 - \lambda x_3 + 1 = 0 & (x_3)\ (x_1\ \ x_2)\ (x_1\ \ x_2) \end{cases} \tag{12}$$

where for the parameter λ we can pick any value, say 3.14. The structure at the right of the system generalizes multi-homogenization, and leads to the generalized Bézout number 21. While 21 may not be

such a significant savings compared to 27, the main advantage is that one can restrict the choice of the random coefficients of the linear-product start system so it shares the same full permutation symmetry as the target system. In particular, any permutation of the variables leaves the solution set invariant. As start and target system share the same symmetry, so do all paths defined by the homotopy. Solutions which can be transformed into each other by permuting the variables belong to the same orbit. It suffices to compute only one generator per orbit. For this system it suffices to trace only the 8 generating solution paths. $\diamond$

Algorithm 2.6 extract a linear-product structure
Input: $f(\mathbf{x}) = \mathbf{0}$, a polynomial system.
Output: $g(\mathbf{x}) = \mathbf{0}$, a linear-product start system.
Implementation: `phc -r`.

Theorem 2.6 (linear-product start system) *For a system $f(\mathbf{x}) = \mathbf{0}$ and random linear-product start system $g(\mathbf{x}) = \mathbf{0}$ with the same degree structure as f, consider homotopy $h(\mathbf{x}, t) = \gamma(1-t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0}$, with the constant γ chosen at random. All isolated solutions of $f(\mathbf{x}) = \mathbf{0}$ lie at the end of some solution path defined by $h(\mathbf{x}, t) = \mathbf{0}$, starting at the solutions of $g(\mathbf{x}) = \mathbf{0}$. In particular, a solution of $f(\mathbf{x}) = \mathbf{0}$ of multiplicity m is reached by exactly m solution paths.*

Exercise 2.6 Verify by hand and/or with the help of `phc -r` that the linear-product start system respecting the same full permutation symmetry as the system (12) has indeed eight generating solutions. $\diamond$

Bibliographic Notes. A nice introduction to solving polynomial systems by homotopies can be found in [69]. The earliest applications of homotopies for solving polynomial systems ([16], [26], [40], [39], [68], [72] [87], [145], [147]) belong to the dense class, where the number of paths equals the product of the degrees in the system. Multihomogeneous homotopies were introduced in [86, 85] and applied in [139, 140], see also [138]. Similar are the random product homotopies [73, 74], see also [77]. General linear-product start systems started to appear in [135], and were extended in [131, 132], [76], and [144]. A general approach to exploit product structures was developed in [93].

2.3 Newton Polytopes model Sparse Structures

A system is sparse when only few monomials occur with nonzero coefficients. An interesting class of sparse systems have exactly two terms in every equation, we call such systems "binomial". The Cayley trick implements Minkowski's theorem which defines mixed volumes. Via polyhedral homotopies we provide a constructive proof for Bernshtein's theorem. We end this section listing the conditions under which the mixed volume fails to give a sharp root count.

Definition 2.7 Denote $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$. An matrix A with integer coefficients and a vector $\mathbf{c} \in (\mathbb{C}^*)^n$ define a *binomial system*, denoted as $\mathbf{x}^A = \mathbf{c}$. The columns of A define the exponent vectors of the equations in the binomial system, i.e.: for $A = [\mathbf{a}_1 \mathbf{a}_2 \cdots \mathbf{a}_n]$, we have

$$\mathbf{x}^A = \mathbf{x}^{[\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_n]} = [\mathbf{x}^{\mathbf{a}_1} \ \mathbf{x}^{\mathbf{a}_2} \ \cdots \ \mathbf{x}^{\mathbf{a}_n}] \quad \text{and} \quad \mathbf{x}^A = \mathbf{c} \Leftrightarrow [\mathbf{x}^{\mathbf{a}_1} = c_1 \ \mathbf{x}^{\mathbf{a}_2} = c_2 \ \cdots \ \mathbf{x}^{\mathbf{a}_n} = c_n]. \qquad (13)$$

Algorithm 2.7 solve a binomial system
Input: A and $\mathbf{c}$ define $\mathbf{x}^A = \mathbf{c}$, a binomial system.
Output: all solutions in $(\mathbb{C}^*)^n$ to $\mathbf{x}^A = \mathbf{c}$.
Implementation: `phc -b`.

Example 2.7 Consider the following binomial system:

$$\left\{ \begin{array}{l} x_1^3 x_2^2 = 1 \\ x_1^5 x_2^1 = 1 \end{array} \right. \quad \Leftrightarrow \quad \mathbf{x}^A = \mathbf{c} : [x_1 \;\; x_2] \left[\begin{array}{cc} 3 & 5 \\ 2 & 1 \end{array} \right] = [1 \;\; 1]. \tag{14}$$

To make A upper triangular, we define a unimodular matrix M taking the greatest common divisor of 3 and 2: $\gcd(3,2) = 1 = (+1) \cdot 3 + (-1) \cdot 2$. To eliminate the 2 in A, the second row in M will contain the coefficients of the linear combination of 3 and 2: $(+2) \cdot 3 + (-3) \cdot 2 = 0$. So we have

$$M = \left[\begin{array}{cc} 1 & -1 \\ -2 & 3 \end{array} \right], \det(M) = 1 \quad MA = \left[\begin{array}{cc} 1 & -1 \\ -2 & 3 \end{array} \right] \left[\begin{array}{cc} 3 & 5 \\ 2 & 1 \end{array} \right] = \left[\begin{array}{cc} 1 & 4 \\ 0 & -7 \end{array} \right] = U. \tag{15}$$

The unimodular transformation M defines a change of coordinates:

$$\mathbf{y}^M = \mathbf{x} \Rightarrow \mathbf{x}^A = \mathbf{y}^{MA} = \mathbf{y}^U. \tag{16}$$

More explicitly, the coordinate transformation is

$$[x_1 \;\; x_2] = [y_1 \;\; y_2] \left[\begin{array}{cc} 1 & -1 \\ -2 & 3 \end{array} \right] = [x_1 = y_1 y_2^{-2} \;\; x_2 = y_1^{-1} y_2^3]. \tag{17}$$

When doing the coordinate transformation on the system, we recognize the matrix multiplication MA:

$$\left\{ \begin{array}{l} x_1^3 x_2^2 = \left(y_1 y_2^{-2}\right)^3 \left(y_1^{-1} y_2^3\right)^2 = y_1^{1 \cdot 3 + (-2) \cdot 3} y_2^{(-2) \cdot 3 + 3 \cdot 2} = y_1 \\ x_1^5 x_2^1 = \left(y_1 y_2^{-2}\right)^5 \left(y_1^{-1} y_2^3\right)^1 = y_1^{1 \cdot 5 + (-1) \cdot 1} y_2^{-2 \cdot 5 + 3 \cdot 1} = y_1^4 y_2^{-7}. \end{array} \right. \tag{18}$$

The change of coordinates has not altered the number of solutions, which equals

$$|\det(U)| = |\det(MA)| = |\det(U)\det(A)| = |\det(A)| = 7. \tag{19}$$

The upper triangular matrix U is the Hermite normal form of the matrix A. The unimodular transformations M are the discrete analogues to Givens rotations. Making the exponent matrix upper triangular defines coordinate transformations to make the binomial system triangular. ◇

Theorem 2.7 (regularity of a binomial system) *If* $\det(A) \neq 0$ *and* $\mathbf{c} \neq \mathbf{0}$*, then* $\mathbf{x}^A = \mathbf{c}$ *has exactly as many as* $|\det(A)|$ *isolated regular solutions.*

Exercise 2.7 Consider the binomial system $\mathbf{x}^A = \mathbf{c}$ with

$$A = \left[\begin{array}{ccc} 2 & 1 & 3 \\ 3 & 2 & 1 \\ 1 & 3 & 2 \end{array} \right] \quad \text{and} \quad \mathbf{c} = [1 \;\; 1 \;\; 1]. \tag{20}$$

Solve this system with `phc -b`. How many solutions does this system have? Compare this number with the lowest generalized Bézout bound, using `phc -r`. ◇

Definition 2.8 Consider the polynomial system $f(\mathbf{x}) = \mathbf{0}$, $f = (f_1, f_2, \ldots, f_n)$. For $i = 1, 2, \ldots, n$, the *support* A_i of f_i collects the exponents of $\mathbf{x}$ of those monomials which occur in f_i with a nonzero coefficient, so we can write $f_i(\mathbf{x}) = \sum_{\mathbf{a} \in A_i} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}$. The *Newton polytope* P_i of f_i is the convex hull of the support A_i of f_i. We denote $P_i = \text{conv}(A)$. We model the sparsity of the system $f(\mathbf{x}) = \mathbf{0}$ by its tuple of supports

$\mathcal{A} = (A_1, A_2, \ldots, A_n)$ and its tuple of Newton polytopes $\mathcal{P} = (P_1, P_2, \ldots, P_n)$. Denote by $\mathbf{e}_i$ the ith unit vector. The *Cayley embedding* $\kappa(\mathcal{A})$ of a tuple $\mathcal{A}$ adds $n-1$ coordinates to every point in $\mathcal{A}$ as follows:

$$\kappa(\mathcal{A}) = \{ \ (\mathbf{a}, \mathbf{0}) \mid \mathbf{a} \in A_1 \ \} \cup \bigcup_{i=2}^{n} \{ \ (\mathbf{a}, \mathbf{e}_{i-1}) \mid \mathbf{a} \in A_i \ \} . \tag{21}$$

The *Cayley polytope* is the convex hull of $\kappa(\mathcal{A})$.

Algorithm 2.8 the Cayley embedding
Input: $\mathcal{A} = (A_1, A_2, \ldots, A_r)$, a tuple of supports.
Output: $\kappa(\mathcal{A})$ is an $(r-1)$-simplex with vertices A_i.
Implementation: `phc -m`.

Theorem 2.8 (Minkowski's theorem) *Given an n-tuple of polytopes $(P_1, P_2, \ldots, P_n)$, the volume of a general linear combination of these polytopes, denoted by $V(\lambda_1 P_1 + \lambda P_2 + \cdots + \lambda_n P_n)$, is a homogeneous polynomial of degree n in the coefficients λ_1, λ_2, ..., λ_n. The coefficient of $\lambda_1 \lambda_2 \cdots \lambda_n$ in this polynomial is the mixed volume $V(P_1, P_2, \ldots, P_n)$ of the n-tuple.*

Example 2.8 Consider the following system:

$$\begin{array}{ll} f = (f_1, f_2) & \mathcal{A} = (A_1, A_2) \\ = \left\{ \begin{array}{l} x_1^3 x_2 + x_1 x_2^2 + 1 = 0 \\ x_1^4 + x_1 x_2 + 1 = 0 \end{array} \right. & \begin{array}{l} A_1 = \{(3,1), (1,2), (0,0)\} \\ A_2 = \{(4,0), (1,1), (0,0)\} \end{array} \end{array} \tag{22}$$

The Newton polytopes of the system are spanned by the tuple $\mathcal{A} = (A_1, A_2)$, where A_1 and A_2 are the supports of f_1 and f_2 respectively. A cross section of a subdivision of the Cayley polytope is shown in in Figure 1. In this cross section we see a mixed subdivision, for the convex combination $\lambda_1 P_1 + \lambda_2 P_2$, $\lambda_1 + \lambda_2 = 1$, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, where P_1 defines the base and P_2 is at the top of the polytope. The areas of the triangles in the cross section are $\lambda_1^2 \times \text{area}(P_1)$ and $\lambda_2^2 \times \text{area}(P_2)$, as each side of the triangle is scaled by λ_1 and λ_2 respectively. The area of the cell in the subdivision spanned by one edge of P_1 (scaled by λ_1) and the other edge of P_2 (scaled by λ_2) is scaled by $\lambda_1 \times \lambda_2$, as we move the cross section.

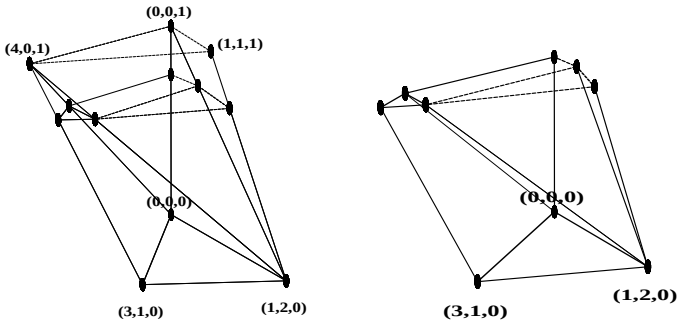


Figure 1: A mixed subdivision induced by a triangulation of the Cayley polytope.

In Figure 2 we show the Minkowski sum of the two polygons P_1 and P_2, with their mixed subdivision corresponding to the triangulation of the Cayley polytope. For this example, Minkowski's theorem becomes

$$\begin{array}{rcl} \text{area}(\lambda_1 P_1 + \lambda_2 P_2) & = & V(P_1, P_1)\lambda_1^2 + V(P_1, P_2)\lambda_1 \lambda_2 + V(P_2, P_2)\lambda_2^2 \\ & = & 5\lambda_1^2 + 8\lambda_1 \lambda_2 + 4\lambda_2^2. \end{array} \tag{23}$$

The coefficients in the polynomial (23) are mixed volumes (or areas in our example): $V(P_1, P_1)$ and $V(P_2, P_2)$ are the respective areas of P_1 and P_2, while $V(P_1, P_2)$ is the mixed area. Note that the area is scaled so that the unit triangle has area one. $\diamond$

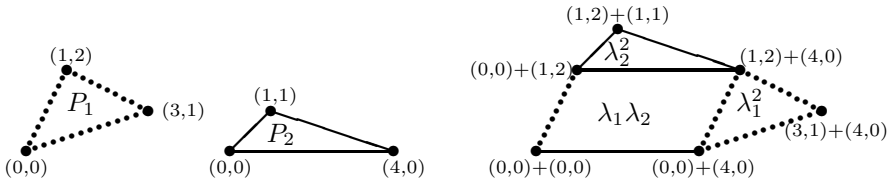


Figure 2: A subdivision of the sum of two polygons P_1 and P_2. The sum is the convex hull of all sums of the vertices of the polygons. The cells in the subdivision are labeled by the multipliers for the area of $\lambda_1 P_1 + \lambda_2 P_2$.

