Newton's Method with Deflation for Isolated Singularities of Polynomial Systems

Anton Leykin[*] Jan Verschelde[†] Ailing Zhao[‡]

13 January 2006

Abstract

We present a modification of Newton's method to restore quadratic convergence for isolated singular solutions of polynomial systems. Our method is symbolic-numeric: we produce a new polynomial system which has the original multiple solution as a regular root. Using standard bases, a tool for the symbolic computation of multiplicities, we show that the number of deflation stages is bounded by the multiplicity of the isolated root. Our implementation performs well on a large class of applications.

2000 Mathematics Subject Classification. Primary 65H10. Secondary 14Q99, 68W30.

Key words and phrases. Newton's method, deflation, numerical homotopy algorithms, symbolic-numeric computations.

1 Introduction

Let $F(\mathbf{x}) = \mathbf{0}$ be a polynomial system of N equations in n unknowns $\mathbf{x} \in \mathbb{C}^n$. We are interested in $\mathbf{x}^*$, an *isolated* solution of $F(\mathbf{x}) = \mathbf{0}$:

$$\text{for small enough } \epsilon > 0 : \{ \ \mathbf{y} \in \mathbb{C}^n : ||\mathbf{y} - \mathbf{x}^*|| < \epsilon \ \} \cap F^{-1}(\mathbf{0}) = \{\mathbf{x}^*\}. \qquad (1)$$

Denote by $A(\mathbf{x})$ the Jacobian matrix of the system $F(\mathbf{x}) = \mathbf{0}$. We call $\mathbf{x}^*$ a *singular* solution of $F(\mathbf{x}) = \mathbf{0} \Leftrightarrow \text{rank}(A(\mathbf{x}^*)) < n$. Let m be the *multiplicity* of the isolated solution $\mathbf{x}^*$ of $F(\mathbf{x}) = \mathbf{0}$.

[*] Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, USA. *Email:* leykin@math.uic.edu. *URL:* http://www.math.uic.edu/˜leykin.

[†] 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. 0105739 and Grant No. 0134611.

[‡] Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, USA. *Email:* azhao1@uic.edu. *URL:* http://www.math.uic.edu/˜azhao1.

Newton's method (also called[1] the method of Gauss-Newton when $N > n$, see [49]) generates a sequence of approximations $\mathbf{x}_k$ for $\mathbf{x}^*$. If $\mathbf{x}^*$ is nonsingular, then the sequence converges quadratically (i.e.: $\|\mathbf{x}_k - \mathbf{x}_{k+1}\| = O(\|\mathbf{x}_{k-1} - \mathbf{x}_k\|^2)$) to $\mathbf{x}^*$, which justifies its widespread usage. But otherwise, if $\mathbf{x}^*$ is singular, the convergence slows down and gets lost when $\mathbf{x}_k \approx \mathbf{x}^*$.

A straightforward approach is to use a working precision of $m \times D$ decimal places to achieve D correct decimal places in the final approximation. Even as multiprecision arithmetic is widely available and nowadays less expensive to use, this approach can only work if all coefficients in the system F have their first $m \times D$ decimal places correct. Our goal is to restore the quadratic convergence of a sequence converging to an isolated singular root without imposing extra requirements of precision on F. This means that we can compute isolated singularities with great accuracy in standard machine arithmetic, effectively *reconditioning* the problem.

Newton's method for singular solutions has been extensively researched. The research up to the mid eighties is surveyed in [23]. We classify research on Newton's method related to our work in two domains:

1. **Detection and treatment of bifurcation points.** When following a solution path of a system defined by a parameter, the solution path may turn back or bifurcate for increasing values of the parameter. Techniques to detect and compute such bifurcation points are generally done via Lyapunov-Schmidt reduction. General references are [1], [11], and [19]; see also [36, 37], [12], and [29]. One could interpret our method as a recursive application of the methods used to compute bifurcation points.

2. **Deflation method for polynomial systems.** A symbolic deflation method was presented in [48], and further developed in [45], [46], and [47]. We discovered this approach from the reference list of [31], which offers a symbolic deflation method whose complexity is quadratic in the multiplicity. As first announced in [57], we provide a numerically stable implementation of a modified symbolic deflation method.

A theoretical framework to study the complexity and numerical stability of Newton's method was developed by Shub and Smale, see [4], and was generalized to overdetermined systems in [10]. See [17, 18] for recent generalizations of this α-theory to multiple roots.

Singular solutions of polynomial systems are investigated in computational algebraic geometry, in particular we distinguish:

1. **Standard bases in computer algebra.** SINGULAR [22] allows the computation of standard bases [21], implementing generalizations [20] of the algorithms in [41]. We use standard bases to show that the number of

[1] In the light of the historical development outlined in [58], one should call Newton's method the Newton-Raphson-Simpson method.

deflations to restore the quadratic convergence of Newton's method is bounded by the multiplicity.

2. **Dual bases in numerical polynomial algebra.** The dual of an ideal, studied by Macaulay [39] with the goal of capturing multiplicity, is relevant from the point of numerical computations, see [54] and also [44]. Differential operators define Gröbner duality [42], providing suitable representations for multiple roots [40].

Similar as in [6], the only numerical parameter needed in our deflation method is a tolerance to decide the rank of a matrix. Just as we deferring the problem of region of convergence, counting on homotopy continuation methods [1, 52] to give us a good initial approximation for an isolated singularity, we defer to methods in numerical linear algebra [15, 38] to determine the numerical rank of a matrix.

