# Polyhedral End Games for Polynomial Continuation*

Birkett Huber[†] Jan Verschelde[‡]

March 5, 1998

**Abstract**

Bernshteĭn's theorem provides a generically exact upper bound on the number of isolated solutions a sparse polynomial system can have in $(\mathbb{C}^*)^n$, with $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$. When a sparse polynomial system has fewer than this number of isolated solutions some face system must have solutions in $(\mathbb{C}^*)^n$. In this paper we address the process of recovering a certificate of deficiency from a diverging solution path. This certificate takes the form of a face system along with approximations of its solutions. We apply extrapolation to estimate the cycle number and the face normal. Applications illustrate the practical usefulness of our approach.

**keywords :** homotopy continuation, polynomial systems, Newton polytopes, Bernshteĭn bound, cycle number.

**AMS(MOS) Classification :** 14Q99, 52A39, 52B20, 65H10.

## 1 Introduction

All isolated complex solutions to polynomial systems can be approximated numerically by homotopy continuation methods. The strategy is to set up a collection of implicitly defined paths which can then be traced to the solutions of the original system. We refer to [13] for an introduction and to [11] for a survey on recent research developments.

The key problem is that, while things can be arranged so that every isolated solution lies on the end of some path, there may be many paths which do not lead to finite solutions. In fact, these extra paths are often especially difficult to follow. In $\mathbb{C}^n$ they are divergent in the sense that some coordinates will become arbitrarily large, leaving us with the problem of deciding if a path is indeed diverging or if it is just converging to a solution with large coordinates. Even if one uses some compactifying technique, e.g. following the paths in a randomly chosen affine chart, one still runs into the problem that, in practice, the solutions at infinity will often be singular, which makes it hard to decide whether the solution really lies at infinity.

*Work benefited from U.S. National Science Foundation, Grant No. DMS-9508742 and Fund for Scientific Research - Flanders (Belgium) Grant No. G.0261.96. The first author also received partial support from the David Sarnoff Research Center. The work was conducted when the second author was post-doctor at the K.U.Leuven, Belgium.

†Texas A&M University, Department of Mathematics, College Station, Texas 77843, USA. E-mail: huber@math.tamu.edu

‡Michigan State University, Department of Mathematics, East Lansing, MI 48824-1027, USA. E-mail: jan@math.msu.edu or jan.verschelde@na-net.ornl.gov

To improve the accuracy of approximations to singular solutions, Morgan, Sommese and Wampler developed several *end games*, modifications to the continuation process designed to cope with the numerical difficulties encountered as a path approaches a singular endpoint, see for instance [15], [16] and [17]. In [21], the same problem was addressed by Sosonkina et al. These methods themselves do not distinguish between solutions at infinity and finite solutions: to decide if a path leads to a solution at infinity one must estimate it to high accuracy and see if it lies on the hyperplane at infinity. Before applying special methods to improve an approximation, we first want to find out whether or not a path failure is associated to a solution at infinity, because we expect only finite singular solutions to be important for applications.

This paper is set in the context of polyhedral homotopy methods, based on Bernshteĭn's theorem [1], which bounds the number of isolated solutions a system can have in $(\mathbb{C}^*)^n$. An elaboration of these methods can be found in [9], [25] and [26]. The extension to affine space, i.e.: for $\mathbb{C}^n$, is presented in [10] and [12]. Bernshteĭn's bound counts roots in a toric compactification of $(\mathbb{C}^*)^n$. This compactification is constructed from combinatorial data associated to the monomials which appear. Toric compactifications are explained in [5] and [6]. While Bernshteĭn's bound is exact for generic problems, practical applications are seldom generic. Accordingly homotopy methods based on these counts still require the following of some divergent paths. Our technique for determining if a path is diverging is based on the proof of Bernshteĭn's discriminant condition: solution paths which leave $(\mathbb{C}^*)^n$ correspond to solutions of some face system.

By an inexpensive side-calculation we numerically estimate a path direction which, for diverging paths, approximates an outer normal to the face which causes the deficiency. Hereby we have a certificate that a path really diverges. Identifying the deficient face in a family of systems can lead to the development of more performant homotopies. While these degeneracies can be detected by algebraic techniques, e.g. applying sparse resultants to degenerate systems [19, 20], our approach requires only numerical data readily available during path continuation. In particular, we need neither polynomial factorization nor any combinatorial information on the Newton polytopes to use our techniques. The relevant faces are found during the process of continuation.

Another contribution of this paper is an extrapolation technique for estimating the so-called cycle number of a path. This number plays a critical role in the end games described by Morgan, Sommese and Wampler in [15], [16] and [17].

The outline of this paper is as follows. In the next section we explain the theoretical background. Our polyhedral end game is elaborated in the third section. The fourth section contains the application of Richardson extrapolation. In the fifth section we present an extrapolation method to estimate the cycle number. Implementation issues are discussed in the sixth section. Before concluding we list some practical examples.

## 2 Face systems and power series

We consider Laurent polynomials $f(\mathbf{x}) \in \mathbb{C}[x_1, x_1^{-1}, x_2, x_2^{-1}, \ldots, x_n, x_n^{-1}]$ denoted by

$$f(\mathbf{x}) = \sum_{\mathbf{q} \in \mathcal{A}} c_{\mathbf{q}} \mathbf{x}^{\mathbf{q}} \quad \text{with} \quad c_{\mathbf{q}} \in \mathbb{C}^* \quad \text{and} \quad \mathbf{x}^{\mathbf{q}} = x_1^{q_1} x_2^{q_2} \cdots x_n^{q_n}. \tag{1}$$

The finite subset $\mathcal{A} \subset \mathbb{Z}^n$ is called the *support* of $f$. The *Newton polytope* of $f$ is the convex hull $Q := \mathrm{conv}(\mathcal{A})$. Let $\langle .,. \rangle$ be the standard inner product.

**Definition 2.1** For a linear functional $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ we define the *face* $\partial_\omega \mathcal{A}$ and the *face polynomial* $\partial_\omega f$ by

$$\partial_\omega \mathcal{A} := \left\{ \ \mathbf{q} \in \mathcal{A} \mid \langle \mathbf{q}, \omega \rangle = \min_{\mathbf{p} \in \mathcal{A}} \langle \mathbf{p}, \omega \rangle \ \right\} \quad \text{and} \quad \partial_\omega f(\mathbf{x}) = \sum_{\mathbf{q} \in \partial_\omega \mathcal{A}} c_\mathbf{q} \mathbf{x}^\mathbf{q}. \tag{2}$$

A polynomial system $F(\mathbf{x}) = \mathbf{0}$ is determined by a tuple $F := (f_1, f_2, \ldots, f_n)$ of $n$ Laurent polynomials. We say that a system $F$ has supports $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ if each $\mathcal{A}_i$ is the support of $f_i$, for $i = 1, 2, \ldots, n$. The mixed volume $\mathcal{M}(Q_1, Q_2, \ldots, Q_n)$ is abbreviated by $\mathcal{M}(\mathcal{A})$. Similarly we set $\partial_\omega(F) := (\partial_\omega f_1, \partial_w f_2, \ldots, \partial_\omega f_n)$. Precise mathematical definitions and interpretations of the mixed volume can be found in [5] or [6].