Exercise 2.8 Consider the system (1). How many different mixed subdivisions of the supports for this system can you find? ◇

Definition 2.9 For a finite point set A in n-space, we define the *lifting function* ω which assigns to every point $\mathbf{a} \in A$ a weight $\omega(\mathbf{a})$. We denote the lifted point set by $\widehat{A} = \omega(A)$ and accordingly its convex hull by $\widehat{P} = \mathrm{conv}(\widehat{A})$. A triangulation $\triangle$ of A is *regular* if for every cell $C \in \triangle$: $\widehat{C}$ is a facet of the lower hull of $\widehat{A}$. Then every cell C has a unique *inner normal* $\mathbf{v}$, $v_{n+1} = 1$. This inner normal $\mathbf{v}$ is the solution to a system of linear equalities and inequalities:

$$\begin{cases} \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle = \langle \widehat{\mathbf{b}}, \mathbf{v} \rangle, \forall \widehat{\mathbf{a}}, \widehat{\mathbf{b}} \in \widehat{C} \\ \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle = \langle \widehat{\mathbf{b}}, \mathbf{v} \rangle, \forall \widehat{\mathbf{a}} \in \widehat{C}, \forall \widehat{\mathbf{b}} \in \widehat{A} \setminus \widehat{C} \end{cases} \quad \text{or} \quad \forall \widehat{\mathbf{a}} \in \widehat{C} : \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle = \min_{\widehat{\mathbf{b}} \in \widehat{A}} \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle. \tag{24}$$

The cell $\widehat{C} = \partial_{\mathbf{v}} \widehat{A}$ spans the simplex $\mathrm{conv}(\widehat{C}) = \partial_{\mathbf{v}} \widehat{P}$. The *volume* $V(P)$ of a polytope is computed as

$$V(P) = \sum_{C \in \triangle} V(C). \tag{25}$$

For a point set A, $V(A)$ denotes $V(\mathrm{conv}(A))$.

Algorithm 2.9 dynamic lifting
Input: A a finite set of points in n-space.
Output: $\widehat{A}$ a lifted set of points;
$\triangle$ a regular triangulation of A;
$V(A)$ the volume of $\mathrm{conv}(A)$.
Implementation: `phc -m`.

Example 2.9 The dynamic lifting algorithm considers incrementally the points in a support with respect to a regular triangulation of the previously processed points. Points inside a simplex are lifted out, i.e.: we can always find a heigh enough lifting value so it does not belong to the lower hull. Points not in any simplex of the current triangulation are lifted sufficiently heigh as to preserve all simplices already computed. The key operation is to compute a barycentric decomposition of a point with respect to a simplex. We illustrate this by the simplex spanned by vertices $\mathbf{c}_0 = (0,0)$, $\mathbf{c}_1 = (3,2)$, and $\mathbf{c}_2 = (2,4)$. Table 1 shows the three different situations which may occur. ◇

Theorem 2.9 (cost of placeable triangulations) *The construction of a placeable triangulation $\triangle$ of a point set A in n-space takes at least $O((\#\triangle)n^3 + (\#A)n^2)$, at most $O((\#\triangle)n^3 + (\#A)(\#\triangle)n^2)$, and on average $O((\#\triangle)n^3 + (\#A)(\#\triangle)n)$ arithmetical operations.*

Exercise 2.9 Consider again the system (22) introduced in Example 2.8. Use `phc -m`, choose for dynamic lifting, and compute the Minkowski polynomial for this example. ◇

point	barycentric decomposition											pivoting
$\mathbf{x} = (2,3)$:	$\mathbf{x}$	$=$	$+$	$\frac{1}{8}$	$\mathbf{c}_0$	$+$	$\frac{1}{4}$	$\mathbf{c}_1$	$+$	$\frac{5}{8}$	$\mathbf{c}_2$	no new simplex
$\mathbf{y} = (5,1)$:	$\mathbf{y}$	$=$	$-$	$\frac{1}{3}$	$\mathbf{c}_0$	$+$	$\frac{9}{4}$	$\mathbf{c}_1$	$-$	$\frac{7}{8}$	$\mathbf{c}_2$	$[\mathbf{y}, \mathbf{c}_1, \mathbf{c}_2][\mathbf{c}_0, \mathbf{c}_1, \mathbf{y}]$
$\mathbf{z} = (1,5)$:	$\mathbf{z}$	$=$	$+$	$\frac{1}{8}$	$\mathbf{c}_0$	$-$	$\frac{3}{4}$	$\mathbf{c}_1$	$+$	$\frac{13}{8}$	$\mathbf{c}_2$	$[\mathbf{c}_0, \mathbf{z}, \mathbf{c}_2]$

Table 1: Three possible updates of the simplex $[\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2]$ with one point, $\mathbf{x}$, $\mathbf{y}$, or $\mathbf{z}$. Either we have no, two, or one new simplex by interchanging the vertex with negative coefficient with the point.

Definition 2.10 For a tuple of supports $\mathcal{A} = (A_1, A_2, \ldots, A_n)$ in n-space, a *lifting function* ω defines *lifted supports* $\widehat{\mathcal{A}} = (\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_n)$. A *regular mixed-cell configuration* $\mathcal{C}$ consists of all cells $C \in \mathcal{C}$ for which there is a unique inner normal $\mathbf{v}$, $v_{n+1} = 1$, satisfying the following systems of linear equalities and inequalities:

$$\begin{cases} \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle = \langle \widehat{\mathbf{b}}, \mathbf{v} \rangle, \forall \widehat{\mathbf{a}}, \widehat{\mathbf{b}} \in \widehat{C}_i \\ \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle = \langle \widehat{\mathbf{b}}, \mathbf{v} \rangle, \forall \widehat{\mathbf{a}} \in \widehat{C}_i, \forall \widehat{\mathbf{b}} \in \widehat{A}_i \setminus \widehat{C}_i \end{cases} \quad \text{or} \quad \forall \widehat{\mathbf{a}} \in \widehat{C}_i : \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle = \min_{\widehat{\mathbf{b}} \in \widehat{A}_i} \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle, \quad i = 1, 2, \ldots, n. \tag{26}$$

Then *the mixed volume* $V(\mathcal{P})$ of the corresponding tuple of Newton polytopes $\mathcal{P}$ is

$$V(\mathcal{P}) = \sum_{C \in \mathcal{C}} V(C). \tag{27}$$

For a tuple of supports $\mathcal{A}$ spanning the Newton polytopes $\mathcal{P}$, $V(\mathcal{A})$ denotes $V(\mathcal{P})$.

Algorithm 2.10 lift and prune
Input: $\mathcal{A}$ a tuple of supports of $f(\mathbf{x}) = \mathbf{0}$.
Output: $\widehat{\mathcal{A}}$, lifted supports;
$\mathcal{C}$ a regular mixed-cell configuration;
$V(\mathcal{A})$, the mixed volume of $\mathcal{A}$.
Implementation: `phc -m`.

Example 2.10 Take the supports $\mathcal{A}$ from the system (22) in Example 2.8 and assign an integer lifting to the points in $\mathcal{A}$ to obtain $\widehat{\mathcal{A}}$:

$$\widehat{\mathcal{A}} = (\widehat{A}_1, \widehat{A}_2), \quad \widehat{A}_1 = \{(3,1,1), (1,2,1), (0,0,0)\}, \widehat{A}_2 = \{(4,0,0), (1,1,1), (0,0,1)\}. \tag{28}$$

As the Newton polytopes each have three edges, there are 9 edge-edge combinations to try. An edge-edge combination leads to a mixed cell if there exists an inner normal $\mathbf{v} \neq \mathbf{0}$ which satisfies the conditions in (26). Two of the nine combinations lead to mixed cells:

$$\begin{array}{ll} \widehat{\mathcal{C}}^{(1)} = (\widehat{C}_1, \widehat{C}_2) & \widehat{\mathcal{C}}^{(2)} = (\widehat{C}_1, \widehat{C}_2) \\ \quad \widehat{C}_1 = \{(1,2,1), (0,0,0)\} & \quad \widehat{C}_1 = \{(1,2,1), (0,0,0)\} \\ \quad \widehat{C}_2 = \{(4,0,0), (1,1,1)\} & \quad \widehat{C}_2 = \{(1,1,1), (0,0,1)\} \\ \mathbf{v} = (1,-4,7), V(\mathcal{C}^{(1)}) = 7 & \mathbf{v} = (1,-1,1), V(\mathcal{C}^{(2)}) = 1 \end{array} \tag{29}$$

Adding the volumes of mixed cells, we arrive at eight, which is the mixed volume of the tuple of Newton polytopes spanned by the supports. The cells $\widehat{\mathcal{C}}^{(1)}$ and $\widehat{\mathcal{C}}^{(2)}$ constitute a regular mixed-cell configuration for $\mathcal{A}$. Observe that both cells share the same edge and belong to the same branch of the tree of all edge-edge combinations. Early pruning in this tree using linear programming on the conditions (26) is very important for an efficient mixed-volume computation. $\diamond$

Theorem 2.10 (mixed-cell configuration) *For any random lifting function ω on the supports $\mathcal{A}$, all cells C in the mixed-cell configuration $\mathcal{C}$ induced by ω are fine mixed, i.e., for $C = (C_1, C_2, \ldots, C_n)$: $\#C_i = 2$, $i = 1, 2, \ldots, n$. Moreover, the mixed volume is invariant under lifting.*

Exercise 2.10 Consider the cyclic 7-roots problem (see [7] and [10]):

$$f(\mathbf{z}) = \begin{cases} z_0 + z_1 + z_2 + z_3 + z_4 + z_5 + z_6 = 0 \\ z_0z_1 + z_1z_2 + z_2z_3 + z_3z_4 + z_4z_5 + z_5z_6 + z_6z_0 = 0 \\ z_0z_1z_2 + z_1z_2z_3 + z_2z_3z_4 + z_3z_4z_5 + z_4z_5z_6 + z_5z_6z_0 + z_6z_0z_1 = 0 \\ z_0z_1z_2z_3 + z_1z_2z_3z_4 + z_2z_3z_4z_5 + z_3z_4z_5z_6 + z_4z_5z_6z_0 + z_5z_6z_0z_1 + z_6z_0z_1z_2 = 0 \\ z_0z_1z_2z_3z_4 + z_1z_2z_3z_4z_5 + z_2z_3z_4z_5z_6 + z_3z_4z_5z_6z_0 + z_4z_5z_6z_0z_1 \\ \qquad + z_5z_6z_0z_1z_2 + z_6z_0z_1z_2z_3 = 0 \\ z_0z_1z_2z_3z_4z_5 + z_1z_2z_3z_4z_5z_6 + z_2z_3z_4z_5z_6z_0 + z_3z_4z_5z_6z_0z_1 + z_4z_5z_6z_0z_1z_2 \\ \qquad + z_5z_6z_0z_1z_2z_3 + z_6z_0z_1z_2z_3z_4 = 0 \\ z_0z_1z_2z_3z_4z_5z_6 - 1 = 0 \end{cases} \tag{30}$$

Use `phc -m` to compute a regular mixed-cell configuration. Compare the performance of dynamic lifting against a static lifting, with random lifting values. ◇

Definition 2.11 Let $\mathcal{A} = (A_1, A_2, \ldots, A_n)$ be a typle of supports. The ith polynomial $g_i(\mathbf{x})$ of a *random coefficient start system* $g(\mathbf{x}) = \mathbf{0}$ has support A_i and random complex coefficients. A random coefficient start system combined with a lifting function ω on the supports $\mathcal{A}$ defines *a polyhedral homotopy* $\widehat{g}(\mathbf{x}, t) = \mathbf{0}$, as follows:

$$\text{if} \quad g_i(\mathbf{x}) = \sum_{\mathbf{a} \in A_i} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} \quad \text{then} \quad \widehat{g}_i(\mathbf{x}, t) = \sum_{\mathbf{a} \in A_i} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} t^{\omega(\mathbf{a})}. \tag{31}$$

The lifting function ω induces a regular mixed-cell configuration $\mathcal{C}$, whose cells are characterized by inner normals $\mathbf{v}$: $\widehat{C} = \partial_{\mathbf{v}} \widehat{\mathcal{A}}$. The cells define the start systems $\partial_v g(\mathbf{x}) = \mathbf{0}$ in the polyhedral homotopy:

$$\partial_{\mathbf{v}} g_i(\mathbf{x}) = \sum_{\mathbf{a} \in C_i} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}, \quad C = (C_1, C_2, \ldots, C_n) \in \mathcal{C}. \tag{32}$$

The inner normals define a *polyhedral coordinate change* necessary to start the deformation process:

$$\text{for } i = 1, 2, \ldots, n: \; x_i = y_i s^{v_i}, \quad t = s. \tag{33}$$

After executing the coordinate change on the polyhedral homotopy we obtain the homotopy

$$\widehat{g}_i(\mathbf{y}, s) = \sum_{\mathbf{a} \in A_i} c_{\mathbf{a}} \mathbf{y}^{\mathbf{a}} s^{\langle (\mathbf{a}, \omega(\mathbf{a})), \mathbf{v} \rangle} = \sum_{\mathbf{a} \in C_i} c_{\mathbf{a}} \mathbf{y}^{\mathbf{a}} s^m + \sum_{\mathbf{a} \in A_i \setminus C_i} c_{\mathbf{a}} \mathbf{y}^{\mathbf{a}} s^{\langle (\mathbf{a}, \omega(\mathbf{a})), \mathbf{v} \rangle} = \mathbf{0}, \quad m = \min_{\mathbf{a} \in A_i} \langle \widehat{\mathbf{a}}, \mathbf{v} \rangle. \tag{34}$$

We see (after clearing s^m): $\widehat{g}_i(\mathbf{y}, s = 0) = \partial_{\mathbf{v}} g_i(\mathbf{y})$. So for every inner normal $\mathbf{v}$, there is one homotopy which defines $V(C)$ solution paths.