In the next section we describe our method, followed by an introduction to standard bases and our proof in the third section. Our symbolic-numeric implementations and numerical results are described in sections four and five.

Acknowledgements. Our modified method with numerical results was presented at a poster session at ISSAC'04, 6 July 2004, Santander, Spain. We are grateful for the reactions of Grégoire Lecerf – in particular for pointing at the work of Kunkel. We thank Erich Kaltofen and Zhonggang Zeng for useful remarks at that poster session. Also discussions with Andrew Sommese and Charles Wampler at Oberwolfach helped us. We are grateful to the referees for helpful suggestions.

2 A Modified Deflation Method

A singular root $\mathbf{x}^*$ of a square (i.e.: $N = n$) system $F(\mathbf{x}) = \mathbf{0}$ with Jacobian matrix $A(\mathbf{x})$ satisfies

$$\left\{ \begin{array}{r} F(\mathbf{x}) = \mathbf{0} \\ \det(A(\mathbf{x})) = 0. \end{array} \right. \tag{2}$$

The augmented system (2) forms the basic idea for deflation. If $\mathbf{x}^*$ is isolated and $\mathrm{corank}(A(\mathbf{x}^*)) = 1$, then $\mathbf{x}^*$ as root of (2) has a lower multiplicity.

We find deflation used repeatedly first in [48], and later modified in [45] and applied in [46, 47].

In theory, $\det(A(\mathbf{x})) = 0$ (or maximal minors) could be used to form new equations. But this is neither good symbolically because the determinant is usually of high degree and leads to expression swell, nor numerically, as evaluating polynomials of high degree is numerically unstable.

Instead of using the determinant, on a system F of N equations in n variables, we proceed along the following three steps to form new equations:

1. Let $r = \operatorname{rank}(A(\mathbf{x}_0), \epsilon)$ for $\mathbf{x}_0 \approx \mathbf{x}^*$ and tolerance ϵ, $0 < \epsilon \ll 1$. For numerical stability, we compute the rank via a Singular Value Decomposition (SVD) of the matrix $A = A(\mathbf{x}_0)$. The numerical rank r equals the number of singular values larger than the tolerance ϵ.
2. Let $\mathbf{h} \in \mathbb{C}^{r+1}$ be a random vector. For numerical stability, we generate random numbers on the complex unit circle. We use $\mathbf{h}$ as scaling equation to obtain a unique vector in the kernel of the Jacobian matrix.
3. Let $B \in \mathbb{C}^{n \times (r+1)}$ be a random matrix, also with numbers on the complex unit circle. Using B, we form $C(\mathbf{x}) = A(\mathbf{x})B$. Notice that $C = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{r+1}]$ is an $N \times (r+1)$ matrix with polynomial entries.

With probability one (exceptional pairs of vectors $\mathbf{h}$ and matrices B belong to a proper algebraic subset of $\mathbb{C}^{r+1} \times \mathbb{C}^{n\times(r+1)}$) we have

$$\begin{aligned} & \operatorname{rank}(A(\mathbf{x}^*)) = r \Leftrightarrow \operatorname{corank}(C(\mathbf{x}^*)) = 1 \\ & \Leftrightarrow \text{there is a unique } \boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{r+1} \end{pmatrix} : G(\mathbf{x}^*, \boldsymbol{\lambda}) = \begin{pmatrix} \displaystyle\sum_{i=1}^{r+1} \lambda_i \mathbf{c}_i(\mathbf{x}^*) \\ \displaystyle\sum_{i=1}^{r+1} h_i \lambda_i - 1 \end{pmatrix} = \mathbf{0}. \end{aligned} \tag{3}$$

The random $\mathbf{h}$ and B guarantee the existence and uniqueness of the solution $\boldsymbol{\lambda}$ to $G(\mathbf{x}, \boldsymbol{\lambda})$ when $\mathbf{x} = \mathbf{x}^*$. Note that[2] instead of multiplying the Jacobian matrix of $F(\mathbf{x}) = \mathbf{0}$ by B, we could use B for a generic coordinate change $\mathbf{x} = B\mathbf{y}$, which would after application of the chain rule on $F(B\mathbf{y})$ be equivalent to the formation of $C = A(\mathbf{x})B$.

In one deflation step, we add the equations of $G(\mathbf{x}, \boldsymbol{\lambda})$ instead of $\det(A(\mathbf{x})) = 0$ to the system $F(\mathbf{x}) = \mathbf{0}$, adding $r+1$ extra variables λ_1, λ_2, …, λ_{r+1}. The flowchart for our modified deflation algorithm is displayed in Figure 1.

3 A Bound on the Number of Deflations

The termination of our algorithm in Figure 1 depends on the following theorem.

Theorem 3.1 *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.*

The answer to the question "How much less?" can be understood by looking at a standard basis for the ideal generated by the given polynomials in the system. We use standard bases to prove the termination of our algorithm, as explained in the next two subsections.