**Theorem 2.2** ([1], Theorems A&B) *A system $F(\mathbf{x}) = \mathbf{0}$ with supports $\mathcal{A}$ has at most $\mathcal{M}(\mathcal{A})$ isolated solutions in $(\mathbb{C}^*)^n$. Furthermore, when $\mathcal{M}(\mathcal{A}) > 0$, $F$ has fewer than $\mathcal{M}(\mathcal{A})$ isolated solutions if and only if $\partial_\omega F$ has a solution in $(\mathbb{C}^*)^n$ for some linear functional $\omega$.*

In [9], [25] and [26] algorithms are presented for producing a start system $G$ with supports $\mathcal{A}$ along with a complete complement of $\mathcal{M}(\mathcal{A})$ solutions $\mathbf{z}_i$ in $(\mathbb{C}^*)^n$. Using the homotopy $H(\mathbf{x}, t) := (1-t)G(\mathbf{x}) + tF(\mathbf{x})$ we can find all $\mathcal{M}(\mathcal{A})$ solutions of $F(\mathbf{x})$ by using numerical continuation to follow the solution curves of $H(\mathbf{x}, t) = \mathbf{0}$ from $t = 0$ to $t = 1$. As long as $G$ is sufficiently generic, the solution set of $H(\mathbf{x}, t)$ will break up into $\mathcal{M}(\mathcal{A})$ nonsingular curves $\mathbf{x}_i(t)$ such that $\mathbf{x}_i(0) = \mathbf{z}_i$, $H(\mathbf{x}_i(t), t) \equiv \mathbf{0}$, and all solutions of $F(\mathbf{x}) = \mathbf{0}$ can be found among the limit points $\lim_{t \to 1} \mathbf{x}_i(t)$.

In this paper we are interested in the *deficient* case, when $F(\mathbf{x})$ fails to have the full $\mathcal{M}(\mathcal{A})$ number of isolated roots in $(\mathbb{C}^*)^n$. In this case, there will have to be some number of divergent paths, i.e., paths for which $\lim_{t \to 1} \mathbf{x}_i(t) \notin (\mathbb{C}^*)^n$. Our main tool for understanding this situation is the following theorem:

**Theorem 2.3** ([17], [27]) $\forall \mathbf{x}(t)$, $H(\mathbf{x}(t), t) \equiv \mathbf{0}$, $\exists s > 0$, $\exists m \in \mathbb{N} \setminus \{0\}$, $\exists \omega \in \mathbb{Z}^n$, *such that the curve $(\mathbf{x}(t), t)$ can be expressed for $t \approx 1$ by a power series of the form:*

$$\left\{ \begin{array}{rcl} x_i(s) & = & a_i s^{\omega_i} \left( 1 + \displaystyle\sum_{i=1}^{\infty} a_{i,j} s^j \right), \ i = 1, 2, \ldots, n \\ t(s) & = & 1 - s^m. \end{array} \right. \tag{3}$$

For convenience, we refer to $\omega$ as the *direction of the path* or the *path direction.* In [17], the constant $m$ is called the *cycle number*, because $m$ can be computed by integrating a circular sample of solution paths, see also [15]. As mentioned in [15], the cycle number is always equal to or strictly less than the multiplicity.

Note that $H(\mathbf{x}(s), t(s)) = F(\mathbf{x}(s)) + s^m \big( G(\mathbf{x}(s)) - F(\mathbf{x}(s)) \big)$. Thus the lowest-order coefficient of $H(\mathbf{x}(s), t(s))$ (as a Laurent power series in $s$) will be given by the lowest-order terms of $F(\mathbf{x}(s))$ and the system made up of these terms is exactly $\partial_\omega F(\mathbf{x}(s))$. This last statement follows easily by substituting (3) into a monomial $\mathbf{x}^\mathbf{q}$,

$$\mathbf{x}(s)^\mathbf{q} = \left( \prod_{i=1}^{n} a_i^{q_i} \right) s^{\langle \mathbf{q}, \omega \rangle} (1 + O(s)). \tag{4}$$

Since $H(\mathbf{x}(s), t(s)) \equiv \mathbf{0}$, we have shown the following proposition:

**Proposition 2.4** *Let* $\mathbf{x}(s)$ *be a solution curve as in* (3) *and let* $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in (\mathbb{C}^*)^n$ *be the vector of leading coefficients. Then* $\partial_\omega F(\mathbf{a}) = \mathbf{0}$.

Note that when $\lim_{t \to 1} \mathbf{x}(t)$ lies in $(\mathbb{C}^*)^n$ we have $\omega = \mathbf{0}$ and the leading coefficients are actually solutions to $F = \partial_{\mathbf{0}} F$. For a solution path diverging to infinity we have $\omega_i < 0$ for some $i$, whereas $\omega_i > 0$, for a solution with $i$th component equal to zero.

With these preliminaries we can state our goal more precisely: to give numerical techniques to determine if a path is diverging, and if so to identify the face system $\partial_\omega F$ and its solution $\mathbf{a}$.

# 3 The polyhedral end game

In this section we describe the key ideas of the polyhedral end game. The path direction will be approximated by extrapolation on the power series of the logarithms of the moduli of the solutions. The expansion for the error series is the basis for the extrapolation to determine the cycle number. Lastly, we sketch the chart switching, once the face normal is known.

At this point we assume that continuation has reached the point where we have samples of $t_k$ close enough to 1 so that (3) holds. How to decide when to initiate and abandon the end game will be described in section 6.

Our end game consists of two stages. First we estimate the face normal by extrapolation along the solution path. In the second stage we approximate the solution at infinity by numerical continuation after the application of a coordinate transformation.

## 3.1 Approximating the normal by extrapolation

Taking the logarithm of the absolute value of the expression in (3) we derive

$$\log(|x_i(s)|) = \log(|a_i|) + \omega_i \log(s) + \sum_{j=1}^{\infty} c_{i,j} s^j, \tag{5}$$

where $\sum_{j=1}^{\infty} c_{i,j} s^j$ is the Taylor expansion of $\log(1 + \sum_{j=1}^{\infty} a_{i,j} s^j)$ in the power series for $x_i(s)$. During continuation we generate a sequence of points $[\mathbf{x}(s_k), t(s_k)]$, for $k = 0, 1, \ldots, r$, with $s_0 > s_1 > \cdots > s_r > 0$. We can then use these points along with (5) to estimate the $\omega_i$.