Example 2.11 The lifting of the supports of the system (22) (first introduced in Example 2.8) defines not only a regular mixed-cell configuration (see Example 2.10), but also a set of so-called polyhedral homotopies. For every mixed cell, there is one homotopy. We start by introducing a continuation parameter t whose exponent is defined by the lifting:

$$\begin{aligned} \widehat{g} &= (\widehat{g}_1, \widehat{g}_2) \\ &= \begin{cases} c_{31}^{(1)} x_1^3 x_2 t^1 + c_{12}^{(1)} x_1 x_2^2 t^1 + c_{00}^{(1)} t^0 = 0 \\ c_{40}^{(2)} x_1^4 t^0 + c_{11}^{(2)} x_1 x_2 t^1 + c_{00}^{(2)} t^1 = 0 \end{cases} \end{aligned} \qquad \begin{aligned} \widehat{\mathcal{A}} &= (\widehat{A}_1, \widehat{A}_2) \\ \widehat{A}_1 &= \{(3,1,1), (1,2,1), (0,0,0)\} \\ \widehat{A}_2 &= \{(4,0,0), (1,1,1), (0,0,1)\} \end{aligned} \tag{35}$$

Notice the introduction of the random coefficients $c_{ij}^{(k)} \in \mathbb{C}$. The system $\widehat{g}(\mathbf{x}, t = 1) = \mathbf{0}$ is a random coefficient start system to solve any system $f(\mathbf{x}) = \mathbf{0}$ with supports $\mathcal{A}$. To solve $\widehat{g}(\mathbf{x}, t = 1) = \mathbf{0}$, we change coordinates, using the inner normals to the mixed cells. The first mixed cell $\widehat{\mathcal{C}}^{(1)}$ of (29) of Example 2.10 has inner normal $\mathbf{v} = (1, -4, 7)$. The coordinate change $x_1 = y_1 s^1$, $x_2 = y_2 s^{-4}$, $t = s^7$ on $\widehat{g}(\mathbf{x}, t) = \mathbf{0}$ produces the homotopy $\widehat{g}(\mathbf{y}, s) = \mathbf{0}$:

$$\left\{ \begin{array}{l} c_{31}^{(1)} y_1^3 y_2 s^{3\cdot 1+1\cdot(-4)+1\cdot 7} + c_{12}^{(1)} y_1 y_2^2 s^{1\cdot 1+2\cdot(-4)+1\cdot 7} + c_{00}^{(1)} s^{0\cdot 7} = c_{31}^{(1)} y_1^3 y_2 s^6 + c_{12}^{(1)} y_1 y_2^2 + c_{00}^{(1)} = 0 \\ c_{40}^{(2)} y_1^4 s^{4\cdot 1+0\cdot(-4)+0\cdot 7} + c_{11}^{(2)} y_1 y_2 s^{1\cdot 1+1\cdot(-4)+1\cdot 7} + c_{00}^{(2)} s^{1\cdot 7} = \left(c_{40}^{(2)} y_1^4 + c_{11}^{(2)} y_1 y_2 + c_{00}^{(2)} s^3 \right) s^4 = 0 \end{array} \right. \tag{36}$$

After clearing the s^4 from the second equation, we observe that the exponent vector of those terms which occur with s^0 belong to the support of the mixed cell $\widehat{\mathcal{C}}^{(1)}$. At $s = 0$, the continuation starts at the seven solutions of the binomial system with supports $\mathcal{C}^{(1)}$. The construction is similar for the second cell $\widehat{\mathcal{C}}^{(2)}$. $\diamond$

Algorithm 2.11 solve a random coefficient system
Input: $g(\mathbf{x}) = \mathbf{0}$, a system with random coefficients;
$\widehat{\mathcal{A}}$, lifted supports of g;
$\mathcal{C}$ a regular mixed-cell configuration.
Output: all solutions in $(\mathbb{C}^*)^n$ to $g(\mathbf{x}) = \mathbf{0}$.
Implementation: `phc -m`.

Theorem 2.11 (Bernshteĭn's first theorem) *A generic system $g(\mathbf{x}) = \mathbf{0}$ has exactly as many isolated solutions in $(\mathbb{C}^*)^n$ as the mixed volume of its Newton polytopes.*

Exercise 2.11 Consider again the cyclic 7-roots problem (30). Use the regular mixed-cell configuration computed in Exercise 2.10 to create a random coefficient start system. With this start system, solve the system. Verify the results by running `phc -b` on this system. $\diamond$

Definition 2.12 Let $\mathcal{A} = (A_1, A_2, \ldots, A_n)$ be the supports of the polynomials in $f = (f_1, f_2, \ldots, f_n)$. Any nonzero vector $\mathbf{v}$ defines the face system

$$\partial_{\mathbf{v}} f = (\partial_{\mathbf{v}} f_1, \partial_{\mathbf{v}} f_2, \ldots, \partial_{\mathbf{v}} f_n) \quad \text{with} \quad \partial_{\mathbf{v}} f_i = \sum_{\mathbf{a} \in \partial_{\mathbf{v}} A_i} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}. \tag{37}$$

whose supports $\partial_{\mathbf{v}} \mathcal{A} = (\partial_{\mathbf{v}} A_1, \partial_{\mathbf{v}} A_2, \ldots, \partial_{\mathbf{v}} A_n)$ span faces of the corresponding Newton polytopes.

The following example illustrates the notion of Newton polytopes being in *general position*.

Example 2.12 Consider the following polynomial system:

$$f(\mathbf{x}) = \left\{ \begin{array}{r} c_{111} x_1 x_2 + c_{110} x_1 + c_{101} x_2 + c_{100} = 0 \\ c_{222} x_1^2 x_2^2 + c_{210} x_1 + c_{201} x_2 = 0 \end{array} \right. \tag{38}$$

where the coefficients c_{ijk} are nonzero complex numbers. Figure 3 shows the two Newton polygons of the polynomials in the system (38).

As the face systems can have no solution with all components different from zero, the system must have exactly as many isolated solutions as the mixed volume, and this *for any nonzero choice of the coefficients*. $\diamond$

Algorithm 2.12 polyhedral end game
Input: $h(\mathbf{x}, t) = \mathbf{0}$, a homotopy;
$\mathbf{x}^t$, solution for $t \approx 1$.
Output: either $\mathbf{x}^1$ is a regular solution of $h(\mathbf{x}, t) = \mathbf{0}$,
or $\mathbf{x}^1$ is a solution of a face system $\partial_{\mathbf{v}} h(\mathbf{x}, t) = \mathbf{0}$.
Implementation: `phc -p`.

Figure 3: The Newton polygons of the system (38). Observe that for any inner normal $\mathbf{v} \neq \mathbf{0}$: $\partial_\omega A_1 + \partial_\omega A_2 \leq 3$. This means that all face systems have no solutions in $(\mathbb{C} \setminus \{0\})^2$.

Theorem 2.12 (Bernshteĭn's second theorem) *If $\forall \mathbf{v} \neq \mathbf{0}$, $\partial_{\mathbf{v}} f(\mathbf{x}) = \mathbf{0}$ has no solutions in $(\mathbb{C}^*)^n$, then $V(\mathcal{A})$ is exact and all solutions are isolated. Otherwise, for $V(\mathcal{A}) \neq 0$: $V(\mathcal{A}) >$ #isolated solutions.*

Exercise 2.12 Consider the system (5), introduced in Example 2.3. Take $a = 0$ and show that the Newton polytopes of this system are not in general position. (*Hint:* for $a = 0$, all monomials in the system have the same degree). ◇

Bibliographic Notes. Recommended introductions to polyhedral methods to solve polynomial systems are [20], [126], and [127]. See [41] for the relation with discriminants and resultants. A survey on toric resultants is in [27]. For a nice introduction to polytopes, see [146]. For an overview on efficient techniques to compute mixed volumes, see [70]. Implementing constructive proofs of Bernshteĭn's theorems [9], polyhedral homotopies were introduced in [52] and [136] to solve sparse systems more efficiently. The mixed volume was nicknamed the BKK bound in [13] to honor Bernshteĭn [9], Kushnirenko [62], and Khovanskiĭ [58]. The polyhedral version of the Cayley trick is due to Bernd Sturmfels, see [123] and [41]. Another application of this trick is in [51]. Mixed subdivisions were introduced in [124, 125] to generalize Viro's method [55, 56] to complete intersections. The lift-and-prune approach was presented in [28]. Exploitation of symmetry was studied in [133] and the dynamic lifting of [134] led to incremental polyhedral continuation. See [130] for a Toric Newton, as a sequel to polyhedral endgames [54], implementing Bernshteĭn's second theorem. Extensions to count all affine roots (also those with zero components) were proposed in [30], [37], [53], [81], [100, 101], and [102]. Very efficient calculations of mixed volumes are described in [21], [34, 35, 38], [71], and [128]. The stability of numerical continuation methods for polyhedral methods is addressed in [36] and [59].

2.4 Isolated Singularities computed by Deflation

The convergence of Newton's method slows down to a halt when approaching a singular solution. In this section we show how to restore the quadratic convergence by deflation, which is an effective reconditioner, allowing the accurate computation of isolated singularities. After we have located the isolated singularity accurately, we can compute its multiplicity, as the number of elements in a basis for the dual space.

Definition 2.13 Let R be the rank of an isolated singular root $\mathbf{x}^*$ of a system $f(\mathbf{x}) = \mathbf{0}$ of N equations in n unknowns. We define a *deflation of f at $\mathbf{x}^*$* as the system

$$F(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} f(\mathbf{x}) & = & \mathbf{0} \\ \frac{\partial f}{\partial \mathbf{x}} B \boldsymbol{\lambda} & = & \mathbf{0} \\ \mathbf{h}\boldsymbol{\lambda} & = & 1 \end{cases} \tag{39}$$

using $R+1$ multipliers $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{R+1})$ as extra variables, together with an N-by-$(R+1)$ matrix B of random complex numbers and a random vector of $R+1$ complex coefficients $\mathbf{h}$.

Example 2.13 Consider the following simple example

$$f(x,y) = \begin{cases} x^2 = 0 \\ xy = 0 \\ y^2 = 0 \end{cases} \quad A(x,y) = \begin{bmatrix} 2x & 0 \\ y & x \\ 0 & 2y \end{bmatrix} \quad \begin{matrix} \mathbf{x}^* = (0,0), m = 3 \\ \text{Rank}(A(\mathbf{x}^*)) = 0 \end{matrix} \tag{40}$$

In Example 2.14 we will show that $\mathbf{x}^*$ occurs with multiplicity $m = 3$. Since the rank is zero, we right multiply $A(\mathbf{x})$ with a random 2-by-1 matrix $B = [b_{11}\ b_{21}]^T$ and introduce one multiplier λ_1, so we obtain:

$$\mathbf{Dfl}(\mathbf{f})(\mathbf{x},\mathbf{y},\lambda_{\mathbf{1}}) = \begin{cases} f(x,y) = 0 \\ \begin{bmatrix} 2x & 0 \\ y & x \\ 0 & 2y \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} \lambda_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ h_1\lambda_1 = 1, \quad \text{random } h_1 \in \mathbb{C} \end{cases} \tag{41}$$

For a certain value λ_1^* of the multiplier λ_1, $\mathbf{Dfl}(\mathbf{f})(\mathbf{x},\mathbf{y},\lambda_{\mathbf{1}}) = \mathbf{0}$ has $(0,0,\lambda_1^*)$ as regular zero. ◇

Algorithm 2.13 Newton with deflation
Input: $f(\mathbf{x}) = \mathbf{0}$, a polynomial system;
$\mathbf{x}^0$ is an initial guess for the solution.
Output: $F(\mathbf{x},\boldsymbol{\lambda}) = \mathbf{0}$, a deflated system,
$(\mathbf{x}^*,\boldsymbol{\lambda}^*)$ is a regular solution of $F(\mathbf{x},\boldsymbol{\lambda}) = \mathbf{0}$.
Implementation: `phc -v`.

Theorem 2.13 (#deflations to restore quadratic convergence) *The number of deflations needed to restore the quadratic convergence of Newton's method converging to an isolated solution is strictly less than the multiplicity of the isolated solution.*

Exercise 2.13 Consider again the system (1) first introduced in Example 2.1. Apply the deflation algorithm (using `phc -v`) to the solutions found at the end of the solution paths. ◇

Once we have accurately located the isolated singularity, we can compute its multiplicity. For this, we need to look at the dual space of differential operators.

Definition 2.14 We abbreviate the differential operator $\frac{\partial^{a_1+a_2+\cdots+a_n}}{\partial x_1^{a_1}\partial x_2^{a_2}\cdots\partial x_n^{a_n}}$ by $\partial_{\mathbf{a}}$, with $\mathbf{a} = (a_1, a_2, \ldots, a_n)$. Denote $|\mathbf{a}| = a_1 + a_2 + \cdots + a_n$. For any $\mathbf{z}$, we define the linear operator $\partial_{\mathbf{a}}[\mathbf{z}]$ as

$$\partial_{\mathbf{a}}[\mathbf{z}] : \mathbb{C}[\mathbf{x}] \to \mathbb{C} : p \mapsto \frac{1}{a_1!a_2!\cdots a_n!}\left(\frac{\partial^{|\mathbf{a}|}p}{\partial x_1^{a_1}\partial x_2^{a_2}\cdots\partial x_n^{a_n}}\right)(\mathbf{z}). \tag{42}$$

For a system $f(\mathbf{x}) = \mathbf{0}$ and root $\mathbf{x}^*$, consider the sequence of *multiplicity matrices* S_k, where the columns of S_k are indexed by $\partial_{\mathbf{a}}$, for $|\mathbf{a}| \leq k$, and the rows by $\mathbf{x}^{\mathbf{b}}f_i$, for $|\mathbf{b}| \leq k-1$. The entries of S_k are the values of $\partial_{\mathbf{a}}(\mathbf{x}^{\mathbf{b}}f_i)$, evaluated at $\mathbf{x}^*$. By convention, $S_0 = f(\mathbf{x}^*)$.

Algorithm 2.14 dual space and multiplicity
Input: $f(\mathbf{x}) = \mathbf{0}$, a polynomial system;
$\mathbf{x}^*$ is a solution to $f(\mathbf{x}) = \mathbf{0}$.
Output: differential operators which span the dual space $D[\mathbf{x}^*]$,
$m = \dim(D[\mathbf{x}^*])$ is the multiplicity of the solution $\mathbf{x}^*$.
Implementation: `phc -v`.