[2] We thank Alistair Spence for pointing this observation out to us, referring to [11]

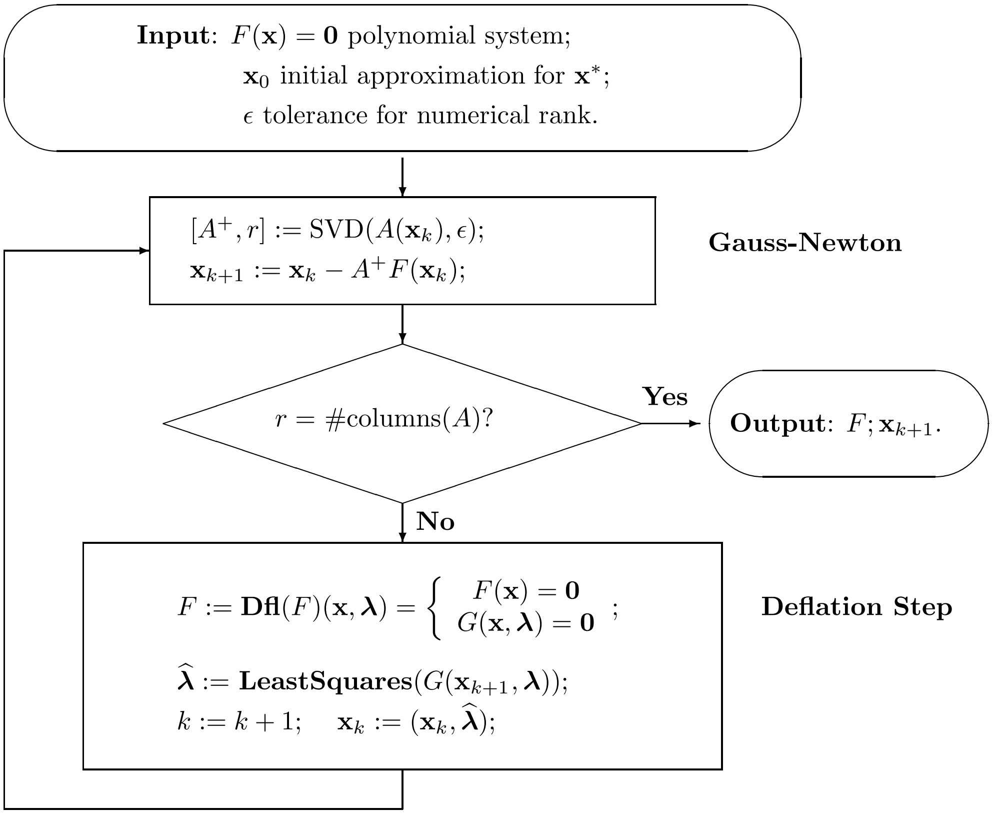
Input: $F(\mathbf{x}) = \mathbf{0}$ polynomial system;
$\mathbf{x}_0$ initial approximation for $\mathbf{x}^*$;
ϵ tolerance for numerical rank.
$[A^+, r] := \mathrm{SVD}(A(\mathbf{x}_k), \epsilon)$;
$\mathbf{x}_{k+1} := \mathbf{x}_k - A^+ F(\mathbf{x}_k)$;
Gauss-Newton
$r = \#\mathrm{columns}(A)$?
Yes
Output: $F; \mathbf{x}_{k+1}$.
No
$F := \mathbf{Dfl}(F)(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} F(\mathbf{x}) = \mathbf{0} \\ G(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \end{cases}$;
$\widehat{\boldsymbol{\lambda}} := \mathbf{LeastSquares}(G(\mathbf{x}_{k+1}, \boldsymbol{\lambda}))$;
$k := k + 1; \quad \mathbf{x}_k := (\mathbf{x}_k, \widehat{\boldsymbol{\lambda}})$;
Deflation Step

Figure 1: Flowchart for a modified deflation method.

A duality analysis of our method was presented in [6]. Like the analysis in [6] gives a better understanding on the number of needed deflations (establishing "depth" as a tighter bound), the shape of the standard basis (visualized by its staircase) leads to a more accurate bound.

3.1 Standard Bases for Local Orderings

Let $R = k[x_1, \ldots, x_n]$ be the ring of polynomials in n variables with coefficients in the field k. We use the following multi-degree notation: $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$, where $\alpha = (\alpha_1, \ldots, \alpha_n)$ is a vector of nonnegative integers.

A multiplicative ordering $\leq$ on the monoid $\{\mathbf{x}^\alpha \mid \alpha \in \mathbb{Z}^n_{\geq 0}\}$ is a *local ordering* if $\mathbf{x}^\alpha < 1$ for all $\alpha \neq (0, 0, \ldots, 0)$.

To any *weight vector* $\omega \in \mathbb{Z}^n_{<0}$ we may associate the ordering $\leq_\omega$ by setting

$$\mathbf{x}^\alpha \leq_\omega \mathbf{x}^\beta \Leftrightarrow \langle \alpha, \omega \rangle \leq \langle \beta, \omega \rangle, \tag{4}$$

where $\langle \cdot, \cdot \rangle$ is the usual inner product. Note that it is possible to have $\mathbf{x}^\alpha =_\omega \mathbf{x}^\beta$ (unless there are no integer vectors orthogonal to ω). In this case, the order is refined by, say, a lexicographic order.

In presence of a monomial ordering $\leq_\omega$, a polynomial

$$f(\mathbf{x}) = \sum_{\alpha \in \mathbb{Z}^n_{\geq 0}} c_\alpha \mathbf{x}^\alpha \in R, \text{ where } \operatorname{supp}(f) = \{ \ \alpha \mid c_\alpha \neq 0 \ \} \text{ is finite,} \tag{5}$$

has the following attributes associated with it:

$$\begin{aligned}
\mathbf{le}(f) &= \text{the leading exponent} = \max_{\leq_\omega} \operatorname{supp}(f) \\
\mathbf{lm}(f) &= \text{the leading monomial} = \mathbf{x}^{\mathbf{le}(f)} \\
\mathbf{lc}(f) &= \text{the leading coefficient} = c_{\mathbf{le}(f)} \\
\mathbf{lt}(f) &= \text{the leading term} = \mathbf{lc}(f)\mathbf{lm}(f)
\end{aligned}$$

Let $\mathcal{I} \subset R$ be an ideal. We call a set of polynomials $S \subset \mathcal{I}$ a *standard basis* of $\mathcal{I}$ if for any $f \in \mathcal{I}$ there is $g \in S$ such that $\mathbf{lm}(g) | \mathbf{lm}(f)$. Alternatively, S is a standard basis iff the *initial ideal* $\mathbf{in}(\mathcal{I}) = \langle \{\mathbf{lm}(f) | f \in \mathcal{I}\} \rangle$ is generated by the leading monomials $\mathbf{lm}(S) = \{\mathbf{lm}(g) | g \in S\}$.