As an example consider the most simple case: $m = 1 = r$. Computing differences of consecutive instances of (5) yields

$$\frac{\log(|x_i(s_0)|) - \log(|x_i(s_1)|)}{\log(s_0) - \log(s_1)} = \omega_i + O(s_0). \tag{6}$$

Unfortunately this estimate is really only order $(1 - t(s_0))^{\frac{1}{m}}$ in the observable quantity $t(s_0)$, so that we need higher-order estimates when $m > 1$.

Higher-order estimates for $\omega_i$ can be obtained by taking more than two points at a time along the path. Assume that $m$ is known, then we compute $s_k = (1 - t(s_k))^{\frac{1}{m}}$, for $k = 0, 1, \ldots, r$. Choose multipliers $\lambda_0, \lambda_1, \ldots, \lambda_r$ so that

$$\sum_{k=0}^{r} \lambda_k s_k^j = 0, \quad \text{for } j = 0, 1, \ldots, r-1, \tag{7}$$

is satisfied. Since $s_0 > s_1 > \cdots > s_r > 0$, the Vandermonde system (7) is nonsingular and has a unique solution. Adding up consecutive multiplied instances of (5) yields

$$\begin{array}{rcl} \lambda_0 \log(|x_i(s_0)|) & = & \omega_i \lambda_0 \log(s_0) + \lambda_0 \log(|a_i|) + \sum_{j=1}^{\infty} c_{i,j} \lambda_0 s_0^j \\ & \vdots & \\ + \ \lambda_r \log(|x_i(s_r)|) & = & \omega_i \lambda_r \log(s_r) + \lambda_r \log(|a_i|) + \sum_{j=1}^{\infty} c_{i,j} \lambda_r s_r^j \\ \hline \sum_{k=0}^{r} \lambda_k \log(|x_i(s_k)|) & = & \omega_i \left( \sum_{k=0}^{r} \lambda_k \log(s_k) \right) + \sum_{j=r}^{\infty} c_{i,j} \sum_{k=0}^{r} \lambda_k s_k^j . \end{array} \tag{8}$$

By $\log(1 - t(s_k)) = m \log(s_k)$ we find that

$$\frac{\sum_{k=0}^{r} \lambda_k \log(|x_i(s_k)|)}{\frac{1}{m} \sum_{k=0}^{r} \lambda_k \log(1 - t(s_k))} = \omega_i + O(s_0^r). \tag{9}$$

While this gives an order $O(s_0^r)$ estimate of the quantities $\omega_i$, the error to the estimate is $O((1 - t(s_0))^{\frac{r}{m}})$. Furthermore, if the wrong value of $m$ is used, then the accuracy drops off to $O((1 - t(s_0))^{\frac{1}{m}})$, because only order $O(s_0)$ can be obtained. In this case, we obtain $\frac{\omega_i}{m}$ as estimate for the path direction.

If one follows the paths in projective space one ends up with a slightly different version of the extrapolation procedure. Here the solution path will have the form

$$\mathbf{z}(s) = [z_0(s) : z_1(s) : \cdots : z_n(s)] \tag{10}$$

where each $z_i(s) = a_i s^{\omega_i}(1 + O(s))$. The corresponding solution in affine space equals $x_i(s) = \frac{z_i(s)}{z_0(s)}$ so that the normal we want has coordinates $\omega_i - \omega_0$. The leading coefficient in the expansion of the affine solution $x_i(s)$ equals $\frac{a_i}{a_0}$.

Let $\lambda_0, \lambda_1, \ldots, \lambda_r$ be determined as in (7), then

$$\frac{\sum_{k=0}^{r} \lambda_k \big(\log(|z_i(s_k)|) - \log(|z_0(s_k)|)\big)}{\frac{1}{m} \sum_{k=0}^{r} \lambda_k \log(1 - t(s_k))} = \omega_i - \omega_0 + O(s_0^r) \tag{11}$$

determines the extrapolation procedure.

## 3.2 Approximating the cycle number by extrapolation

In their end games for polynomial continuation Morgan, Sommese and Wampler propose three methods to determine the cycle number: the circular sample [15], averaging [16] and fitting [17]. We briefly summarize these approaches.

The first method ([15]) follows the definition of the cycle number closely, as samples of the solution path are taken at evenly spaced points on a circle round $t = 1$. The circular sample is recommended when the condition of the Jacobian matrix forces us to stay relatively far away from the target. In [17] it is considered as a last resort when cheaper samples fail. Secondly, averaging is based on the observation that the spacing between estimated end points for paths converging to the same solution will be significantly smaller than the spacing between other solutions. Interpolation on truncated Taylor series is used in [16] to group the solution paths that converge to the same solution. Although this method is relatively inexpensive compared to the circular sample, some numerical experience is required to determine the parameters of the method. Thirdly, the idea of the fitting method in [17] is to consider the slope of the error for various predictions of the cycle number. The predicted value that gives the smallest error is taken as the guess. Although the last method also relies on power series, we propose here a more elegant and efficient way to compute the cycle number.

To obtain explicit approximations for $m$, we must perform a *geometric sample* of the solution paths: $1 - t_k = h^k(1 - t_0)$ or $s_k = h^{\frac{k}{m}} s_0$ for some $h$ with $0 < h < 1$. Define samples $e_i^{(k)}$ of the error expansion by subtraction of consecutive differences of (5):

$$e_i^{(k)} := (\log|x_i(s_k)| - \log|x_i(s_{k+1})|) - (\log|x_i(s_{k+1})| - \log|x_i(s_{k+2})|). \qquad (12)$$

Because the solution paths are sampled geometrically with ratio $h$, we derive

$$\begin{aligned} e_i^{(k)} &= \sum_{j=1}^{\infty} c_{i,j} s_k^j - 2\sum_{j=1}^{\infty} c_{i,j} s_{k+1}^j + \sum_{j=1}^{\infty} c_{i,j} s_{k+2}^j & (13) \\ &= \sum_{j=1}^{\infty} c_{i,j} (h^{\frac{k}{m}} s_0)^j - 2\sum_{j=1}^{\infty} c_{i,j} (h^{\frac{k+1}{m}} s_0)^j + \sum_{j=1}^{\infty} c_{i,j} (h^{\frac{k+2}{m}} s_0)^j & (14) \\ &= c_{i,1} h^{\frac{k}{m}} s_0 \left( 1 + \sum_{j=1}^{\infty} c'_{i,j} (h^{\frac{k}{m}} s_0)^j \right), & (15) \end{aligned}$$

where the coefficients $c'_{ij}$ are derived from the reordering of the terms in the power series. The exponent of the leading term can be isolated by consecutive differences of the logarithms, so that we obtain

$$\frac{1}{m}\log(h) = \log(e_i^{(k+1)}) - \log(e_i^{(k)}) + O(h^{\frac{k}{m}}). \qquad (16)$$

Thus we arrive at the estimate

$$m \approx \frac{\log(h)}{\log(e_i^{(k+1)}) - \log(e_i^{(k)})}. \qquad (17)$$

Note that $e_i^{(k)}$ is directly computable from the logarithms of path coordinates, this formula is equivalent to that arrived at by taking $e_i^{(k)} := \log(h)|w_i^{(k)} - w_i^{(k+1)}|$ where the $w_i^{(k)}$'s are the estimates of the path directions obtained at step $s_k$. Hereby we estimate the cycle number component-wise, though we could also set $e^{(k)} := \log(h)\sum_{i=1}^{n} |w_i^{(k)} - w_i^{(k+1)}|$ and apply (17) accordingly.