Theorem 2.14 (the Hilbert function) *The Hilbert function $H(k)$ and multiplicity m are*

$$H(k) = \text{nullity}(S_k) - \text{nullity}(S_{k-1}), k = 1, 2, \ldots \quad m = \sum_{k=1}^{\infty} H(k). \tag{43}$$

Example 2.14 Consider again the system (40) introduced in Example 2.13. The multiplicity of $\mathbf{x}^* = (0, 0)$ is 3 because the dual space $D[\mathbf{x}^*] = \text{span}\{\partial_{00}[\mathbf{x}^*], \partial_{10}[\mathbf{x}^*], \partial_{01}[\mathbf{x}^*]\}$ is spanned by three elements. The calculation of the basis for $D[\mathbf{x}^*]$ is illustrated below. The algorithm stops when $\text{Nullity}(S_1) = \text{Nullity}(S_2)$:

		$\overbrace{\quad}^{\lvert a\rvert=0}$	$\overbrace{\quad\quad}^{\lvert a\rvert=1}$		$\overbrace{\quad\quad\quad}^{\lvert a\rvert=2}$		
		∂_{00}	∂_{10}	∂_{01}	∂_{20}	∂_{11}	∂_{02}
	f_1	0	0	0	2	0	0
	f_2	0	0	0	0	1	0
S_1	f_3	0	0	0	0	0	2
	xf_1	0	0	0	0	0	0
	xf_2	0	0	0	0	0	0
	xf_3	0	0	0	0	0	0
	yf_1	0	0	0	0	0	0
	yf_2	0	0	0	0	0	0
S_2	yf_3	0	0	0	0	0	0

For this simple example, we can take the natural basis ∂_{00}, ∂_{10}, and ∂_{01} for the null space of S_1. We also immediately see that the dimension of the null space of S_2 is the same as the nullity of S_2. ◇

Exercise 2.14 Compute the multiplicity of the roots of the system (1), after the deflation has reconditioned the singular roots in Exercise 2.13. ◇

Bibliographic Notes. In numerical analysis, Newton's method in the presence of singularities has been studied widely. A classic book is [99]. Research up to the mid eighties is surveyed in [48]. The book [45] describes various techniques in the context of path following methods. These techniques are know as Liapunov-Schmidt reduction [1],[5], see also [79, 80], [24], [60, 61]. To deal with singular solutions at the end of solution paths defined by polynomial homotopies, so-called endgames were developed in [90, 91, 92] and [120]. Newton's method for overdetermined systems was studied in [23]. Computer algebra uses the notion of standard bases, see [82] and its generalization in [46], as implemented in Singular [47]. In numerical polynomial algebra [121], one uses duality, see [94] and algorithms to compute the multiplicity are given in [122] and [22]. Algorithm 2.14 is based on [22], which uses duality in its analysis of Algorithm 2.13, the deflation algorithm of [67]. The deflation is inspired on [98], see also [95], [96], and [97] A symbolic deflation algorithm is in [63].

3 Lecture II: Numerical Irreducible Decomposition

The goal of the second lecture is to identify the main algorithms needed to compute a numerical irreducible decomposition.

There are four parts in this lecture. We first define numerical representations of pure dimensional solutions sets, what we call "witness sets". With a cascade of homotopies we compute witness supersets for all solutions of all dimensions. After filtering points on higher dimensional solution sets from the witness supersets we obtain witness sets for all pure dimensional solution sets of a polynomial system. In the third section we describe factorization methods to decompose pure dimensional solution sets into irreducible components. We end this lecture by introducing diagonal homotopies to intersect witness sets. Diagonal homotopies give rise to an equation-by-equation solver, which can be used as a blackbox solver.

3.1 Witness Sets represent Pure Dimensional Solution Sets

Adding extra equations or extra variables we can reduce the case of an unequal number of equations and unknowns to the square case.

Definition 3.1 For a positive natural number k and a polynomial system $f(\mathbf{x}) = \mathbf{0}$ of N equations in n unknowns, an *embedding of f with k hyperplanes* is for $N = n$:

$$\mathcal{E}_k(f)(\mathbf{x},\mathbf{z}) = \left\{ \begin{array}{rl} f(\mathbf{x}) + B\mathbf{z} = \mathbf{0} & \mathbf{z} = (z_1, z_2, \ldots, z_k), \\ C\mathbf{x} + \mathbf{d} + \mathbf{z} = \mathbf{0} & B \in \mathbb{C}^{N\times k}, C \in \mathbb{C}^{k\times n}, \mathbf{d} \in \mathbb{C}^{k\times 1}, \end{array} \right. \tag{44}$$

where B, C, and $\mathbf{d}$ are random complex matrices, $\mathbf{z}$ contains k *slack variables*. When $N > n$, we need $r = N - n$ more slack variables $\mathbf{z}_-$ and extend the embedding as follows:

$$\mathcal{E}_k(f)(\mathbf{x},\mathbf{z}) = \left\{ \begin{array}{rl} f(\mathbf{x}) + [B_- \; B] \left[\begin{array}{c} \mathbf{z}_- \\ \mathbf{z} \end{array} \right] = \mathbf{0} & \mathbf{z}_- = (z_{-r+1}, \ldots, z_{-1}, z_0), \mathbf{z} = (z_1, z_2, \ldots, z_k), \\ C\mathbf{x} + \mathbf{d} + \mathbf{z} = \mathbf{0} & B_- \in \mathbb{C}^{N\times r}, B \in \mathbb{C}^{N\times k}, C \in \mathbb{C}^{k\times n}, \mathbf{d} \in \mathbb{C}^{k\times 1}. \end{array} \right. \tag{45}$$

Only solutions with $\mathbf{z}_- = \mathbf{0}$ are relevant. When $N < n$, we can set the first $s = n - N$ of the slack variables to zero so the embedding then becomes

$$\mathcal{E}_k(f)(\mathbf{x},\mathbf{z}) = \left\{ \begin{array}{rl} f(\mathbf{x}) + B\mathbf{z}_0 = \mathbf{0} & \mathbf{z}_0 = (z_1, z_2, \ldots, z_s), \mathbf{z} = (z_1, z_2, \ldots, z_k) \\ \mathbf{z}_0 = \mathbf{0} & B \in \mathbb{C}^{N\times k}, C \in \mathbb{C}^{k\times n}, \mathbf{d} \in \mathbb{C}^{k\times 1}. \\ C\mathbf{x} + \mathbf{d} + \mathbf{z} = \mathbf{0} & \end{array} \right. \tag{46}$$

Observe that for all N and n, the index of the last slack variable always equals k, which is the dimension of the relevant solution set.

Example 3.1 Let f be a system of N equations in n variables. For $(N, n) = (2, 2)$ and $k = 1$, we use the following embedding:

$$\left\{ \begin{array}{r} f_1(x_1, x_2) + b_{11} z_1 = 0 \\ f_2(x_1, x_2) + b_{21} z_1 = 0 \\ c_0 + c_1 x_1 + c_2 x_2 + z_1 = 0 \end{array} \right. \tag{47}$$

where the coefficients b_{11}, b_{21}, c_0, c_1, and c_2 are random complex numbers. For $(N, n) = (3, 2)$, and $(N, n) = (2, 3)$, we have the following respective embeddings for $k = 1$:

$$\left\{ \begin{array}{r} f_1(x_1, x_2) + b_{10} z_0 + b_{11} z_1 = 0 \\ f_2(x_1, x_2) + b_{20} z_0 + b_{21} z_1 = 0 \\ f_3(x_1, x_2) + b_{30} z_0 + b_{31} z_1 = 0 \\ c_0 + c_1 x_1 + c_2 x_2 + z_1 = 0 \end{array} \right. \quad \left\{ \begin{array}{r} f_1(x_1, x_2, x_3) + b_{11} z_1 = 0 \\ f_2(x_1, x_2, x_3) + b_{21} z_1 = 0 \\ z_1 = 0 \\ c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3 + z_1 = 0 \end{array} \right. \tag{48}$$

In all three cases, the embedded system has as many equations as unknowns. Only solutions with zero slack variables belong to the one dimensional solution set. ◇

Algorithm 3.1 embed a polynomial system
Input: $f(\mathbf{x}) = \mathbf{0}$, a polynomial system;
$k > 0$, a positive number.
Output: $\mathcal{E}_k(f)(\mathbf{x},\mathbf{z}) = \mathbf{0}$, $\mathbf{z} = (z_1, z_2, \ldots, z_k)$ are slack variables,
the last k equations in $\mathcal{E}_k(f)$ are linear.
Implementation: `phc -c`.

Theorem 3.1 (Bertini's theorem) *Suppose $f(\mathbf{x}) = \mathbf{0}$ defines only one k-dimensional solution set Z. Then*

$$\deg(Z) = \#\{ \; (\mathbf{x},\mathbf{z}) \in \mathbb{C}^{n+k} \mid \mathcal{E}_k(f)(\mathbf{x},\mathbf{z}) = \mathbf{0} \text{ and } \mathbf{z} = \mathbf{0} \; \}. \tag{49}$$

Exercise 3.1 The equations defining the adjacent 2-by-2-minors (coming from [25], see also [49]) of a general 2-by-4-matrix are

$$\left\{ \begin{array}{l} x_{11}x_{22} - x_{21}x_{12} = 0 \\ x_{12}x_{23} - x_{22}x_{13} = 0 \\ x_{13}x_{24} - x_{23}x_{14} = 0 \end{array} \right. \quad \text{for the matrix} \quad \left[\begin{array}{cccc} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \end{array} \right]. \tag{50}$$

With 3 polynomials in 8 variables, we may expect a solution set of dimension five. Use `phc -c` to construct an embedding for this system. Solve the embedded system (with `phc -b`) and verify that the system has indeed a five dimensional solution set. What is the degree of this solution set? ◇

Definition 3.2 A *witness point* is a solution of a polynomial system which lies on a set of generic hyperplanes. The number of generic hyperplanes used to isolate a point from a solution component equals the *dimension of the solution set*. The number of witness points on one component cut out by the same set of generic hyperplanes equals the *degree of the solution set*. A *witness set* for a k-dimensional solution set consists of k random hyperplanes and the set of isolated solutions comprising the intersection of the component with those hyperplanes.

Algorithm 3.2 sampling a solution set
Input: W_L is witness set for k hyperplanes L;
K is a new set of k hyperplanes.
Output: W_K is witness set for hyperplanes K.
Implementation: `phc -p`.

Theorem 3.2 (numerical elimination) *Suppose the system $f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, with $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ defines a k-dimensional solution set. Restricting the k hyperplanes in the embedding $\mathcal{E}_k(f)$ to have only nonzero coefficients with the y-variables, we can eliminate $\mathbf{y}$ from the solution set and via interpolation construct defining equations for the projection of the solution set onto the x-variables.*

Example 3.2 The twisted cubic is a curve in three space defined by the intersection of a quadratic and a cubic surface: $y = x^2$ and $z = x^3$ respectively. These two surfaces are the projections of the twisted cubic on the respective (x, y) and (x, z)-planes. We can realize these projections by restricting the slicing hyperplanes appropriately. Consider the following three systems:

$$\left\{ \begin{array}{l} y - x^2 = 0 \\ z - x^3 = 0 \\ c_0 + c_1x + c_2y + c_3z = 0 \end{array} \right. \quad \left\{ \begin{array}{l} y - x^2 = 0 \\ z - x^3 = 0 \\ c_0 + c_1x + c_2y = 0 \end{array} \right. \quad \left\{ \begin{array}{l} y - x^2 = 0 \\ z - x^3 = 0 \\ c_0 + c_3z = 0 \end{array} \right. \tag{51}$$

Substituting y by x^2 and z by x^3 in the last equation of the first system, we obtain a cubic in one single variable x, defining three generic points on the twisted cubic. Similarly, for the second system, we find only two solutions. Dropping the z from the samples, i.e. $(x, y, z) \mapsto (x, y)$, we sample from $y - x^2 = 0$. The last equation in the third system defines a plane parallel to the (x, y)-plane. Applying $(x, y, z) \mapsto (x, z)$ to the sampled points executes the projection of the twisted cubic on $z - x^3 = 0$. With multivariate interpolation through the projected samples we eliminate in this way the variables z or y from the system. ◇

Exercise 3.2 The following polynomial system represents a spatial Burmester problems [12], used in body

guidance problems.

$$
\begin{aligned}
&f(x_1,x_2,x_3,y_1,y_2,y_3) = \\
&\left\{\begin{array}{r}
-1.10341193720649E+00y_1x_1 - 5.86815682189835E-01y_1x_2 - 7.92036085497022E-01y_1x_3 \\
-5.77733959201612E-01y_2x_1 + 3.87368458550248E-01y_2x_2 + 1.28043505098815E+00y_2x_3 \\
+1.04041310144849E+00y_3x_1 + 1.07931536931741E-01y_3x_2 - 3.40083678408469E-01y_3x_3 \\
-5.68456000000000E-01y_1 + 1.05510000000000E-02y_2 + 8.21000000000000E-04y_3 \\
-1.95379855738832E-01x_1 - 1.94409521104426E-01x_2 - 1.04439406278041E+00x_3 \\
+3.41541444787000E-01 = 0 \\
-1.27667983027595E+00y_1x_1 - 8.22293023657284E-01y_1x_2 + 9.85781640537901E-01y_1x_3 \\
+1.10445261409581E+00y_2x_1 - 8.28754979219007E-02y_2x_2 + 1.48463358281623E+00y_2x_3 \\
+4.94115593345597E-01y_3x_1 + 2.03347173764983E-01y_3x_2 - 4.33616705996826E-02y_3x_3 \\
+9.33643000000000E-01y_1 - 1.33829000000000E-01y_2 + 1.04241900000000E+00y_3 \\
+2.63041345590506E-01x_1 - 1.03548264961166E+00x_2 - 7.93223555891750E-01x_3 \\
+4.19707164888500E-01 = 0 \\
-8.83804023497621E-01y_1x_1 - 1.78651736017982E-01y_1x_2 - 7.38492559710131E-01y_1x_3 \\
+4.43434838274957E-01y_2x_1 + 8.09639929536600E-01y_2x_2 + 1.38546622633458E+00y_2x_3 \\
+1.56573040484672E+00y_3x_1 - 8.99276927801036E-01y_3x_2 - 4.05489960274242E-01y_3x_3 \\
+1.07621800000000E+00y_1 + 3.81121000000000E-01y_2 - 6.69739000000000E-01y_3 \\
+7.52673504306374E-01x_1 - 1.28749695700861E+00x_2 + 1.47238424191569E-01x_3 \\
+7.54400309061000E-01 = 0 \\
-1.24212145914842E+00y_1x_1 - 1.23465051231272E+00y_1x_2 - 4.85226728194913E-01y_1x_3 \\
-4.86814272583431E-01y_2x_1 - 1.94381663901655E-01y_2x_2 + 1.63104483875388E+00y_2x_3 \\
+1.11153625371862E-01y_3x_1 + 6.84956479647339E-03y_3x_2 - 6.08528904330562E-01y_3x_3 \\
+9.04135000000000E-01y_1 + 7.81815000000000E-01y_2 - 3.95866000000000E-01y_3 \\
+5.18838101619745E-01x_1 + 5.93430553942240E-01x_2 - 6.26769769282934E-01x_3 \\
+7.12048812833000E-01 = 0 \\
-6.46023972517329E-01y_1x_1 - 1.29375652786430E+00y_1x_2 - 5.84812917010785E-01y_1x_3 \\
+1.62705334082809E-01y_2x_1 + 6.52017519795460E-01y_2x_2 + 2.86172998155648E-01y_2x_3 \\
+1.58054783715492E+00y_3x_1 - 6.31169332666938E-01y_3x_2 - 1.85713987157407E-01y_3x_3 \\
-5.74712000000000E-01y_1 - 7.49368000000000E-01y_2 + 6.30445000000000E-01y_3 \\
-4.55403271303103E-01x_1 - 5.80692051452173E-01x_2 - 1.21733476968808E+00x_3 \\
+5.68373481377500E-01 = 0
\end{array}\right.
\end{aligned}
\tag{52}
$$