The monomials that do not belong to the initial ideal $\mathbf{in}(\mathcal{I})$ are called *standard monomials*. The minimal generators of $\mathbf{in}(\mathcal{I})$ shall be called the *corners* of the staircase. The corners of the form x_i^a for some i and a are called the *endpoints* of the staircase.

A standard basis S is *reduced* if the leading monomials of its elements form a minimal generating set for the initial ideal $\mathbf{in}(\mathcal{I})$ and the tail $g - \mathbf{lt}(g)$ contains only standard monomials.

Graphically, any monomial ideal can be represented by a staircase in the nonnegative integer lattice $\mathbb{Z}^n_{\geq 0}$. For example, let $\mathcal{I}$ be the ideal of $R = k[x_1, x_2]$ generated by

$$\begin{aligned}
f_1 &= x_1^3 + x_1 x_2^2; \\
f_2 &= x_1 x_2^2 + x_2^3; \\
f_3 &= x_1^2 x_2 + x_1 x_2^2.
\end{aligned} \tag{6}$$

The initial ideal depends on the ordering chosen: the staircase at the left in Figure 2 represents $\mathbf{in}_w(\mathcal{I})$, where $\omega = (-1, -2)$. The staircase at the right in Figure 2 represents $\mathbf{in}_w(\mathcal{I})$, where $\omega = (-2, -1)$.

Observe in Figure 2 that the number of standard monomials is the same for both orderings. This is so for any local ordering, as the standard monomials form a basis of the k-linear space $R_{\langle x_1, \ldots, x_n \rangle} / R_{\langle x_1, \ldots, x_n \rangle} \mathcal{I}$, where $R_{\langle x_1, \ldots, x_n \rangle}$ is

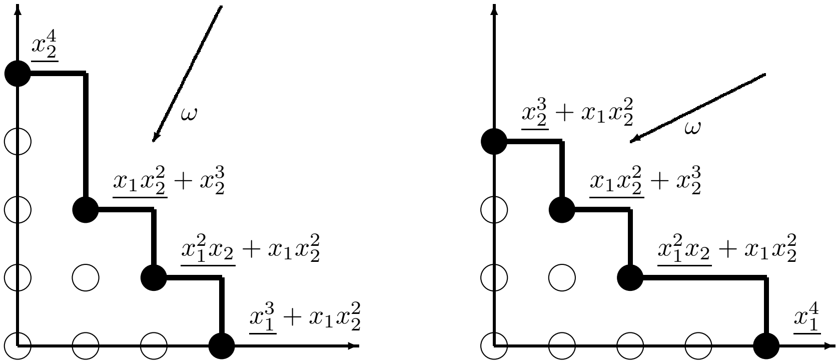
$\underline{x_2^4}$
ω
$\underline{x_1x_2^2} + x_2^3$
$\underline{x_1^2x_2} + x_1x_2^2$
$\underline{x_1^3} + x_1x_2^2$
$\underline{x_2^3} + x_1x_2^2$
ω
$\underline{x_1x_2^2} + x_2^3$
$\underline{x_1^2x_2} + x_1x_2^2$
$\underline{x_1^4}$

Figure 2: Two different staircases of the standard basis of $\mathcal{I}$ with respect to different local orderings $\leq_{(-1,-2)}$ (at the left) and $\leq_{(-2,-1)}$ (at the right). Monomials generating $\mathbf{in}(\mathcal{I})$ are represented by black disks, while the standard monomials are shown as empty circles.

the localization of the polynomial ring R at the origin and $R_{\langle x_1,\ldots,x_n\rangle}\mathcal{I}$ is the extension of the ideal $\mathcal{I}$ in this localized ring (see [5] for details). This linear space is of finite dimension iff the origin is an isolated solution; its dimension, which is the multiplicity of the origin, then equals the number of the standard monomials for any local ordering.

Thanks to Mora [41], there is an algorithm for computing standard bases, generalized by Greuel and Pfister [20], and implemented in the computer algebra system Singular [21]. We will not use standard bases explicitly – except for theoretical purposes – but note an analytic interpretation of the local ordering $\leq_w$: as we approach the origin along a smooth curve

$$c : \mathbb{C} \to \mathbb{C}^n \quad \text{such that} \quad c(t) = \begin{cases} b_1 t^{-\omega_1}(1 + O(t)) \\ \qquad\vdots \\ b_n t^{-\omega_n}(1 + O(t)) \end{cases} \tag{7}$$

with $\omega \in \mathbb{Z}^n_{<0}$ and $(b_1, ..., b_n) \in \mathbb{C}^n \setminus \{0\}$, for every $f \in \mathcal{I}$ the leading term $\mathbf{lt}_w(f)$ becomes dominant, i.e.:

$$f(c(t)) = \mathbf{lt}_\omega(f)(c(t)) + O(t^{-\langle \omega, \mathbf{le}(f)\rangle}). \tag{8}$$

3.2 Understanding the Deflation Method

First of all, let us formulate the goal of what we would call the *symbolic deflation* process: Given a system of polynomial equations $f_i = 0$, $i = 1, 2, \ldots, N$, with the point $\mathbf{x}^* \in \mathbb{C}^n$ as an isolated solution of multiplicity $m > 1$, find a system $g_i = 0$, $i = 1, 2, \ldots, N'$, such that $\mathbf{x}^*$ is still an isolated solution of multiplicity less than m.