In Figure 1 we show the plots of a MATLAB-experiment, obtained from error expansions $e_i^{(k)}$, with randomly generated coefficients $c_{i,j}$. For $m = 2$ we see that formula (17) converges linearly, whereas for $m > 2$ the convergence is less dramatic. In view of the $O(h^{k/m})$-convergence of this approximation it is also critical here to use extrapolation to improve convergence, as will be elaborated in section 5.

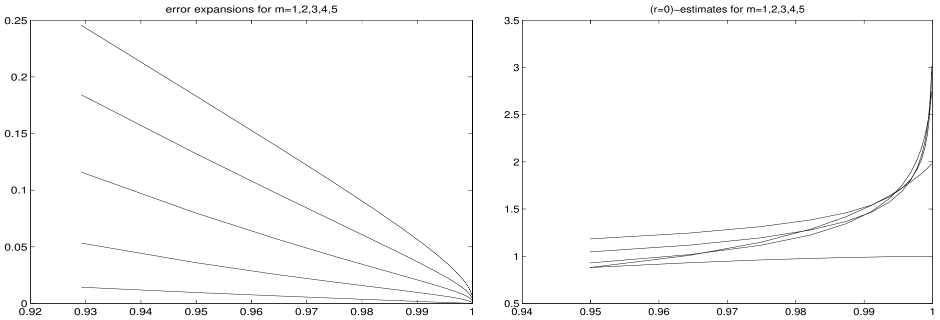


Figure 1: Plots of error expansion for various values of $m$, $m = 1, 2, 3, 4, 5$ and 0th-order estimates for $m$, respectively at the left and the right.

When paths are heading to a singular solution, we observe bad convergence of Newton's method, due to an ill-conditioned Jacobian matrix. In our setting it is natural to perform extrapolation on the power series expansion of the solution branch to estimate the coefficient of the dominant term more accurately. We refer to [7] for a review on this and related methods to improve the convergence of Newton's method in the presence of singularities.

## 3.3 Approximating the coefficients by transformation

Once we have estimated the normals and determined the face they support, we can recalculate the exact integer values for the normal (being careful that we have $m$ exactly and not some multiple).

The conceptually simplest technique to find the $a_i$ is to make a change of variables $u_i := x_i s^{-\omega_i}$, i.e.:

$$u_i(s) = a_i \left( 1 + \sum_{j=1}^{\infty} b_{i,j} s^j \right). \tag{18}$$

Let $\beta_i$ be the minimizing value of $\langle \cdot, \omega \rangle$ on $\mathcal{A}_i$.

If we let $h_i(\mathbf{x}, t) = (1-t) g_i(\mathbf{x}) + t f_i(\mathbf{x})$ be the $i$th polynomial of $H(\mathbf{x}, t)$ then

$$h_i(\mathbf{u}, s) = s^m g_i(\mathbf{u}) + (1 - s^m) f_i(\mathbf{u}) \tag{19}$$

is a polynomial in $s$ whose lowest-order term is precisely $s^{\beta_i}$ times the degeneration of $f_i$ to the face supported by $(\omega_1, \omega_2, \ldots, \omega_n)$. If we abuse notation slightly and define

$$H(\mathbf{u}, s) = (s^{-\beta_1} h_1(\mathbf{u}, s), s^{-\beta_2} h_2(\mathbf{u}, s), \ldots, s^{-\beta_n} h_n(\mathbf{u}, s)). \tag{20}$$

We can use a continuation method on $H(\mathbf{u}, s)$ taking $(\mathbf{u}(s_k), s_k)$ as starting points to determine $a_i = \lim_{s \to 0} u_i(s)$.

While the solutions of these transformed equations will generally be singular, they will at least lie in $(\mathbb{C}^*)^n$ and their multiplicities will be lower then the corresponding solutions of the original problem.

## 4 Richardson extrapolation

The explicit solution of the Vandermonde system in (7) should be avoided for reasons of efficiency and numerical stability. Instead we use an extrapolation method which allows the update of a new point of the path with a minimal amount of work. Our method can be considered as Richardson extrapolation on the logarithms of a power series. Although Richardson extrapolation is treated in any introductory book in numerical analysis, we recommend [3] and [28] as specialized works devoted to extrapolation methods.

We assume that the right value for the cycle number $m$ is known and that we are given samples for

$$\log(|x_i(s_{k+1})|) - \log(|x_i(s_k)|) = \omega_i(\log(s_{k+1}) - \log(s_k)) + \sum_{j=1}^{\infty} c_{ij}(s_{k+1}^j - s_k^j), \qquad (21)$$

for $s_k > s_{k+1} > 0$ and $k = 0, 1, \ldots, r-1$. To obtain more accurate estimates for $\omega_i$, we have to eliminate the dominant terms in the sum of (21). Although we can extrapolate without performing a geometric sample, the description of the algorithm becomes simpler if we take $s_k = h^{k/m} s_0$ for some $h$ with $0 < h < 1$. Then (21) becomes

$$\log(|x_i(s_{k+1})|) - \log(|x_i(s_k)|) = \omega_i \frac{\log(h)}{m} + \sum_{j=1}^{\infty} c_{ij} h^{\frac{kj}{m}} (h^{\frac{j}{m}} - 1) s_0^j. \qquad (22)$$

As example we give a fourth-order extrapolation. In (23) we see the structure of the table of computed values.

$$\begin{array}{lcccccc}
\log(|x_i(s_1)|) - \log(|x_i(s_0)|) = v_{01} & & & & & & \\
 & > & v_{012} & & & & \\
\log(|x_i(s_2)|) - \log(|x_i(s_1)|) = v_{12} & & & > & v_{0123} & & \\
 & > & v_{123} & & & > & v_{01234} \\
\log(|x_i(s_3)|) - \log(|x_i(s_2)|) = v_{23} & & & > & v_{1234} & & \\
 & > & v_{234} & & & & \\
\log(|x_i(s_4)|) - \log(|x_i(s_3)|) = v_{34} & & & & & &
\end{array} \qquad (23)$$

The general formula is given by

$$v_{k\ldots l} = v_{k\ldots l-1} + \frac{v_{k+1\ldots l} - v_{k\ldots l-1}}{1-h}, \quad \text{for} \left\{ \begin{array}{l} k = 0, 1, \ldots, r-1, \\ l = k+2, \ldots, r. \end{array} \right. \qquad (24)$$

So we obtain

$$\omega_i = m \frac{v_{0\ldots r}}{\log(h)} + O(s^r). \qquad (25)$$

Note that for the update of a new point along the path only the last row of each table is needed. However, when $m$ changes, the whole table must be recomputed.