The equations in this polynomial system define a curve of degree twenty. Verify that the degree of the projection onto the (x_1, x_2, x_3) or (y_1, y_2, y_3) coordinates equals ten. ◇

Definition 3.3 Given a polynomial system $f(\mathbf{x}) = \mathbf{0}$, a witness set W_L, and a point $\mathbf{y} \in \mathbb{C}^n$, we define the *membership homotopy* as

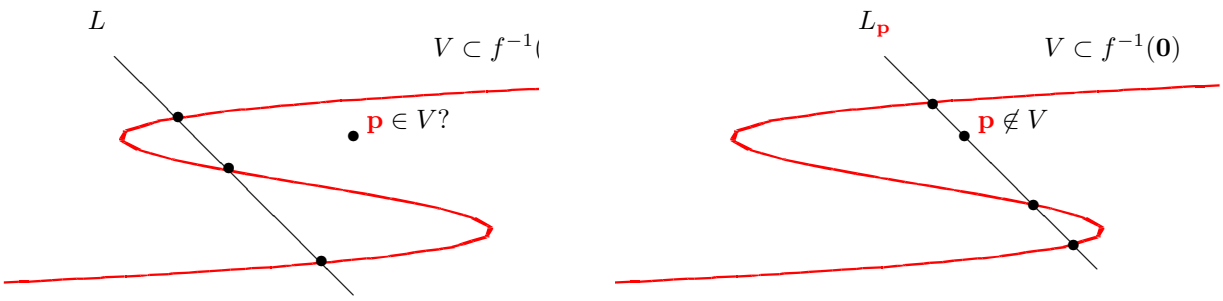
$$
h(\mathbf{x}, t) = (1-t)\begin{pmatrix} f(\mathbf{x}) = \mathbf{0} \\ L(\mathbf{x}) = \mathbf{0} \end{pmatrix} + t\begin{pmatrix} f(\mathbf{x}) = \mathbf{0} \\ L(\mathbf{x}) = L(\mathbf{y}) \end{pmatrix} = \mathbf{0}. \tag{53}
$$

Solution paths start at $t = 0$ at the witness points in W_L and end at $t = 1$ at the solutions of a new witness set with hyperplanes passing through the point $\mathbf{y}$.

Example 3.3 Figure 4 illustrates the homotopy membership test for a cubic component V of a curve $f^{-1}(0)$. We see that the test point does not belong to the component V because it is not one of the three points in the new witness set. As the given point $\mathbf{p}$ may be known only with very limited accuracy (it could be a highly singular point), it is very important to note that the paths move from solutions at $L(\mathbf{x}) = \mathbf{0}$ to the hyperplane $L_{\mathbf{p}}(\mathbf{x}) = 0$ which passes through $\mathbf{p}$. ◇

Algorithm 3.3 homotopy membership test

Input: W_L is witness set for a solution set;
$\mathbf{x}^0$ is any point in space.



V is represented by a witness set $V \cap L$.

new witness set $V \cap L_{\mathbf{p}}$, $\mathbf{p} \notin V \cap L_{\mathbf{p}}$

Figure 4: The homotopy membership test on a cubic component.

Output: yes or no, depending whether $\mathbf{x}^0$ belongs to the set.
Implementation: `phc -f`.

Theorem 3.3 (probability one membership criterium) *Executing the membership homotopy on a point for a given witness set determines with probability one if the point lies in the set or not.*

Exercise 3.3 The following polynomial system

$$\begin{cases} (y - x^2)(y + z) = 0 \\ (z - x^3)(y - z) = 0 \end{cases} \tag{54}$$

was used in [112] to illustrate the homotopy membership test. We see there are four irreducible factors. First use `phc -c` to create an embedding, then with `phc -b` compute witness points for the one dimensional solution set, followed by `phc -f` to have witness sets for the four irreducible factors. Choose four points, one point on each irreducible component. Conduct the membership test using `phc -f` to verify whether the chosen point indeed belongs to only one component and not on the other three components. ◇

Bibliographic Notes. The use of generic points was outline in the paper on numerical algebraic geometry [118] and plays a key role in the numerical irreducible decomposition [107]. The homotopy membership test was proposed in [108], extension to deal with singular solution sets were described in [110]. See [113] for applications tot mechanical design. Witness sets are equivalent to lifting fibers in a geometric resolution [42, 43, 44, 64]. See also [103] for a tutorial on the complexity of these algorithms.

3.2 A Cascade of Homotopies to Compute Witness Supersets

Definition 3.4 For an embedding $\mathcal{E}_i(f)(\mathbf{x}, \mathbf{z}) = \mathbf{0}$ we define a *cascading embedding* $\mathcal{E}_{i-1}(f)(\mathbf{x}, \mathbf{z}) = \mathbf{0}$ by removing the last ith hyperplane from $\mathcal{E}_i(f)$. Then we can define a sequence of homotopies or a *cascading homotopy*

$$h_i(\mathbf{x}, \mathbf{z}, t) = (1 - t)\mathcal{E}_i(f)(\mathbf{x}, \mathbf{z}) + t \begin{pmatrix} \mathcal{E}_{i-1}(f)(\mathbf{x}, \mathbf{z}) \\ z_i \end{pmatrix} = \mathbf{0} \tag{55}$$

which is a homotopy between cascading embeddings, removing the ith hyperplane in each stage. A *witness superset* for an i-dimensional solution set (also called an i-dimensional witness superset) contains a witness set and additional points on higher dimensional solution sets.

Example 3.4 The polynomial system

$$f(\mathbf{x}) = \begin{cases} (x_1^2 - x_2)(x_1 - 0.5) = 0 \\ (x_1^3 - x_3)(x_2 - 0.5) = 0 \\ (x_1 x_2 - x_3)(x_3 - 0.5) = 0 \end{cases} \tag{56}$$

has a one dimensional solution set, i.e.: the twisted cubic, and four isolated points. To compute numerical representations of the twisted cubic and the four isolated points, as given by the solution set of the system (56), we use the following homotopy:

$$H(\mathbf{x}, z_1, t) = \begin{bmatrix} \begin{bmatrix} (x_1^2 - x_2)(x_1 - 0.5) \\ (x_1^3 - x_3)(x_2 - 0.5) \\ (x_1 x_2 - x_3)(x_3 - 0.5) \end{bmatrix} & + & t \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} z_1 \\ t\,(c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3) & + & z_1 \end{bmatrix} = \mathbf{0} \tag{57}$$

At $t = 1$: $H(\mathbf{x}, z_1, t) = \mathcal{E}_1(f)(\mathbf{x}, z_1) = \mathbf{0}$. At $t = 0$: $H(\mathbf{x}, z_1, t) = f(\mathbf{x}) = \mathbf{0}$. As t goes from 1 to 0, the hyperplane is removed from the embedded system, and z_1 is forced to zero. Figure 5 summarizes the number of solution paths traced in the cascade of homotopies.

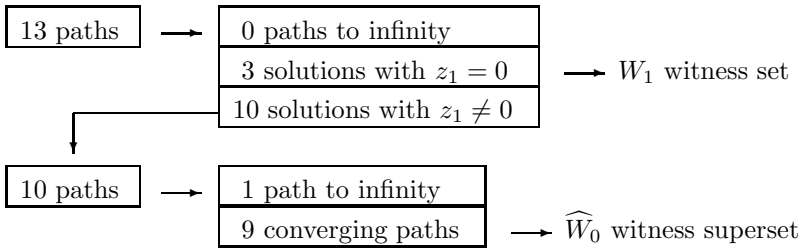


Figure 5: Summary of the #paths in the cascade on system (56), to compute a witness set for the twisted cubic and approximations to the four isolated solutions.

The set $\widehat{W}_0$ of Figure 5 contains, in addition to the four isolated roots, also points on the twisted cubic. The points in $\widehat{W}_0$ which lie on the twisted cubic are considered *junk* and must be filtered out, using the homotopy membership test. ◇

Algorithm 3.4 cascade of homotopies
Input: $\mathcal{E}_k(f)(\mathbf{x}, \mathbf{z}) = \mathbf{0}$, $\mathbf{z} = (z_1, z_2, \ldots, z_k)$;
all solutions with $\mathbf{z} \neq \mathbf{0}$.
Output: $[W_0, W_1, \ldots, W_{k-1}]$, W_i is an i-dimensional witness superset.
Implementation: `phc -c`.

Theorem 3.4 (superwitness set generation) *For an embedding $\mathcal{E}_i(f)(\mathbf{x}, \mathbf{z})$ of $f(\mathbf{x}) = \mathbf{0}$ with i random hyperplanes and i slack variables $\mathbf{z} = (z_1, z_2, \ldots, z_i)$, we have (1) solutions with $\mathbf{z} = \mathbf{0}$ contain $\deg W$ generic points on every i-dimensional component W of $f(\mathbf{x}) = \mathbf{0}$; (2) solutions with $\mathbf{z} \neq \mathbf{0}$ are regular; and (3) the solution paths defined by the cascading homotopy starting at $t = 0$ with all solutions with $z_i \neq 0$ reach at $t = 1$ all isolated solutions of $\mathcal{E}_{i-1}(f)(\mathbf{x}, \mathbf{z}) = \mathbf{0}$.*

Exercise 3.4 A 7-bar mechanism (used in [107]) which permits motion gives rise to the following poly-

nomial system:

$$
f(\mathbf{t},\mathbf{T}) = \left\{ \begin{array}{l}
t_1T_1 - 1 = 0 \\
t_2T_2 - 1 = 0 \\
t_3T_3 - 1 = 0 \\
t_4T_4 - 1 = 0 \\
t_5T_5 - 1 = 0 \\
t_6T_6 - 1 = 0 \\
0.71035834160605t_1 + 0.46t_2 - 0.41t_3 + 0.24076130055512 + 1.07248215701824i = 0 \\
(-0.11 + 0.49i)t_2 + 0.41t_3 - 0.50219518117959t_4 + 0.41t_5 = 0 \\
0.50219518117959t_4 + (-0.09804347826087 + 0.43673913043478i)t_5 \\
\qquad -0.77551855666366t_6 - 1.2 = 0 \\
0.71035834160605T_1 + 0.46T_2 - 0.41T_3 + 0.24076130055512 - 1.07248215701824i = 0 \\
(-0.11 - 0.49i)T_2 + 0.41T_3 - 0.50219518117959T_4 + 0.41T_5 = 0 \\
0.50219518117959T_4 + (-0.09804347826087 - 0.43673913043478i)T_5 \\
\qquad -0.77551855666366T_6 - 1.2 = 0
\end{array} \right. \tag{58}
$$

where $\mathbf{t} = (t_1, t_2, \ldots, t_6)$ and $\mathbf{T} = (T_1, T_2, \ldots, T_6)$. The solutions to this system are a curve of degree six and six isolated solutions. Use `phc -c` to construct $\mathcal{E}_1(f)$ and call `phc -c` again to run the cascade. Check the output to verify whether a witness set for the curve has six points and check the size of the witness superset $\widehat{W}_0$. Count how many paths have been tracked. $\diamond$

Bibliographic Notes. The cascade of homotopies was presented in [105]. The complexity of finding all dimensions and degrees of the solutions of a polynomial system was adressed in [42], see also [103]. The cascade in [105] gave the first efficient numerical implementation for this problem.

3.3 Factoring Solution Sets into Irreducible Components

Definition 3.5 For a witness set W_L representing a k-dimensional solution set of $f(\mathbf{x}) = \mathbf{0}$, *the linear trace* T is a linear function of the constant c in the kth hyperplane of L. In particular, the value of $T(c)$ is the sum of all second coordinates the solutions obtained at for the value c of the constant of the kth hyperplane.

Algorithm 3.5 linear trace test
Input: W_L, $W_{L'}$, $W_{L''}$, witness sets on parallel slices;
a partition of the witness set.
Output: yes if every set in the partition is a factor, no otherwise.
Implementation: `phc -f`.

Theorem 3.5 (linear trace criterium) *Consider three hyperplanes L, L', and L'', with respective constants c_0, c_1, and c_2. Linear Interpolation through $(c_0, T(c_0))$ and $(c_1, T(c_1))$ gives constants a and b in the formula $T(c) = a + bc$. If $T(c_2)$ corresponds to the actual value of the linear trace at c_2, then the points used in the linear interpolation represent a factor of the solution set.*

Example 3.5 Take a planar cubic, defined by $f(x, y)$. Consider y as a function of x: $y = y(x)$, then:

$$
\begin{array}{rcl}
f(x, y(x)) & = & (y - y_1(x))(y - y_2(x))(y - y_3(x)) \\
& = & y^3 - t_1(x)y^2 + t_2(x)y - t_3(x)
\end{array} \tag{59}
$$

As $\deg(f) = 3$, the coefficient with y^2 in $f(x, y(x))$ must be linear, it is the linear trace $t_1(x) = c_1x + c_0$. To determine the coefficients c_0 and c_1, we sample the cubic at $x = x_0$ and $x = x_1$. Denote the samples

by $\{(x_0, y_{00}), (x_0, y_{01}), (x_0, y_{02})\}$ and $\{(x_1, y_{10}), (x_1, y_{11}), (x_1, y_{12})\}$. The coefficients c_0 and c_1 are then the solutions of the linear system

$$\begin{cases} y_{00} + y_{01} + y_{02} = c_1 x_0 + c_0 \\ y_{10} + y_{11} + y_{12} = c_1 x_1 + c_0 \end{cases} \tag{60}$$

With t_1 we can predict the sum of the y's for a fixed choice of x. For example, samples at $x = x_2$ are $\{(x_2, y_{20}), (x_2, y_{21}), (x_2, y_{22})\}$. Comparing the predicted value $t_1(x_2)$ with the actual sum is implemented by the equality $c_1 x_2 + c_0 = y_{20} + y_{21} + y_{22}$. If the samples would have been drawn from a quartic instead of a cubic, the the actual sum of the y-values at x_2 would have been different from $t_1(x_2)$ and the test would fail. Figure 6 shows the setup for the linear trace test. ◇

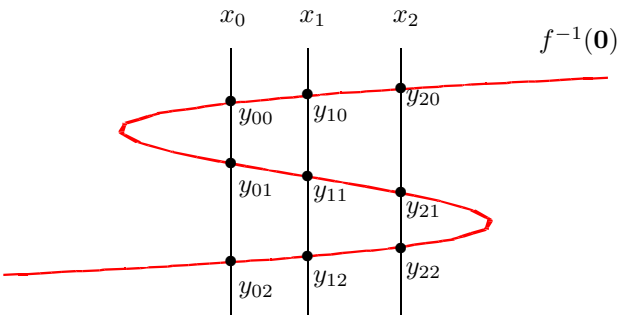


Figure 6: Illustration of the linear trace test on a cubic. We compare the predicted value for $t_1(x_2)$ with the actual sum of the y-values at x_2.