The best deflation one can cook up is the one that corresponds to the maximal ideal annihilating x_i^*, i.e.: $g_i = x_i - x_i^*$, $i = 1, 2, \ldots, n$. However, from a practical angle of numerical methods what we actually need is an algorithm that would relate the deflated system to the original one in a numerically stable way and taking into account the fact that the isolated solution $\mathbf{x}^*$ may be known only approximately.

3.2.1 A Symbolic Deflation Method

Here we assume that everything is exact and, therefore, without loss of generality we may assume that the isolated solution $\mathbf{x}^*$ is the origin.

Consider the ideal $\mathcal{I}$ generated by the polynomials f_i of the original system. We call an ideal $\mathcal{I}'$ a *deflation* of $\mathcal{I}$ if $\mathcal{I}' \supset \mathcal{I}$, $\mathcal{I}' \neq R$, and the multiplicity of the origin for $\mathcal{I}'$ is lower than that for the original ideal $\mathcal{I}$.

If the multiplicity $m > 1$, it means that the initial ideal $\mathbf{in}(\mathcal{I})$ does not contain x_i for some i.

Proposition 3.2 *Suppose $m > 1$ and let g be an element of a reduced standard basis of $\mathcal{I}$ with respect to a local monomial ordering $\leq$, such that $\mathbf{lm}(g) = x_i^d$, for some $i \in \{1, ..., n\}$ and $d > 1$. Then the ideal $\mathcal{I}' = \mathcal{I} + \langle \partial g/\partial x_i \rangle$ is a deflation of $\mathcal{I}$.*

Proof. The derivative $\partial g/\partial x_i$ can not contain monomials $> x_i^{d-1}$. Therefore, $\mathcal{I}'$ contains $\mathcal{I}$ properly, since $\mathbf{lm}(\partial g/\partial x_i) = x_i^{d-1}$ is a standard monomial for $\mathcal{I}$. The appended generator $\partial g/\partial x_i$ still vanishes at the origin, hence, $\mathcal{I}' \neq R$. □

Mora's tangent cone algorithm [41] for computing standard bases is expensive symbolically and, more importantly, unstable numerically. Can we find x_i and g in the proposition in a less straightforward way? The next lemma gives a positive answer.

A linear coordinate change $T : \mathbb{C}^n \to \mathbb{C}^n$ induces an automorphism of the polynomial ring $R = \mathbb{C}[x_1, ..., x_n]$, which we call T as well: $T(f)(x) = f(T(x))$. The ideal $T(\mathcal{I}) = \{T(f) \mid f \in \mathcal{I}\} = \langle T(f_1), \ldots, T(f_N) \rangle$ represents the system after the change of coordinates.

Let $A(\mathbf{x})$ be the Jacobian matrix of the system $F(\mathbf{x}) = 0$, i.e.: an N-by-n matrix with polynomial entries $A_{ij}(\mathbf{x}) = \partial f_i/\partial x_j$. The origin is singular iff $c = \mathrm{corank}(A(\mathbf{0})) > 0$. Since the Jacobian matrix is rank-deficient, the kernel of $A(\mathbf{0})$ is nonzero.

Lemma 3.3 *Take a nonzero vector $\boldsymbol{\lambda} \in \ker A(\mathbf{0}) \subset \mathbb{C}^n$ and let $T : \mathbb{C}^n \to \mathbb{C}^n$ be a linear coordinate transformation such that:*

$$T_i(\mathbf{x}) = \lambda_i x_1 + \sum_{j=2}^{n} \mu_{ij} x_j, \qquad \text{for } i = 1, 2, \ldots, n, \tag{9}$$

where $[\boldsymbol{\lambda}, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n]$ is a nonsingular matrix.

Then $\partial_1(T(\mathcal{I})) = \{\frac{\partial}{\partial x_1} f \mid f \in T(\mathcal{I})\}$ is a deflation of $T(\mathcal{I})$.

Proof. For all $i = 1, 2, \ldots, N$,

$$\frac{\partial}{\partial x_1}(f_i(T(\mathbf{x}))) = \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}(T(\mathbf{x})) \cdot \frac{\partial T_j}{\partial x_1}(\mathbf{x}) = \sum_{j=1}^{n} \left(\frac{\partial f_i}{\partial x_j}(T(\mathbf{x})) \right) \lambda_j. \tag{10}$$

The last expression is equal to 0 when $\mathbf{x} = \mathbf{0}$, since $\boldsymbol{\lambda} \in \ker A(\mathbf{0})$.