One of the advantages of organizing the extrapolation method in this way, is that we gain control over the quality of extrapolation, over choosing the right order. Table 1 presents the errors of the approximations generated on a random example. We see that a fifth-order extrapolator yields the best results.

Enlarging the range of $s$-values where we can apply the extrapolation also significantly enhances the precision, as illustrated by the errors in Table 2. We see, however, that the results might get spoiled when the value for $s_r$ becomes too tiny.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1.0E-02 | | | | | | | | |
| 9.8E-03 | 3.5E-02 | | | | | | | |
| 9.6E-03 | 3.4E-02 | 3.0E-04 | | | | | | |
| 9.4E-03 | 3.4E-02 | 2.9E-04 | 5.0E-06 | | | | | |
| 9.2E-03 | 3.3E-02 | 2.8E-04 | 4.6E-06 | 2.2E-07 | | | | |
| 9.0E-03 | 3.2E-02 | 2.6E-04 | 4.3E-06 | 2.1E-07 | 1.1E-07 | | | |
| 8.8E-03 | 3.2E-02 | 2.5E-04 | 4.1E-06 | 1.8E-07 | 9.2E-08 | 1.9E-06 | | |
| 8.6E-03 | 3.1E-02 | 2.4E-04 | 3.8E-06 | 1.7E-07 | 4.5E-08 | 1.2E-06 | 2.4E-05 | |
| 8.4E-03 | 3.0E-02 | 2.3E-04 | 3.5E-06 | 1.5E-07 | 3.7E-08 | 7.4E-07 | 1.5E-05 | 2.5E-04 |
| 8.2E-03 | 2.9E-02 | 2.2E-04 | 3.3E-06 | 1.4E-07 | 4.3E-08 | 7.1E-07 | 1.1E-05 | 1.6E-04 |

Table 1: Table with errors, the first column contains the $s$-values.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8.0E-02 | | | | | | | | |
| 4.0E-02 | 6.5E-02 | | | | | | | |
| 2.0E-02 | 3.1E-02 | 3.2E-03 | | | | | | |
| 1.0E-02 | 1.5E-02 | 9.0E-04 | 1.3E-04 | | | | | |
| 5.0E-03 | 7.3E-03 | 2.4E-04 | 1.7E-05 | 1.2E-06 | | | | |
| 2.5E-03 | 3.6E-03 | 6.1E-05 | 2.2E-06 | 7.7E-08 | 2.1E-09 | | | |
| 1.2E-03 | 1.8E-03 | 1.5E-05 | 2.8E-07 | 4.8E-09 | 5.1E-11 | 1.5E-11 | | |
| 6.2E-04 | 9.0E-04 | 3.9E-06 | 3.5E-08 | 3.0E-10 | 1.3E-12 | 2.8E-13 | 4.6E-14 | |
| 3.1E-04 | 4.5E-04 | 9.8E-07 | 4.4E-09 | 1.9E-11 | 3.2E-14 | 1.0E-14 | 5.8E-15 | 5.6E-15 |
| 1.6E-04 | 2.3E-04 | 2.5E-07 | 1.0E-08 | 1.3E-08 | 1.4E-08 | 1.4E-08 | 1.4E-08 | 1.4E-08 |

Table 2: Table with errors, the first column contains the $s$-values.

In Table 3 we show the influence of a change of $m$ on a real-life application (the system `cassou`, see section 7). The errors are obtained by subtracting consecutive approximations.

# 5 An extrapolation method for the cycle number

To obtain more accurate approximations for $m$ than (17), we must eliminate the higher-order terms of (26):

$$\log(e_i^{(k+1)}) - \log(e_i^{(k)}) = \frac{1}{m}\log(h) + \sum_{j=1}^{\infty} c''_{i,j} h^{\frac{kj}{m}} (h^{\frac{j}{m}} - 1) s_0^j. \tag{26}$$

The coefficients $c''_{i,j}$ are derived from the expansion of the logarithm of the right-hand side of (13). We propose to use lower-order approximations for $m$, as needed in the computation of the factors $h^{\frac{1}{m}}$ used in the extrapolation on (26). This method can be seen as a particular case of repeated Richardson extrapolation, as proposed in [4] (see also [3]). Our case is particular because it suffices to estimate but one exponent.

In (27) we show the extrapolation table for $r = 4$.

| | | | | | |
|---|---|---|---|---|---|
| Extrapolation data with $r = 4$ and $m = 1$: | | | | | |
| 3.516E-04 | 2.095E-03 | 1.275E-03 | 1.021E-03 | 9.032E-04 | -6.14183790759625E-03 |
| 3.516E-04 | 2.857E-03 | 1.737E-03 | 1.391E-03 | 1.231E-03 | 8.36854378574308E-03 |
| 3.516E-04 | 2.852E-03 | 1.730E-03 | 1.386E-03 | 1.226E-03 | 8.33524580077066E-03 |
| 3.516E-04 | 2.153E-03 | 1.280E-03 | 1.026E-03 | 9.075E-04 | -5.06172469313672E-01 |
| Extrapolation after adjusting $m$ from 1 to 2: | | | | | |
| 1.875E-02 | 1.263E-02 | 5.358E-03 | 5.782E-03 | 5.234E-03 | -4.27880775736961E-06 |
| 1.875E-02 | 1.722E-02 | 7.318E-03 | 7.875E-03 | 7.132E-03 | 6.03853343130524E-06 |
| 1.875E-02 | 1.717E-02 | 7.327E-03 | 7.845E-03 | 7.104E-03 | 5.92564608977373E-06 |
| 1.875E-02 | 5.128E-01 | 4.943E-01 | 4.942E-01 | 4.947E-01 | -1.00000509348483E+00 |
| Extrapolation data with $r = 4$ and $m = 2$: | | | | | |
| 1.568E-02 | 3.806E-03 | 5.339E-04 | 3.538E-05 | 2.038E-06 | -2.24040227098794E-06 |
| 1.568E-02 | 5.189E-03 | 7.219E-04 | 4.927E-05 | 2.871E-06 | 3.16718489453018E-06 |
| 1.568E-02 | 5.178E-03 | 7.082E-04 | 4.869E-05 | 2.817E-06 | 3.10857245558752E-06 |
| 1.568E-02 | 3.896E-03 | 4.508E-04 | 4.029E-05 | 2.405E-06 | -1.00000268888446E+00 |