Exercise 3.5 Consider again the system (50) for which you computed a witness set in Exercise 3.1. Give this witness set as input to `phc -f` and compute the number of irreducible factors. Use the combinatorial enumeration of all possible factors, validating each choice with the linear trace. How many irreducible factors does this system have? Try higher dimensional instances of this problem, i.e.: pick a general 2-by-n matrix and factor. For which n becomes the combinatorial method too expensive? ◇

Definition 3.6 Let the witness set W_L represents a k-dimensional solution set of $f(\mathbf{x}) = \mathbf{0}$, cut out by k random hyperplanes L. For k other hyperplanes K, we move W_L to W_K, using the homotopy $h_{L,K,alpha}(\mathbf{x}, t) = \mathbf{0}$, from $t = 0$ to 1:

$$h_{L,K,\alpha}(\mathbf{x}, t) = \begin{pmatrix} f(\mathbf{x}) \\ \alpha(1-t)L(\mathbf{x}) + tK(\mathbf{x}) \end{pmatrix} = \mathbf{0}, \quad \alpha \in \mathbb{C}, \tag{61}$$

where α is chosen at random. To turn back we generate another random constant β, and use

$$h_{K,L,\beta}(\mathbf{x}, t) = \begin{pmatrix} f(\mathbf{x}) \\ \beta(1-t)K(\mathbf{x}) + tL(\mathbf{x}) \end{pmatrix} = \mathbf{0}, \quad \beta \in \mathbb{C}. \tag{62}$$

The homotopies $h_{L,K,\alpha}(\mathbf{x}, t) = \mathbf{0}$ and $h_{K,L,\beta}(\mathbf{x}, t) = \mathbf{0}$ define a *loop in* $\mathbb{C}$ between $t = 0$ and $t = 1$.

Example 3.6 Figure 7 shows the Riemann surface of $z^3 - w = 0$. A witness set consists of three points, obtained by cutting the surface by a line. Moving the line around the singular point $(0, 0)$, we observe that the order of the points on the line gets permuted. Such a permutation can only happen if the corresponding algebraic curve is irreducible. ◇

Algorithm 3.6 monodromy breakup algorithm

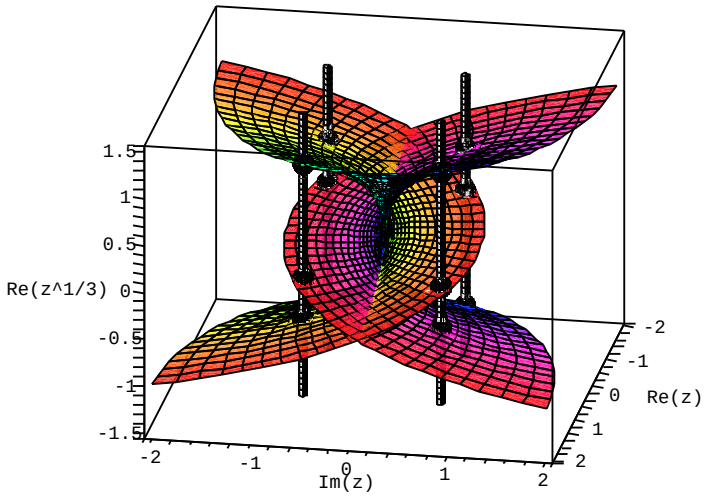


Figure 7: A four dimensional representation of the Riemann surface of $z^3 - w$. The vertical axis shows the real part of the cube root, while the coloring (or gray scale) of the surface encodes the imaginary part $\texttt{Im}(z^{1/3})$ of the cube root. The sheets do not intersect each other!

Input: W_L witness set for a pure dimensional solution set.
Output: $\mathcal{P}$ is a partition of W_L,
every set in $\mathcal{P}$ represents an irreducible factor.
Implementation: `phc -f`.

Theorem 3.6 (monodromy group actions) *While looping around singularities, permutations among generic points can only happen when the generic points belong to the same irreducible factor. Moreover, transitivity between the permutations applies.*

Exercise 3.6 Consider once more the system (50) for which you computed a witness set in Exercise 3.1. Answer the same questions as in Exercise 3.5, but now use the monodromy breakup algorithm. Also try higher dimensional instances of this problem. For which n becomes the monodromy breakup algorithm more efficient than the combinatorial method? ◇

Bibliographic Notes. The factorization over the complex numbers is also known as "Absolute Factorization". See [15] for an excellent treatment of methods for this problem. Finding polynomial time algorithms for factoring multivariate polynomials was listed as an challenging open problem in [57]. Recent papers on this problem are [14], [18, 17], [33], [31, 32], [50], [104], and [115]. The theoretical use of homotopies to study the complexity of this problem was done in [8]. Computer algebra methods to graph Riemann surfaces are described in [19] and [6]. We proposed the use of monodromy in [109] and the validation by the linear trace in [111].

3.4 Diagonal Homotopies to Intersect Pure Dimensional Solution Sets

Definition 3.7 Consider two pure dimensional algebraic sets V_A and V_B in $\mathbb{C}^n$. Denote $\dim(V_A) = a$, $\dim(V_B) = b$, $\deg(V_A) = \alpha$, and $\deg(V_B) = \beta$. For simplicity we consider only complete intersections: V_A is defined by $f_A(\mathbf{x}) = \mathbf{0}$, a system of $n - a$ equations, V_B by $f_B(\mathbf{x}) = \mathbf{0}$, a system of $n - b$ equations, $\mathbf{x} \in \mathbb{C}^n$. Generic points on V_A and V_B satisfy the respective linear systems $L_A(\mathbf{x}) = \mathbf{0}$ and $L_B(\mathbf{x}) = \mathbf{0}$, of respectively a and b linear equations. Summarizing our notation, we have

$$\#\{\ \mathbf{x} \in \mathbb{C}^n \mid f_A(\mathbf{x}) = \mathbf{0}, L_A(\mathbf{x}) = \mathbf{0}\ \} = \alpha \quad \text{and} \quad \#\{\ \mathbf{x} \in \mathbb{C}^n \mid f_B(\mathbf{x}) = \mathbf{0}, L_B(\mathbf{x}) = \mathbf{0}\ \} = \beta. \tag{63}$$

Again for simplicity we assume $\dim(A \cap B) = 0$. To compute $A \cap B$, we define (a very special case of) the *diagonal homotopy*:

$$h(\mathbf{x}, \mathbf{y}, t) = \begin{cases} f_A(\mathbf{x}) = \mathbf{0} \\ f_B(\mathbf{y}) = \mathbf{0} \\ (1-t) \begin{pmatrix} L_A(\mathbf{x}) \\ L_B(\mathbf{y}) \end{pmatrix} + t(\mathbf{x} - \mathbf{y}) = \mathbf{0} \end{cases} \tag{64}$$

which starts at $t = 0$ at the $\alpha \times \beta$ solutions in $A \times B \in \mathbb{C}^{n+n}$. At $t = 1$, we find solutions at the diagonal $\mathbf{x} = \mathbf{y}$, in $A \cap B$.

Example 3.7 This example illustrates why we need new homotopies to intersect witness sets. Stacking two (possibly identical) systems to find $A \cap B$ is not sufficient. For example, suppose A is the line $x_2 = 0$, as solution of $f_A(x_1, x_2) = x_1 x_2 = 0$ and B is the line $x_1 - x_2 = 0$, as solution of $f_B(x_1, x_2) = x_1(x_1 - x_2) = 0$. The problem is that $A \cap B = \{(0, 0)\}$ does not occur as an irreducible solution component of the system

$$\begin{cases} f_A(x_1, x_2) = x_1 x_2 = 0 \\ f_B(x_1, x_2) = x_1(x_1 - x_2) = 0 \end{cases} \tag{65}$$

as the origin $(0, 0)$ is a nonisolated solution on the solution line $x_1 = 0$ of the system obtained by stacking f_A and f_B. ◇

Algorithm 3.7 diagonal homotopy
Input: witness sets W_A and W_B representing sets A and B.
Output: $[W_0, W_1, \ldots, W_k]$ superwitness sets representing $A \cap B$.
Implementation: `phc -w`.

Theorem 3.7 (witness set intersection) *Every isolated solution of $A \cap B$ lies at the end of a path defined by the diagonal homotopy.*

Exercise 3.7 Consider again the system (5) first introduced in Example 2.3. To compute all exceptional values for the parameter a we could look for those vectors for which the Jacobian matrix of the system becomes singular. In particular, consider the system

$$\begin{cases} \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ c_0 + c_1\lambda_1 + c_2\lambda_2 + c_3\lambda_3 = 0 \end{cases} \tag{66}$$

where the coefficients c_0, c_1, c_2, and c_2 are randomly chosen complex numbers. Compute witness sets for the systems (5) and (66) and take the product of the degrees. Compare this product with the number of solutions from the system obtained just by stacking (5) and (66) together and solving it by `phc -b`.

Collect the values for a and verify what happens in all these exceptional instances of the parameter a. For isolated singularities, the "verify" means to compute the multiplicity structure after deflation. Positive dimensional solution sets are "verified" by a witness set, eventually partitioned into sets corresponding to irreducible components. ◇

Definition 3.8 Let $f(\mathbf{x}) = \mathbf{0}$ be a polynomial system of N equations. Let k be a natural number, $0 < k < N$. A *witness stone* is a superwitness set for the solution set defined by the first k equations of $f(\mathbf{x}) = \mathbf{0}$.

Example 3.8 We consider again the system (56), from Example 3.4. Figure 8 shows the flow of the equation-by-equation solver. The total number of paths to be tracked is 24, ten paths diverge to ∞. ◇

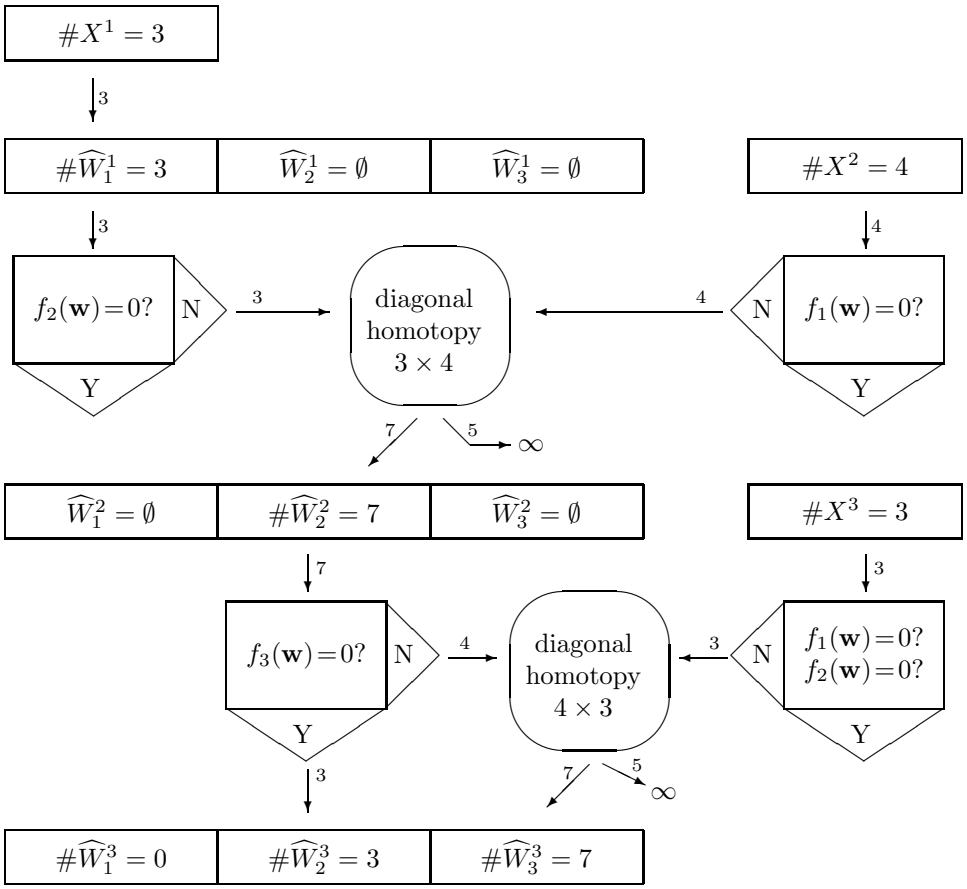


Figure 8: Flow of the equation-by-equation solver on system (56).

Algorithm 3.8 equation-by-equation solver
Input: $f(\mathbf{x}) = \mathbf{0}$, a polynomial system.
Output: $[W_0, W_1, \ldots, W_{n-1}]$ superwitness sets to represent $f^{-1}(\mathbf{0})$.
Implementation: `phc -a`.

Theorem 3.8 (equation-by-equation solver) *The equation-by-equation solver finds superwitness sets for all solution sets of all dimensions defined by a polynomial system.*

Exercise 3.8 Consider again the system (58), solved in Exercise 3.4. Run `phc -a` on this system and experiment with different orders of the equations. What is the best order of the equations? ◇

Bibliographic Notes. The diagonal homotopies were presented in [114]. In [116] we developed a version in intrinsic coordinates. The equation-by-equation solver was described in [106]. See [117] for another survey on numerical algebraic geometry. Using lifting fibers, an equation-by-equation solver was presented in [44] and [64].

4 Lecture III: Software and Applications

This last lecture is not really a lecture, but an invitation to a hands-on session, to use `phc` to solve the exercises of Lectures I and II. The sections below describe some essential information about PHCpack, such as input-output formats and modes of operation.

4.1 Examples, Applications, Benchmarks

Polynomial systems occur in a wide variety of applications in science and engineering. A system that was once an application become an example to illustrate certain tools and techniques or a benchmark to compare the performance. The format in which `phc` accepts polynomial systems on input is as follows:

```
3  8
  x11*x22 - x21*x12;
  x12*x23 - x22*x13;
  x13*x24 - x23*x14;
```

The first line contains respectively the number of equations and the number of unknowns. If the second number is omitted, then `phc` assumes the system has as many unknowns as equations. The polynomials are read in symbolic form, and separated by semicolons.