Take any $g = b_1 T(f_1) + \cdots + b_N T(f_N) \in T(\mathcal{I})$, where $b_i \in R$ for all i. Then

$$\begin{aligned} \frac{\partial g}{\partial x_1} &= b_1 \frac{\partial (T(f_1))}{\partial x_1} + \cdots + b_N \frac{\partial (T(f_N))}{\partial x_1} \\ &+ T(f_1) \frac{\partial b_1}{\partial x_1} + \cdots + T(f_N) \frac{\partial b_N}{\partial x_1}. \end{aligned}$$

In view of (10), the last expression evaluates to 0 at $\mathbf{x} = \mathbf{0}$. Therefore, $\partial_1(T(\mathcal{I}))$ is a proper ideal annihilating the origin. On the other hand, there is an element g of a reduced standard basis of $T(\mathcal{I})$ with respect to a local ordering such that $\mathbf{in}(g) = x_1^d$ with $d > 1$. According to the Proposition 3.2 the ideal $\mathcal{I}' = T(\mathcal{I}) + \langle \partial g / \partial x_1 \rangle$ is a deflation of $T(\mathcal{I})$. So is $\partial_1(T(\mathcal{I}))$, for it contains $\mathcal{I}'$. □

Lemma 3.3 leads to Algorithm 1.

```
Algorithm 1 G = Symbolic_Deflation($F$)
Require: $F$, a finite set of polynomials in $R$, such that the ideal $\langle F \rangle$ has multiplicity $m > 1$ at the origin.
Ensure: $G$, a finite set of polynomials in $R$, such that the ideal $\langle G \rangle + \langle F \rangle$ is a deflation of $\langle F \rangle$.
  Compute the Jacobian $A$ of $F$ at the origin;
  Pick a nonzero vector $\boldsymbol{\lambda} \in \ker A(\mathbf{0})$;
  $G := \left\{ \sum_{i=1}^{n} \lambda_i \frac{\partial f}{\partial x_i} \mid f \in F \right\}$.
```

3.2.2 A Numeric Deflation Method

Our method formalized in Algorithm 2 is a numerical version of Algorithm 1. In this section we explain the transition from the symbolic to the numeric deflation method.

Consider a point $P = (\mathbf{x}, \boldsymbol{\lambda}) \in \mathbb{C}^{n, r+1}$ and let $\mathbf{x} = \mathbf{x}_0$. When this specialization is performed, the values for $\boldsymbol{\lambda}$ are determined by the following system of $N + 1$ linear equations:

$$\left\{ \begin{array}{rcl} A(\mathbf{x}_0)\boldsymbol{\lambda} & = & 0 \\ \langle \mathbf{h}, \boldsymbol{\lambda} \rangle & = & 1 \end{array} \right. \tag{11}$$

where $\mathbf{h} = (h_1, \ldots, h_{r+1})$ is a vector of random complex numbers.

Algorithm 2 G = Numeric_Deflation$(F, \mathbf{x}_0)$

```
Require: $F = \{f_1, \ldots, f_N\}$, a finite set of polynomials in $R$, such that the ideal
  $\langle F \rangle$ has multiplicity $m > 1$ at the point $\mathbf{x}^* \approx \mathbf{x}_0$.
Ensure: $G$, a finite set of polynomials in $R' = R[\lambda_1, \ldots, \lambda_{r+1}]$, where $r =$
  $\operatorname{rank} A(\mathbf{x}_0)$, such that
  • the ideal $\langle G \rangle \subset R'$ has an isolated solution at the point $P = (\mathbf{x}^*, \boldsymbol{\lambda}^*) \in \mathbb{C}^{n,r+1}$;
  • the vector $\boldsymbol{\lambda}^*$ is determined uniquely;
  • the multiplicity of $P$ is less than $m$.

  Compute the Jacobian matrix $A(\mathbf{x})$ of $F$;
  $r := \operatorname{rank} A(\mathbf{x}_0)$;   (numerical rank at the approximate solution $\mathbf{x}_0$)
  (R1) Generate a random matrix $B \in \mathbb{C}^{n\times(r+1)}$;
  $C(\mathbf{x}) := A(\mathbf{x})B$;   ($N \times (r+1)$ matrix with polynomial entries)
  Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{r+1})^T$ be a vector of indeterminates;
  Consider $N$ new polynomials $g_i(\mathbf{x}, \boldsymbol{\lambda}) = (C(\mathbf{x})\boldsymbol{\lambda})_i \in R'$;
  (R2) $h(\boldsymbol{\lambda}) := h_1\lambda_1 + \cdots + h_{r+1}\lambda_{r+1} - 1$, where the $h_i$ are random numbers
  in $\mathbb{C}$;
  $G := F \cup \{g_1, \ldots, g_N\} \cup \{\mathbf{h}\}$.
```

Observe how C is created: the randomization step **R1** insures that the $r+1$ columns of $C(\mathbf{x}^*)$ are random combinations of the columns of $A(\mathbf{x}^*)$ and, therefore, $\operatorname{corank} C(\mathbf{x}^*) = 1$ with probability one. Then $\boldsymbol{\lambda}$ is bound to live in the one-dimensional $\ker C(\mathbf{x}^*)$. The randomization step **R2** makes sure one nonzero vector is picked out from the kernel. This proves the uniqueness of $\boldsymbol{\lambda}^*$.

Assume that the original system is of corank 1 (we can always replace $F(\mathbf{x})$ with $F(\mathbf{x})B$, where B is as above). The correctness of the numeric deflation process relies on the following Proposition.

Proposition 3.4 *Let $\mathbf{x}^* \in \mathbb{C}^n$ be an isolated solution of $F(\mathbf{x}) = \mathbf{0}$ (in $\mathbb{C}[\mathbf{x}]$) and* $\operatorname{corank}(A(\mathbf{x}^*)) = 1$.