Table 3: Progress of the errors on four components for various orders, as $m$ changes. The first column contains the $s$-values, the last one the actual approximations for the path direction, which equals $(0, 0, 0, -1)$.

$$\begin{array}{lllll}
\log(e_i^{(1)}) - \log(e_i^{(0)}) = e_i^{(01)} & & & & \\
 & > e_i^{(012)} & & & \\
\log(e_i^{(2)}) - \log(e_i^{(1)}) = e_i^{(12)} & & > e_i^{(0123)} & & \\
 & > e_i^{(123)} & & > e_i^{(01234)} & \\
\log(e_i^{(3)}) - \log(e_i^{(2)}) = e_i^{(23)} & & > e_i^{(1234)} & & \\
 & > e_i^{(234)} & & & \\
\log(e_i^{(4)}) - \log(e_i^{(3)}) = e_i^{(34)} & & & &
\end{array} \tag{27}$$

The formulas for computing the values in (27) are given by

$$e_i^{(k\ldots l)} = e_i^{(k+1\ldots l)} + \frac{e_i^{(k\ldots l-1)} - e_i^{(k+1\ldots l)}}{1 - h^{(k\ldots l-1)}}, \tag{28}$$

with

$$h^{(k\ldots l)} = h^{\frac{l-k-1}{m^{(k\ldots l)}}} \quad \text{and} \quad m^{(k\ldots l)} = \frac{\log(h)}{e_i^{(k\ldots l)}}, \tag{29}$$

for $k = 0, 1, \ldots, r$, $l = k + 2, \ldots, r$. Note that the estimates $m^{(k\ldots l)}$ are real numbers.

The cost of the update of this table equals only $O(r)$ if we store the last diagonal of (27). However, for accuracy it is recommended to arrange the computations column-wise, so that the same lower-order estimate for $m$ can be used to proceed.

In Figure 2 we show the result of the extrapolation on the same data as depicted in Figure 1. On Figure 2 we see that the extrapolation clearly helps in determining the cycle number, also for higher values of $m$. As the order increases, the distinction between the estimates for $m = 1, 2, 3, 4, 5$ becomes more and more dramatic.

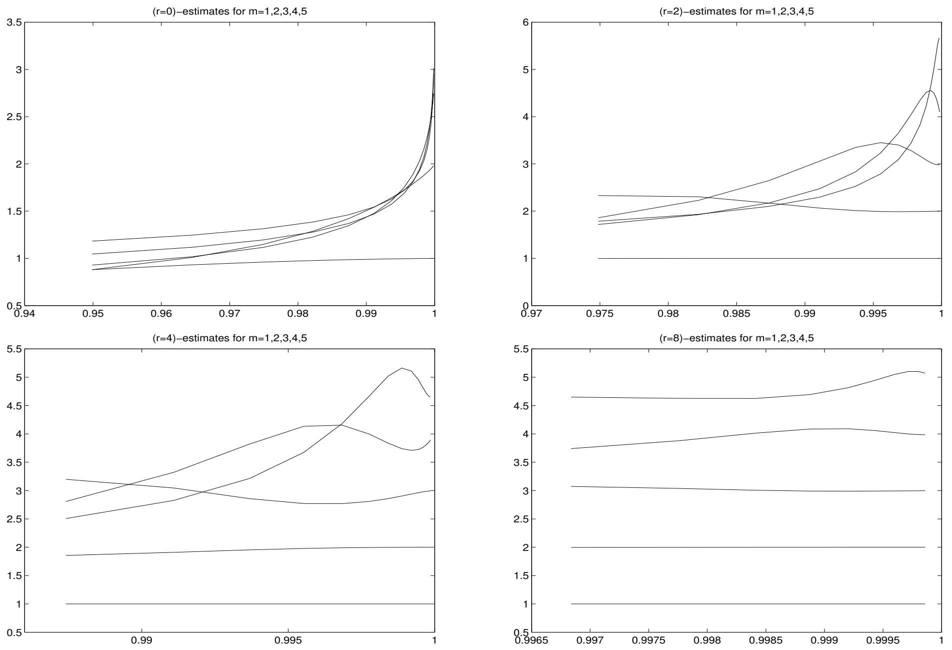


Figure 2: Higher-order estimates for $m$ by extrapolation. From the upper-left to the lower-right corner we respectively see estimates of order 0,2,4 and 8.

Next we formulate our main result, which shows that the proposed extrapolation method is numerically stable. We may use formula (17) to obtain higher estimates of $m$, because the error on the multiplication factor in the extrapolation is $O(h^{k/m})$, which renders $O(h^{2k/m})$ when multiplied with the dominant term of the power series (26).

**Theorem 5.1** *Let* $\frac{1}{M} = \frac{1}{m} + ch^{k/m}$*, then* $\exists c' > 0$*:* $|h^{1/M} - h^{1/m}| \leq c'h^{k/m}$*, for* $k \gg m$*.*

*Proof.* We may write

$$h^{\frac{1}{M}} = h^{\frac{1}{m}} h^{ch^{\frac{k}{m}}} + h^{\frac{1}{m}} - h^{\frac{1}{m}} = h^{\frac{1}{m}} + h^{\frac{1}{m}} \left( h^{ch^{\frac{k}{m}}} - 1 \right). \tag{30}$$

We want to prove the following inequality on the error:

$$-c'h^{\frac{k}{m}} \leq h^{\frac{1}{m}} (h^{ch^{\frac{k}{m}}} - 1) \leq c'h^{\frac{k}{m}} \quad \text{or equivalently} \quad 1 - c'h^{\frac{k-1}{m}} \leq h^{ch^{\frac{k}{m}}} \leq 1 + c'h^{\frac{k-1}{m}}. \tag{31}$$

By taking logarithms we arrive at

$$\log(1 - c'h^{\frac{k-1}{m}}) \leq h^{\frac{k}{m}} \log(h) \leq \log(1 + c'h^{\frac{k-1}{m}}). \tag{32}$$

For $k \gg m$, the first terms of the power series of the logarithms are dominant, so we have

$$-c'h^{\frac{k-1}{m}} \leq h^{\frac{k}{m}} \log(h) \leq c'h^{\frac{k-1}{m}} \quad \text{or equivalently} \quad -c' \leq h^{\frac{1}{m}} \log(h) \leq c'. \tag{33}$$

Hence there exists an appropriate value for $c'$ that satisfies the inequality. □

Note however that, as with the extrapolation on the path directions, the estimates may get spoiled by rounding errors, as the distances between two consecutive approximations become too tiny, or equivalently, when $h \approx 1$. We will next show how in the implementation we can deal with this problem.