The `i` and `I` represent the imaginary unit $\sqrt{-1}$ and should not be used as names of variables. Similarly, the `e` and `E` are used to denote floating-point numbers in scientific notation and should also not be used as symbols for variables in the polynomials.

Because any polynomial system can serve as a start system, solutions may be added to the input file. `phc` will recognize a list of solutions if they are in the format

```
THE SOLUTIONS :
1 2
============================================================
solution 1 :
t :  1.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.23606797749979E+00   0.00000000000000E+00
 y :  0.00000000000000E+00  -1.27201964951407E+00
== err :  1.174E-15 = rco :  1.079E-01 = res :  4.441E-16 ==
```

The banner `THE SOLUTIONS :` is followed by the number of solutions in the list and the dimension of the solution vectors. All complex values are listed as a sequence of real and imaginary numbers. The first value is the value for the continuation parameter `t`, then followed by the multiplicity `m`.

The components of the solution vector occur in the order in which `phc` read the polynomials. If `y` was read before `x`, then in the solution vectors, values for `y` will come before the values for `x`. One could impose

on `phc` an order of the symbols, for example: `x` before `y`, by giving the null polynomial `x + y - x - y` as the first part of the input system.

The `err`, `rco`, and `res` report information from Newton's method: `err` is the magnitude of the correction term $|\Delta \mathbf{x}|$, `rco` is an estimate for the inverse of the condition number, and `res` is the magnitude of the residual vector $|f(\mathbf{x})|$.

4.2 Using phc in blackbox or toolbox mode

`PHC` stands for Polynomial Homotopy Continuation in the software package `PHCpack`. We write `PHCpack` when we intend the whole package, i.e.: source code, examples, makefiles, executables, etc. Most users never compile the code, but use the program in executable form (either on Linux, Windows, MacOS X, or Sun), and then we refer to the program as `phc` as this is the recommended name to save the program. Windows users should know that the program runs from the command line, so one should open a DOS command prompt window first to use the program.

PHCpack offers two blackbox solvers. The original one, invoked as `phc -b` was already operational in the first public release of the source code [129]. This blackbox solver is primarily aimed at approximating all isolated solutions of a polynomial system. A new blackbox facility is `phc -a`, which provides a first implementation of the new equation-by-equation solver. In principle, this option computes superwitness sets for all solutions, of all dimensions, but currently, it is not yet as stable as `phc -b`. Calling `phc` without any option shows an overview of all available options and runs the program in full interactive mode.

4.3 PHCmaple: A Maple Interface to PHCpack

PHCmaple is a Maple module which provides an interface to the executable `phc`, presented in [65]. The use of a computer algebra system like Maple to formulate and document polynomial systems is quite natural. PHCmaple facilitates the use of `phc`. Currently, PHCmaple does not yet support the complete functionality of `phc` and for this tutorial we will not use PHCmaple.

4.4 Parallel PHCpack calls phc from C programs

Path following methods enjoy the advantage that they scale very well on multiprocessor machines. In [137] and [66], the path trackers in PHCpack were called from C programs using MPI on clusters. In addition to the parallel aspects of the code, the interface to PHCpack is a low-level alternative to PHCmaple. As this interface turns PHCpack into a state machine, the calling program does not need to define any data structures (for polynomial systems, lists of solutions, etc.) which are already provided by PHCpack.

References

[1] E.L. Allgower, K. Böhmer, A. Hoy, and V. Janovský. Direct methods for solving singular nonlinear equations. *ZAMM Z. Angew. Math. Meth.*, 79(4):219–231, 1999.

[2] E.L. Allgower and K. Georg. *Numerical Continuation Methods, an Introduction*, volume 13 of *Springer Ser. in Comput. Math.* Springer–Verlag, 1990.

[3] E.L. Allgower and K. Georg. Continuation and path following. *Acta Numerica*, pages 1–64, 1993.

[4] E.L. Allgower and K Georg. Numerical Path Following. In P.G. Ciarlet and J.L. Lions, editors, *Techniques of Scientific Computing (Part 2)*, volume 5 of *Handbook of Numerical Analysis*, pages 3–203. North-Holland, 1997.

[5] E.L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*, volume 45 of *Classics in Applied Mathematics*. SIAM, 2003.

[6] D.A. Aruliah and R.M. Corless. Numerical parametrization of affine varieties using ODEs. In J. Gutierrez, editor, *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004)*, pages 12–18. ACM, 2004.

[7] J. Backelin and R. Fröberg. How we proved that there are exactly 924 cyclic 7-roots. In *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, pages 101–111. ACM, 1991.

[8] C. Bajaj, J. Canny, T. Garrity, and J. Warren. Factoring rational polynomials over the complex numbers. *SIAM J. Comput.*, 22(2):318–331, 1993.

[9] D.N. Bernshteĭn. The number of roots of a system of equations. *Functional Anal. Appl.*, 9(3):183–185, 1975. Translated from *Funktsional. Anal. i Prilozhen.*, 9(3):1–4,1975.

[10] G. Björk and R. Fröberg. A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots. *J. Symbolic Computation*, 12(3):329–336, 1991.

[11] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer–Verlag, 1998.

[12] O. Bottema and B. Roth. *Theoretical Kinematics*. North-Holland, Amsterdam, 1979.

[13] J. Canny and J.M. Rojas. An optimal condition for determining the exact number of roots of a polynomial system. In S.M. Watt, editor, *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation (ISSAC 1991)*, pages 96–101. ACM, 1991.

[14] G. Chèze. Absolute polynomial factorization in two variables and the knapsack problem. In J. Gutierrez, editor, *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004)*, pages 87–94. ACM, 2004.

[15] G. Chèze and A. Galligo. Four lectures on polynomial absolute factorization. In *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 339–394. Springer–Verlag, 2005.

[16] S.N. Chow, J. Mallet-Paret, and J.A. Yorke. Homotopy method for locating all zeros of a system of polynomials. In H.O. Peitgen and H.O. Walther, editors, *Functional differential equations and approximation of fixed points*, volume 730 of *Lecture Notes in Mathematics*, pages 77–88. Springer–Verlag, 1979.

[17] R.M. Corless, A. Galligo, I.S. Kotsireas, and S.M. Watt. A geometric-numeric algorithm for factoring multivariate polynomials. In T. Mora, editor, *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation (ISSAC 2002)*, pages 37–45. ACM, 2002.

[18] R.M. Corless, M.W. Giesbrecht, M. van Hoeij, I.S. Kotsireas, and S.M. Watt. Towards factoring bivariate approximate polynomials. In B. Mourrain, editor, *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation (ISSAC 2001)*, pages 85–92. ACM, 2001.

[19] R.M. Corless and D.J. Jeffrey. Graphing elementary Riemann surfaces. *SIGSAM Bulletin*, 32(1):11–17, 1998.

[20] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer-Verlag, 1998.

[21] Y. Dai, S. Kim, and M. Kojima. Computing all nonsingular solutions of cyclic-n polynomial using polyhedral homotopy continuation methods. *J. Comput. Appl. Math.*, 152(1-2):83–97, 2003.

[22] B.H. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In M. Kauers, editor, *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation*, pages 116–123. ACM, 2005.

[23] J.P. Dedieu and M. Shub. Newton's method for overdetermined systems of equations. *Math. Comp.*, 69(231):1099–1115, 1999.

[24] P. Deuflard, B. Friedler, and P. Kunkel. Efficient numerical path folowing beyond critical points. *SIAM J. Numer. Anal.*, 24:912–927, 1987.

[25] P. Diaconis, D. Eisenbud, and B. Sturmfels. Lattice walks and primary decomposition. In B.E. Sagan and R.P. Stanley, editors, *Mathematical Essays in Honor of Gian-Carlo Rota*, volume 161 of *Progress in Mathematics*, pages 173–193. Birkhäuser, 1998.

[26] F.J. Drexler. Eine Methode zur Berechnung sämtlicher Lösungen von Polynomgleichungssystemen. *Numer. Math.*, 29(1):45–58, 1977.

[27] I.Z. Emiris. Toric resultants and applications to geometric modelling. In *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 269–300. Springer–Verlag, 2005.

[28] I.Z. Emiris and J.F. Canny. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symbolic Computation*, 20(2):117–149, 1995.

[29] I.Z. Emiris and B. Mourrain. Computer algebra methods for studying and computing molecular conformations. *Algorithmica*, 25:372–402, 1999.

[30] I.Z. Emiris and J. Verschelde. How to count efficiently all affine roots of a polynomial system. *Discrete Applied Mathematics*, 93(1):21–32, 1999.

[31] A. Galligo and D. Rupprecht. Semi-numerical determination of irreducible branches of a reduced space curve. In B. Mourrain, editor, *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation (ISSAC 2001)*, pages 137–142. ACM, 2001.

[32] A. Galligo and D. Rupprecht. Irreducible decomposition of curves. *J. Symbolic Computation*, 33(5):661–677, 2002.

[33] S. Gao, E Kaltofen, J May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In J. Gutierrez, editor, *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004)*, pages 167–174. ACM, 2004.

[34] T. Gao and T.Y. Li. Mixed volume computation via linear programming. *Taiwan J. of Math.*, 4:599–619, 2000.

[35] T. Gao and T.Y. Li. Mixed volume computation for semi-mixed systems. *Discrete Comput. Geom.*, 29(2):257–277, 2003.

[36] T. Gao, T.Y. Li, J. Verschelde, and M. Wu. Balancing the lifting values to improve the numerical stability of polyhedral homotopy continuation methods. *Appl. Math. Comput.*, 114:233–247, 2000.

[37] T. Gao, T.Y. Li, and X. Wang. Finding isolated zeros of polynomial systems in C^n with stable mixed volumes. *J. of Symbolic Computation*, 28(1-2):187–211, 1999.

[38] T. Gao, T.Y. Li, and M. Wu. Algorithm 846: MixedVol: a software package for mixed-volume computation. *ACM Trans. Math. Softw.*, 31(4):555–560, 2005.

[39] C.B. Garcia and T.Y. Li. On the number of solutions to polynomial systems of equations. *SIAM J. Numer. Anal.*, 17(4):540–546, 1980.

[40] C.B. Garcia and W.I. Zangwill. Finding all solutions to polynomial systems and other systems of equations. *Math. Programming*, 16(2):159–176, 1979.

[41] I.M. Gel'fand, M.M. Kapranov, and A.V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants*. Birkhäuser, 1994.

[42] M. Giusti and J. Heintz. La détermination de la dimension et des points isolées d'une variété algébrique peuvent s'effectuer en temps polynomial. In D. Eisenbud and L. Robbiano, editors, *Computational Algebraic Geometry and Commutative Algebra, Cortona 1991*, volume XXXIV of *Symposia Mathematica*, pages 216–256. Cambridge University Press, 1993.

[43] M. Giusti and J. Heinz. Kronecker's smart, little black boxes. In DeVore, R.A. and Iserles, A. and Süli, E., editor, *Foundations of Computational Mathematics*, volume 284 of *London Mathematical Society Lecture Note Series*, pages 69–104. Cambridge University Press, 2001.

[44] M. Giusti, G. Lecerf, and B. Salvy. A gröbner free alternative for polynomial system solving. *J. Complexity*, 17(1):154–211, 2001.

[45] W.J.F. Govaerts. *Numerical Methods for Bifurcations of Dynamical Equilibria*. SIAM, 2000.

[46] G.M. Greuel and G. Pfister. Advances and improvements in the theory of standard bases and syzygies. *Arch. Math.*, 66:163–196, 1996.

[47] G.M. Greuel and G. Pfister. *A Singular Introduction to Commutative Algebra*. Springer-Verlag, 2002.

[48] A. Griewank. On solving nonlinear equations with simple singularities or nearly singular solutions. *SIAM Review*, 27(4):537–563, 1985.

[49] S. Hoşten and J. Shapiro. Primary decomposition of lattice basis ideals. *Journal of Symbolic Computation*, 29(4&5):625–639, 2000.

[50] Y. Huang, W. Wu, H.J. Stetter, and L. Zhi. Pseudofactors of multivariate polynomials. In C. Traverso, editor, *Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation (ISSAC 2000)*, pages 161–168. ACM, 2000.

[51] B. Huber, J. Rambau, and F. Santos. The Cayley trick, lifting subdivisions and the Bohne-Dress theorem on zonotopal tilings. *J. Eur. Math. Soc.*, 2(2):179–198, 2000.

[52] B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comp.*, 64(212):1541–1555, 1995.

[53] B. Huber and B. Sturmfels. Bernstein's theorem in affine space. *Discrete Comput. Geom.*, 17(2):137–141, 1997.

[54] B. Huber and J. Verschelde. Polyhedral end games for polynomial continuation. *Numerical Algorithms*, 18(1):91–108, 1998.

[55] I. Itenberg and E. Shustin. Viro theorem and topology of real and complex combinatorial hypersurfaces. *Israel Math. J.*, 133:189–238, 2003.

[56] I. Itenberg and O. Viro. Patchworking algebraic curves disproves the ragsdale conjecture. *The Mathematical Intelligencer*, 18(4):19–28, 1996.

[57] E. Kaltofen. Challenges of symbolic computation: my favorite open problems. *J. Symbolic Computation*, 29(6):891–919, 2000.

[58] A.G. Khovanskiĭ. Newton polyhedra and the genus of complete intersections. *Functional Anal. Appl.*, 12(1):38–46, 1978. Translated from *Funktsional. Anal. i Prilozhen.* 12(1),51–61,1978.

[59] S. Kim and M. Kojima. Numerical stability of path tracing in polyhedral homotopy continuation methods. *Computing*, 73(4):329–348, 2004.

[60] P. Kunkel. Efficient computation of singular points. *IMA J. Numer. Anal.*, 9:421–433, 1989.

[61] P. Kunkel. A tree-based analysis of a family of augmented systems for the computation of singular points. *IMA J. Numer. Anal.*, 16:501–527, 1996.

[62] A.G. Kushnirenko. Newton Polytopes and the Bézout Theorem. *Functional Anal. Appl.*, 10(3):233–235, 1976. Translated from *Funktsional. Anal. i Prilozhen.* 10(3),82–83,1976.

[63] G. Lecerf. Quadratic Newton iteration for systems with multiplicity. *Found. Comput. Math.*, 2:247–293, 2002.

[64] G. Lecerf. Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers. *J. Complexity*, 19(4):564–596, 2003.

[65] A. Leykin and J. Verschelde. PHCmaple: A Maple interface to the numerical homotopy algorithms in PHCpack. In Quoc-Nam Tran, editor, *Proceedings of the Tenth International Conference on Applications of Computer Algebra (ACA'2004)*, pages 139–147, 2004.