Consider the augmented system

$$G(\mathbf{x}, \boldsymbol{\lambda}) = (f_1, \ldots, f_N, g_1, \ldots, g_N, h)(\mathbf{x}, \boldsymbol{\lambda}) = 0,$$

with the new $N+1$ equations

$$g_i(\mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda} \cdot \nabla f_i(\mathbf{x}) = \sum_{j=1}^{n} \lambda_j \frac{\partial f_i(\mathbf{x})}{\partial x_j}, \ (i = 1, ..., N) \tag{12}$$

$$h(\boldsymbol{\lambda}) = \mathbf{h} \cdot \boldsymbol{\lambda} - 1 = \sum_{j=1}^{n} h_j \lambda_j - 1, \tag{13}$$

For a generic choice of coefficients $\mathbf{h} = (h_1, \ldots, h_{r+1})$, there exists a unique $\boldsymbol{\lambda}^* \in \mathbb{C}^n$ such that system $G(\mathbf{x}, \boldsymbol{\lambda})$ of equations in $\mathbb{C}[\mathbf{x}, \boldsymbol{\lambda}]$ has an isolated solution at $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$.

Moreover, the multiplicity of $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ *in* $G(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$ *is lower than that of* $\mathbf{x}^*$ *in* $F(\mathbf{x}) = \mathbf{0}$.

Proof. Consider $g_1 = \cdots = g_N = h = 0$ as a system of equations in the local ring $R_* = \mathbb{C}[\mathbf{x}, \boldsymbol{\lambda}]_{(\mathbf{x}^*, \boldsymbol{\lambda}^*)}$ that is linear in $\boldsymbol{\lambda}$.

The specialization of this system at $\mathbf{x} = \mathbf{x}^*$ makes it a linear system (with constant coefficients) of full rank with the unique solution: $\boldsymbol{\lambda}^*$. Therefore, using row operations in the ring R_* it is possible to reduce the system to the system of the form

$$\left\{ \begin{array}{rcl} \lambda_1 & = & a_1(\mathbf{x}), \\ & \vdots & \\ \lambda_n & = & a_n(\mathbf{x}), \end{array} \right. \tag{14}$$

where $a_i(\mathbf{x})$ are rational expressions. Note that $a_i(\mathbf{x}^*) = \lambda_i^*$.

Now we conclude that considering multiplicity of the augmented system $G(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$ with indeterminate $\boldsymbol{\lambda}$ in the ring R^* is equivalent to looking at the system $G(\mathbf{x}) = \mathbf{0}$ with fixed $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ in the local ring $\mathbb{C}[\mathbf{x}]_{\mathbf{x}^*}$.

Assuming $\mathbf{x}^*$ is the origin, the multiplicity drops by Lemma 3.3. □

Proof of Theorem 3.1. To show that at most $m-1$ deflation steps are needed to restore the quadratic convergence of an isolated root $\mathbf{x}^*$ of multiplicy m, it suffices to show that after one deflation step, the augmented system has the same root with its multiplicity decreased by at least one. This statement is shown by Proposition 3.4 in case the Jacobian matrix $A(\mathbf{x}^*)$ has corank one. To reduce to the general case, our deflation algorithm replaces $A(\mathbf{x})$ by $A(\mathbf{x})B$ using a random matrix B of $r+1$ columns, where $r = \text{rank}(A(\mathbf{x}^*))$. □

Remark 3.5 By bounding the number of deflations steps with multiplicity, Theorem 3.1 guarantees the termination of the deflation procedure. However, a stronger bound can be given: one may show that the number of deflations is equal to or less than the maximal total degree of monomials under the staircase for any local monomial ordering.

This statement can be rephrased in the language of dual bases of differential functionals; in this form it has been proved in [6]. The discussion of the correspondence between the two methods is beyond the scope of this paper.

In practice, finer information – staircases, dual bases, multiplicity structure – is not available at the time when the deflation is applied. The algorithm of [6] for computing the multiplicity structure, in fact, depends on the precision of the solution approximation, hence, on deflation.

4 A Symbolic-Numeric Implementation

The method was tested and developed in Maple 9. Since release 2.3 of PHCpack [56], the deflation algorithm is part of the validation (`phc -v`) module. In this section we briefly address symbolic-numeric issues.

4.1 Avoiding the Expression Swell

Although the symbolic implementation performs well for a couple of deflation steps, the multiplication of polynomial matrices by a random matrices leads to expression swell, amplified by the doubling of the size of the systems in each deflation stage.

Inspired by automatic differentiation [24, 50] and [30], we found that we should first evaluate the Jacobian matrices before multiplication with the random matrices. Furthermore, observing that the multiplier variables occur linearly in the block structure of the Jacobian matrices of the deflated systems, we presented in [32] a directed acyclic graph to evaluate the Jacobian matrices of the deflated systems efficiently. We refer to [32] for a detailed description of the efficient evaluation of the Jacobian matrices.

Another significant reduction of the expression swell lies in treating the corank one case separately, an issue we address in the Section 4.3 below.

4.2 A Posteriori Validation of the Numerical Rank

Our implementation defers the problem of the computation of the numerical rank to the established SVD and the recent techniques presented in [15] and [38].