# 6 Implementation issues

The extrapolation process relies on points $(\mathbf{x}(t_k), t_k)$ sampled along a solution path for $t_k$ close enough to 1 for (3) to hold. Unfortunately since the continuation problem defining this path becomes ill-conditioned as $t \approx 1$, we must also require $t_k$ to be far enough from 1 that points on the path can be determined with acceptable accuracy. These competing considerations define the *end-game operating range* introduced in [17]. We use a pre-set bound on the condition of the Jacobi matrix to determine when $t$ is too close to 1.

Note that the geometric sample is natural in an end-game setting, because as the path-following becomes more difficult when $t \approx 1$, the adaptive step-size control requires a lot of failures to reduce step size. With a geometric series, the step size decreases automatically. Still, the number of samples near the end of the path commonly exceeds the number of steps before the start of the end game.

In the automatic estimation of $m$ we need to set a threshold value for augmenting $m$. This means that the estimated value for $m$ will only be augmented after a number of consecutive guesses that yielded the same value. This prevents random changes in the convergence behavior to have an influence on the estimated value for $m$.

The extrapolation methods suffer from rounding errors when the distances between the $s$-values become too tiny. To arrive at greater distances, we suggest invoking the extrapolation with a certain delay, i.e.: not after every step, but on the results obtained after every fourth or fifth step. For instance, $0.9^5 \approx 0.6$.

The numerical techniques operate in two stages: during the continuation stage, the extrapolation method is applied to approximate the normals to the faces. In the validation stage, the program builds a frequency table of path directions and computes the corresponding face systems.

# 7 Applications

The algorithms described above have been implemented and added to the software package PHC, presented in [23] and publicly available at [22]. We have tested the methods on several deficient polynomial systems derived from practical applications. Next we list an overview of the characteristics and comment on the computational results.

## 7.1 An overview on the benchmark

In Table 4 we give an overview of our benchmark. To save space, we omit the algebraic description of the problems (available from the web site at [22]) and provide only the reference.

| polynomial systems | | | characteristics | | | | #roots | |
|---|---|---|---|---|---|---|---|---|
| name | description | ref | $n$ | $D$ | $B$ | $MV$ | $\mathbb{C}$ | $\mathbb{R}$ |
| boon | neurofysiology application | [24] | 6 | 1,024 | 216 | 20 | 8 | 8 |
| cassou | system of Cassou-Noguès | [11] | 4 | 1,344 | 312 | 24 | 16 | 4 |
| ipp | six-revolute joint | [14] | 8 | 256 | 96 | 64 | 32 | 10 |
| heart | heart-dipole problem | [18] | 8 | 576 | 193 | 121 | 4 | 2 |
| d1 | interval benchmark D1 | [8] | 12 | 4,068 | 320 | 192 | 48 | 16 |
| cyclic8 | cyclic 8-roots problem | [2] | 8 | 40,320 | 20,352 | 2,560 | 1,152 | 64 |

Table 4: Characteristics of polynomial systems in benchmark: dimension $n$, total degree $D$, best known Bézout bound $B$ and mixed volume $MV$. The last two columns contain the total number of finite isolated solutions, respectively in complex and in real space.

The root counts $D$, $B$ and $MV$ listed in Table 4 give an idea about the intrinsic cost of the respective homotopy continuation methods based on these root counts. The gap between the actual number of finite solutions illustrates their (sometimes poor) performance. Jointly with the dimension $n$, the mixed volume $MV$ provides already an estimate for the amount of computational work. In Table 5 are the results of the execution times on one node of a IBM SP2 computer.

| systems | $MV$ | $G(\mathbf{x})$ | $[0, 0.9]$ | $[0.9, 1]$ | total |
|---|---|---|---|---|---|
| boon | 0.03 | 0.75 | 1.08 | 1.02 | 1.88 |
| cassou | 0.13 | 1.03 | 17.89 | 56.53 | 75.58 |
| ipp | 7.71 | 5.33 | 13.34 | 24.81 | 51.19 |
| heart | 17.48 | 23.85 | 41.91 | 15.92 | 99.16 |
| d1 | 3.00 | 53.30 | 50.67 | 32.61 | 139.58 |
| cyclic8 | 186.23 | 1123.17 | 1090.98 | 469.86 | 2870.18 |

Table 5: Timings in cpu sec. for computing the mixed volume ($MV$), solving a random coefficient start system ($G(\mathbf{x})$), the continuation before the end game ($[0, 0.9]$), for the end game ($[0.9, 1]$) and the total timings.

## 7.2 On the computed path directions

The settings in our experiments are as follows. For all applications we use a fourth-order (i.e.: $r = 4$) extrapolation method. The threshold for augmenting the estimate for $m$ equals five. A typical value for the tolerance on the inverse condition of the Jacobi matrix is $10^{-10}$ to mark the end-game operating range. As measure for the accuracy of the computed path direction, we take the difference between the two last consecutive approximations.

`boon` All diverging paths go to the same face. For this face, we derive the outer normal $(0, 0, 0, 0, -1, -1)$ from the path directions, computed with an accuracy about $10^{-6}$. The estimated $m$ is one for all paths.

`cassou` There are eight diverging solution paths, all with the same path direction, equal to $(0, 0, 0, -1)$, computed with accuracy ranging between $10^{-5}$ and $10^{-7}$. The structure of the face system is so simple that the solution can be read off: $(b, c, c, e)$, for any $b, c, e \in \mathbb{C}$. The estimated $m$ is two for all diverging paths.

`ipp` Sixteen paths diverge to infinity, to the face with outer normal $(-1, -1, 0, 0, -1, -1, 0, 0)$. The accuracy of the computed path direction ranges between $10^{-3}$ and $10^{-5}$, whereas the approximations to the zero components in the normal range from $10^{-9}$ and $10^{-12}$. The estimated $m$ equals one for all paths. The deficient face is hard to recognize when looking at the system. Another important advantage of the polyhedral end game is that the diverging paths to infinity can be separated from the other sixteen paths which go to the connected component of solutions.

`heart` Of the 121 paths, only four paths converge to finite solutions. The two distinct deficient faces have outer normals $(-1, -1, -1, -1, 0, 0, 0, 0)$ and $(0, 0, 0, 0, -1, -1, -1, -1)$ respectively. Sixteen paths are diverging to the second face, 101 to the first one. The accuracy for the path directions to the first deficient face ranges between $10^{-2}$ and $10^{-7}$. For the second face, the accuracy lies between $10^{-1}$ and $10^{-3}$. The estimated $m$ equals one for all paths.