[66] A. Leykin and J. Verschelde. Factoring solution sets of polynomial systems in parallel. In Tor Skeie and Chu-Sing Yang, editors, *Proceedings of the 2005 International Conference on Parallel Processing Workshops. 14-17 June 2005. Oslo, Norway. High Performance Scientific and Engineering Computing*, pages 173–180. IEEE Computer Society, 2005.

[67] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theoretical Computer Science*, 359(1-3):111–122, 2006.

[68] T.Y. Li. On Chow, Mallet-Paret and Yorke homotopy for solving systems of polynomials. *Bulletin of the Institute of Mathematics. Acad. Sin.*, 11:433–437, 1983.

[69] T.Y. Li. Solving polynomial systems. *The Mathematical Intelligencer*, 9(3):33–39, 1987.

[70] T.Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In F. Cucker, editor, *Handbook of Numerical Analysis. Volume XI. Special Volume: Foundations of Computational Mathematics*, pages 209–304. North-Holland, 2003.

[71] T.Y. Li and X. Li. Finding mixed cells in the mixed volume computation. *Found. Comput. Math.*, 1(2):161–181, 2001. Software available at `http://www.math.msu.edu/~li`.

[72] T.Y. Li and T. Sauer. Regularity results for solving systems of polynomials by homotopy method. *Numer. Math.*, 50(3):283–289, 1987.

[73] T.Y. Li, T. Sauer, and J.A. Yorke. Numerical solution of a class of deficient polynomial systems. *SIAM J. Numer. Anal.*, 24(2):435–451, 1987.

[74] T.Y. Li, T. Sauer, and J.A. Yorke. The random product homotopy and deficient polynomial systems. *Numer. Math.*, 51(5):481–500, 1987.

[75] T.Y. Li, T. Sauer, and J.A. Yorke. The cheater's homotopy: an efficient procedure for solving systems of polynomial equations. *SIAM J. Numer. Anal.*, 26(5):1241–1251, 1989.

[76] T.Y. Li, T. Wang, and X. Wang. Random product homotopy with minimal BKK bound. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 503–512. AMS, 1996. Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics. Park City, Utah, July 17-August 11, 1995, Park City, Utah.

[77] T.Y. Li and X. Wang. Solving deficient polynomial systems with homotopies which keep the subschemes at infinity invariant. *Math. Comp.*, 56(194):693–710, 1991.

[78] T.Y. Li and X. Wang. Nonlinear homotopies for solving deficient polynomial systems with parameters. *SIAM J. Numer. Anal.*, 29(4):1104–1118, 1992.

[79] T.Y. Li and X. Wang. Solving real polynomial systems with real homotopies. *Math. Comp.*, 60:669–680, 1993.

[80] T.Y. Li and X. Wang. Higher order turning points. *Appl. Math. Comput.*, 64:155–166, 1994.

[81] T.Y. Li and X. Wang. The BKK root count in C^n. *Math. Comp.*, 65(216):1477–1484, 1996.

[82] F. Mora. An algorithm to compute the equations of tangent cones. In J. Calmet, editor, *Computer Algebra. EUROCAM'82, European Computer Algebra Conference. Marseille, France, April 1982.*, volume 144 of *Lecture Notes in Computer Science*, pages 158–165. Springer-Verlag, 1982.

[83] A. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems.* Prentice-Hall, 1987.

[84] A. Morgan and V. Shapiro. Box-bisection for solving second-degree systems and the problem of clustering. *ACM Trans. Math. Soft.*, 13(2):152–167, 1987.

[85] A. Morgan and A. Sommese. Computing all solutions to polynomial systems using homotopy continuation. *Appl. Math. Comput.*, 24(2):115–138, 1987. Errata: *Appl. Math. Comput.* 51 (1992), p. 209.

[86] A. Morgan and A. Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Appl. Math. Comput.*, 24(2):101–113, 1987.

[87] A.P. Morgan. A method for computing all solutions to systems of polynomial equations. *ACM Trans. Math. Softw.*, 9(1):1–17, 1983.

[88] A.P. Morgan and A.J. Sommese. Coefficient-parameter polynomial continuation. *Appl. Math. Comput.*, 29(2):123–160, 1989. Errata: *Appl. Math. Comput.* 51:207(1992).

[89] A.P. Morgan and A.J. Sommese. Generically nonsingular polynomial continuation. In E.L. Allgower and K. Georg, editors, *Computational Solution of Nonlinear Systems of Equations*, pages 467–493. AMS, 1990.

[90] A.P. Morgan, A.J. Sommese, and C.W. Wampler. Computing singular solutions to nonlinear analytic systems. *Numer. Math.*, 58(7):669–684, 1991.

[91] A.P. Morgan, A.J. Sommese, and C.W. Wampler. Computing singular solutions to polynomial systems. *Adv. Appl. Math.*, 13(3):305–327, 1992.

[92] A.P. Morgan, A.J. Sommese, and C.W. Wampler. A power series method for computing singular solutions to nonlinear analytic systems. *Numer. Math.*, 63:391–409, 1992.

[93] A.P. Morgan, A.J. Sommese, and C.W. Wampler. A product-decomposition theorem for bounding Bézout numbers. *SIAM J. Numer. Anal.*, 32(4):1308–1325, 1995.

[94] B. Mourrain. Isolated points, duality and residues. *J. of Pure and Applied Algebra*, 117 & 118:469–493, 1996.

[95] 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.

[96] T. Ojika. Modified deflation algorithm for the solution of singular problems. II. Nonlinear multipoint boundary value problems. *J. Math. Anal. Appl.*, 123:222–237, 1987.

[97] T. Ojika. A numerical method for branch points of a system of nonlinear algebraic equations. *Applied Numerical Mathematics*, 4:419–430, 1988.

[98] 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.

[99] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*, volume 30 of *Classics in Applied Mathematics*. SIAM, 2000.

[100] J.M. Rojas. A convex geometric approach to counting the roots of a polynomial system. *Theoret. Comput. Sci.*, 133(1):105–140, 1994.

[101] J.M. Rojas. Toric intersection theory for affine root counting. *Journal of Pure and Applied Algebra*, 136(1):67–100, 1999.

[102] J.M. Rojas and X. Wang. Counting affine roots of polynomial systems via pointed Newton polytopes. *J. Complexity*, 12:116–133, 1996.

[103] J. Sabia. Algorithms and their complexities. In *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 241–268. Springer–Verlag, 2005.

[104] T. Sasaki. Approximate multivariate polynomial factorization based on zero-sum relations. In B. Mourrain, editor, *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation (ISSAC 2001)*, pages 284–291. ACM, 2001.

[105] A.J. Sommese and J. Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *J. of Complexity*, 16(3):572–602, 2000.

[106] A.J. Sommese, J. Verschelde, and C.W. Wampler. Solving polynomial systems equation by equation. Submitted for publication.

[107] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM J. Numer. Anal.*, 38(6):2022–2046, 2001.

[108] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using projections from points on the components. In E.L. Green, S. Hoşten, R.C. Laubenbacher, and V. Powers, editors, *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*, pages 37–51. AMS, 2001.

[109] A.J. Sommese, J. Verschelde, and C.W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In C. Ciliberto, F. Hirzebruch, R. Miranda, and M. Teicher, editors, *Application of Algebraic Geometry to Coding Theory, Physics and Computation*, pages 297–315. Kluwer Academic Publishers, 2001. Proceedings of a NATO Conference, February 25 - March 1, 2001, Eilat, Israel.

[110] A.J. Sommese, J. Verschelde, and C.W. Wampler. A method for tracking singular paths with application to the numerical irreducible decomposition. In M.C. Beltrametti, F. Catanese, C. Ciliberto, A. Lanteri, and C. Pedrini, editors, *Algebraic Geometry, a Volume in Memory of Paolo Francia*, pages 329–345. Walter de Gruyter, 2002.

[111] A.J. Sommese, J. Verschelde, and C.W. Wampler. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM J. Numer. Anal.*, 40(6):2026–2046, 2002.

[112] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using PHCpack. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 109–130. Springer–Verlag, 2003.

[113] A.J. Sommese, J. Verschelde, and C.W. Wampler. Advances in polynomial continuation for solving problems in kinematics. *ASME J. of Mechanical Design*, 126(2):262–268, 2004.

[114] A.J. Sommese, J. Verschelde, and C.W. Wampler. Homotopies for intersecting solution components of polynomial systems. *SIAM J. Numer. Anal.*, 42(4):552–1571, 2004.

[115] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical factorization of multivariate complex polynomials. *Theoretical Computer Science*, 315(2-3):651–669, 2004. Special Issue on Algebraic and Numerical Algorithms edited by I.Z. Emiris, B. Mourrain, and V.Y. Pan.

[116] A.J. Sommese, J. Verschelde, and C.W. Wampler. An intrinsic homotopy for intersecting algebraic varieties. *J. of Complexity*, 21(4):593–608, 2005.

[117] A.J. Sommese, J. Verschelde, and C.W. Wampler. Introduction to numerical algebraic geometry. In *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 301–337. Springer–Verlag, 2005.

[118] A.J. Sommese and C.W. Wampler. Numerical algebraic geometry. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 749–763. AMS, 1996. Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics. Park City, Utah, July 17-August 11, 1995, Park City, Utah.

[119] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific Press, Singapore, 2005.

[120] M. Sosonkina, L.T. Watson, and D.E. Stewart. Note on the end game in homotopy zero curve tracking. *ACM Trans. Math. Softw.*, 22(3):281–287, 1996.

[121] H.J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.

[122] H.J. Stetter and G.T. Thallinger. Singular systems of polynomials. In O. Gloor, editor, *Proceedings of ISSAC 1998*, pages 9–16. ACM, 1998.

[123] B. Sturmfels. On the Newton polytope of the resultant. *Journal of Algebraic Combinatorics*, 3:207–236, 1994.

[124] B. Sturmfels. On the number of real roots of a sparse polynomial system. In A. Bloch, editor, *Hamiltonian and Gradient Flows: Algorithms and Control*, pages 137–143. AMS, 1994.

[125] B. Sturmfels. Viro's theorem for complete intersections. *Annali della Scuola Normale Superiore di Pisa*, 21(3):377–386, 1994.

[126] B. Sturmfels. Polynomial equations and convex polytopes. *Amer. Math. Monthly*, 105(10):907–922, 1998.

[127] B. Sturmfels. *Solving Systems of Polynomial Equations*. Number 97 in CBMS Regional Conference Series in Mathematics. AMS, 2002.

[128] A. Takeda, M. Kojima, and K. Fujisawa. Enumeration of all solutions of a combinatorial linear inequality system arising from the polyhedral homotopy continuation method. *Journal of the Operations Research Society of Japan*, 45(1):64–82, 2002.

[129] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, 1999. Software available at `http://www.math.uic.edu/~jan`.

[130] J. Verschelde. Toric newton method for polynomial homotopies. *J. Symbolic Computation*, 29(4–5):777–793, 2000.

[131] J. Verschelde and R. Cools. Symbolic homotopy construction. *Applicable Algebra in Engineering, Communication and Computing*, 4(3):169–183, 1993.

[132] J. Verschelde and R. Cools. Symmetric homotopy construction. *J. Comput. Appl. Math.*, 50:575–592, 1994.

[133] J. Verschelde and K. Gatermann. Symmetric Newton polytopes for solving sparse polynomial systems. *Adv. Appl. Math.*, 16(1):95–127, 1995.

[134] J. Verschelde, K. Gatermann, and R. Cools. Mixed-volume computation by dynamic lifting applied to polynomial system solving. *Discrete Comput. Geom.*, 16(1):69–112, 1996.

[135] J. Verschelde and A. Haegemans. The *GBQ*-Algorithm for constructing start systems of homotopies for polynomial systems. *SIAM J. Numer. Anal.*, 30(2):583–594, 1993.

[136] J. Verschelde, P. Verlinden, and R. Cools. Homotopies exploiting Newton polytopes for solving sparse polynomial systems. *SIAM J. Numer. Anal.*, 31(3):915–930, 1994.

[137] J. Verschelde and Y. Wang. Computing feedback laws for linear systems with a parallel pieri homotopy. In Yuanyuan Yang, editor, *Proceedings of the 2004 International Conference on Parallel Processing Workshops. 15-18 August 2004. Montreal, Quebec, Canada. High Performance Scientific and Engineering Computing*, pages 222–229. IEEE Computer Society, 2004.

[138] C.W. Wampler. Bezout number calculations for multi-homogeneous polynomial systems. *Appl. Math. Comput.*, 51(2–3):143–157, 1992.

[139] C.W. Wampler, A.P. Morgan, and A.J. Sommese. Numerical continuation methods for solving polynomial systems arising in kinematics. *ASME J. of Mechanical Design*, 112(1):59–68, 1990.

[140] 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. of Mechanical Design*, 114(1):153–159, 1992.

[141] L.T. Watson. Numerical linear algebra aspects of globally convergent homotopy methods. *SIAM Rev.*, 28(4):529–545, 1986.

[142] L.T. Watson. Globally convergent homotopy methods: a tutorial. *Appl. Math. Comput.*, 31(Spec. Issue):369–396, 1989.

[143] L.T. Watson. Probability-one homotopies in computational science. *J. Comput. Appl. Math.*, 140(1&2):785–807, 2002.

[144] S.M. Wise, A.J. Sommese, and L.T. Watson. Algorithm 801: POLSYS_PLP: a partitioned linear product homotopy code for solving polynomial systems of equations. *ACM Trans. Math. Softw.*, 26(1):176–200, 2000.

[145] A.H. Wright. Finding all solutions to a system of polynomial equations. *Math. Comp.*, 44(169):125–133, 1985.

[146] G.M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer–Verlag, New York, 1995.

[147] W. Zulehner. A simple homotopy method for determining all isolated solutions to polynomial systems. *Math. Comp.*, 50(181):167–177, 1988.

Appendices: Lists of Definitions, Algorithms, Theorems, Examples, and Exercises

The writing of the tutorial started with identifying 22 algorithms, first described by their input-output specification, then preceded with more precise definitions and followed by theoretical statements. As examples and exercises were added to the 22 algorithms, each algorithm is now described by five items. Each item is listed below with a one line title.

The number "22" is rather arbitrary, more limited to the current capabilities of `phc`. Identifying (and implementing (!)) "the" algorithms in numerical algebraic geometry is a long-term research project. This tutorial is just a snapshot to give a taste of what is currently possible.

Another excellent way to walk through the tutorial is to solve the polynomial systems. The lists below can serve as an index to this tutorial.

Appendix A. List of Definitions

Appendix B. List of Algorithms

Appendix C. List of Theorems

Appendix D. List of Examples

Appendix E. List of Exercises

Appendix F. List of Polynomial Systems