`d1` For this system there are six deficient faces. In Table 6 we list their outer normals, as derived from the computed path directions.

| outer normals | $m$ | accuracy | #paths |
|---|---|---|---|
| $(-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ | 1 | $10^{-10}$ | 48 |
| $(0, 0, -1, -1, -1, -1, -1, -1, 0, 0, 0, 0)$ | 1 | $10^{-8}$ | 32 |
| $(0, 0, -2, -2, -2, -2, -2, -2, 2, 0, -1, -1)$ | 2 | $10^{-4}$ | 16 |
| $(0, 0, -2, -2, -2, -2, 0, 0, 2, 0, -1, -1)$ | 2 | $10^{-4}$ | 16 |
| $(0, 0, -2, -2, 0, 0, -2, -2, 2, 0, -1, -1)$ | 2 | $10^{-4}$ | 16 |
| $(0, 0, 0, 0, -2, -2, -2, -2, 2, 0, -1, -1)$ | 2 | $10^{-4}$ | 16 |

Table 6: The outer normals to the deficient faces of system D1.

The structure of the deficient faces is more complicated as some deficient faces are subfaces of other deficient ones. Despite the lower accuracy, measured by the differences of two consecutive path directions, the zeros in the components are approximated to within $10^{-7}$.

`cyclic8` To solve cyclic 8-roots, we add to each component a randomly chosen complex constant. The solutions to this modified problem are all isolated. From the 2,560 paths, 192 diverge to infinity. The outer normals as derived from the path directions are listed in Table 7, as two different orbits respecting the symmetry.

We can use the modified cyclic 8-roots problem as start system in a homotopy to solve the original cyclic 8-roots problem. By doing this we can exploit the symmetry in a straightforward way, so that we only need to follow the 148 generating solution paths. Among the diverging paths, no new deficient faces are found.

| outer normals | $m$ | accuracy | #paths |
|---|---|---|---|
| $(-1,1,-1,1,-1,1,-1,1)$ | 1 | $10^{-3}$ | 32 |
| $(1,-1,1,-1,1,-1,1,-1)$ | 1 | $10^{-3}$ | 32 |
| $(-1,0,0,1,0,-1,1,0)$ | 1 | $10^{-7}$ | 8 |
| $(0,-1,0,0,1,0,-1,1)$ | 1 | $10^{-6}$ | 8 |
| $(1,0,-1,0,0,1,0,-1)$ | 1 | $10^{-7}$ | 8 |
| $(-1,1,0,-1,0,0,1,0)$ | 1 | $10^{-6}$ | 8 |
| $(0,-1,1,0,-1,0,0,1)$ | 1 | $10^{-6}$ | 8 |
| $(1,0,-1,1,0,-1,0,0)$ | 1 | $10^{-7}$ | 8 |
| $(0,1,0,-1,1,0,-1,0)$ | 1 | $10^{-6}$ | 8 |
| $(0,0,1,0,-1,1,0,-1)$ | 1 | $10^{-6}$ | 8 |
| $(0,1,-1,0,1,0,0,-1)$ | 1 | $10^{-6}$ | 8 |
| $(-1,0,1,-1,0,1,0,0)$ | 1 | $10^{-6}$ | 8 |
| $(0,-1,0,1,-1,0,1,0)$ | 1 | $10^{-6}$ | 8 |
| $(0,0,-1,0,1,-1,0,1)$ | 1 | $10^{-6}$ | 8 |
| $(1,0,0,-1,0,1,-1,0)$ | 1 | $10^{-6}$ | 8 |
| $(0,1,0,0,-1,0,1,-1)$ | 1 | $10^{-5}$ | 8 |
| $(-1,0,1,0,0,-1,0,1)$ | 1 | $10^{-6}$ | 8 |
| $(1,-1,0,1,0,0,-1,0)$ | 1 | $10^{-5}$ | 8 |

Table 7: The outer normals to the deficient faces of modified cyclic 8-roots.

## 8 Conclusions

The polyhedral end game augments the robustness of polynomial homotopy continuation methods, because we are now able to separate the solutions at infinity from the other singular solutions. As future research direction we hope to gain insight in dealing with deficient polynomial systems more directly, either by constructing more performant homotopies or by reformulating the problem.

## References

[1] 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.

[2] G. Björk and R. Fröberg. Methods to "divide out" certain solutions from systems of algebraic equations, applied to find all cyclic 8-roots. In M. Gyllenberg and L.E. Persson, editors, *Analysis, Algebra and Computers in Math. research*, volume 564 of *Lecture Notes in Mathematics*, pages 57–70. Dekker, 1994.

[3] C. Brezinski and M. Redivo Zaglia. *Extrapolation Methods*, volume 2 of *Studies in Computational Mathematics*. North-Holland, 1991.

[4] E. Christiansen and H.G. Petersen. Estimation of convergence orders in repeated Richardson extrapolation. *BIT*, 29(1), 1989.

[5] G. Ewald. *Combinatorial Convexity and Algebraic Geometry*, volume 168 of *Graduate Texts in Mathematics*. Springer–Verlag, New York, 1996.

[6] W. Fulton. *Introduction to Toric Varieties*, volume 131 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, New Jersey, 1993.

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

[8] H. Hong and V. Stahl. Safe starting regions by fixed points and tightening. *Computing*, 53(3–4):322–335, 1995.

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

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

[11] T.Y. Li. Numerical solutions of multivariate polynomial systems by homotopy continuation methods. *Acta Numerica*, 6:399–436, 1997.

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

[13] A. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems*. Prentice-Hall, Englewood Cliffs, N.J., 1987.

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

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

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

[17] 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(3):391–409, 1992.

[18] C.V. Nelsen and B.C. Hodgkin. Determination of magnitudes, directions, and locations of two independent dipoles in a circular conducting region from boundary potential measurements. *IEEE Trans. Biomed. Engrg.*, BME-28(12):817–823, 1981.

[19] J. Rojas. Twisted Chow forms and toric perturbations of degenerate polynomial systems. Submitted for publication. Available at http://www-math.mit.edu/~rojas.

[20] J.M. Rojas. Toric laminations, sparse generalized characteristic polynomials, and a refinement of Hilbert's tenth problem. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics. Selected Papers of a Conference, Held at IMPA in Rio de Janeiro, January 1997*, pages 369–381. Springer–Verlag, 1997. Revised version available at http://www-math.mit.edu/~rojas.

[21] 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.

[22] J. Verschelde. PHCPACK: A general-purpose solver for polynomial systems by homotopy continuation. Report TW 265, Dept. of Computer Science, K.U.Leuven, 1997. Software available at http://www.math.msu.edu/~jan.

[23] J. Verschelde and R. Cools. Polynomial Homotopy Continuation: A portable Ada software package. *The Ada Belgium Newsletter*, 4:59–83, 1996. Proceedings of the 1996 Ada-Belgium Seminar, 22 November 1996, Eurocontrol, Brussels, Belgium.

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

[25] 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.

[26] 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.

[27] R. Walker. *Algebraic Curves.* Springer–Verlag, New York, second edition, 1978.

[28] J. Wimp. *Sequence Transformations and Their Applications*, volume 154 of *Mathematics in Science and Engineering.* Academic Press, New York, 1981.