The critical decision to deflate or not depends on the correct determination of the numerical rank of the Jacobian matrix. If we get the rank correctly, then after the deflation – with an accurate root – we obtain an a posteriori validation of all decisions made to determine the numerical rank. If we deflate with an incorrect rank, then there are two cases: either our numerical rank is too low or too high. The first case of a too low numerical rank will be detected early as the algorithm will then produce too many additional constrains. In that case, the calculation in Figure 1 of the initial values $\widehat{\boldsymbol{\lambda}}$ for the multipliers via the least squares problem is already likely to fail and an interactive application of the method may backtrack. In the second case, when the numerical rank is too high, one may not deflate at all in case of full rank and continue applying Newton's method until one is close enough for the threshold to be crossed. A deflation with a too small corank may be remediated by an extra deflation step, although we have no practical experience with this case.

4.3 Special Case Implementations and Extensions

The case where the corank of the Jacobian matrix equals one occurs frequently and can be treated more efficiently than the general case. In [6] propose a modification of the deflation algorithm for the important case of corank one.

In the corank one case, there is no need to multiply the Jacobian matrix with a random matrix. Moreover, as pointed out in [6], subsequent deflations should only concern the original set of equations. Deflating a system of N equations in n unknowns m times then leads to a system of mM equations in mn unknowns as shown in [6].

Another extension, exemplified in [6], concerns the application of our deflation method to analytic systems.

5 Applications and Numerical Results

The implementation has been tested on several examples, available at `http://www.math.uic.edu/~jan/demo.html`, mostly all obtained from surveying the literature. The initial approximations for Newton's method were taken from the end points of solution paths defined by a polynomial homotopy to find all isolated solutions (see [33, 34] for recent surveys). The numerical results reported in Table 1 below are obtained with standard machine arithmetic[3]. We highlight three examples from our benchmark collection.

A simple monomial ideal. Consider the simple polynomial system

$$f(x,y) = \left\{ \begin{array}{l} x^2 = 0 \\ xy = 0 \\ y^2 = 0. \end{array} \right. \qquad (15)$$

Viewing (15) as a monomial ideal, we immediately read off the multiplicity as 3. However, as explained in [51] and [52], this system presents a challenge to numerical solvers: making this overdetermined system square either by adding a random multiple of the last equation to the first two (and then removing the third equation), or by adding one slack variable, increases the multiplicity from three to four. As our deflation departs from Gauss-Newton, only one deflation step is needed to restore the quadratic convergence. Table 1 opens with a summary of these calculations.

The 4-fold cyclic 9-roots. The so-called cyclic 9-roots problem (labelled as `cyclic9` in Table 1) is one of our largest examples. This system is a widely used benchmark in the field of polynomial system solving, e.g.: [2, 3], [14], [16], [25], [35], with theoretical results in [26]. In addition to six two dimensional cubics, there are 5,594 (333 orbits of size 18) isolated regular cyclic 9-roots, and most interestingly for this paper: 162 isolated solutions of multiplicity four. One deflation suffices to restore quadratic convergence on all 162 quadruple roots of this large application.

Lines tangent to a special configuration of 4 spheres. Given four spheres, how many real lines are tangent to all four spheres? A clue to the answer (see the Maple supplements to [53]) is obtained by placing the four

[3] In case the multiple root is the origin, the number of correct digits in the last column of Table 1 equals the negative exponent of ten in the magnitude of the root, i.e.: "24" refers to 10^{-24} as the magnitude of the root. This is explains why the accuracy is higher than what can be achieved from double precision floating-point arithmetic.

spheres so they mutually touch each other. There are three distinct lines connecting the points where the spheres touch each other. Solving the corresponding algebraic system [13] shows the multiplicity of each line to be four, revealing twelve as the answer for this problem. One deflation suffices to compute all solutions accurately to the full machine precision, even with 16-digit approximations for the algebraic numbers $\sqrt{3}$ and $\sqrt{6}$ appearing as coefficients in the system. Computational results are in the last line of Table 1.

These three examples illustrate the motivation of our deflation algorithm: we present a method, designed to handle any kind of general isolated singular solutions of polynomial systems, efficient enough for large problems, and accepting approximate input coefficients.

System	n	m	D	corank($A(\mathbf{x}^*)$)	Inverse Condition#	#Digits
simple [51]	2	3	1	2 → 0	1.0e-08 → 4.1e-01	8 → 24
baker1[27]	2	2	1	1 → 0	1.7e-08 → 3.8e-01	9 → 24
cbms1[55]	3	11	1	3 → 0	4.2e-05 → 5.0e-01	5 → 20
cbms2[55]	3	8	1	3 → 0	1.2e-08 → 5.0e-01	8 → 18
mth191	3	4	1	2 → 0	1.3e-08 → 3.5e-02	7 → 13
decker1[9]	2	3	2	1 → 1 → 0	3.4e-10 → 2.6e-02	6 → 11
decker2[7]	2	4	3	1 → 1 → 1 → 0	4.5e-13 → 6.9e-03	5 → 16
decker3[8]	2	2	1	1 → 0	4.6e-08 → 2.5e-02	8 → 17
kss3[28]	10	638	1	9 → 0	4.4e-12 → 1.4e-02	7 → 16
ojika1[45]	2	3	2	1 → 1 → 0	9.3e-12 → 4.3e-02	5 → 12
ojika2[45]	3	2	1	1 → 0	3.3e-08 → 7.4e-02	6 → 14
ojika3[48]	3	2	1	1 → 0	1.7e-08 → 9.2e-03	7 → 15
		4	1	2 → 0	6.5e-08 → 8.0e-02	6 → 13
ojika4[47]	3	3	2	1 → 1 → 0	1.9e-13 → 2.4e-04	6 → 11
caprasse [43]	4	4	1	2 → 0	1.5e-09 → 9.3e-03	8 → 15
cyclic9 [2, 3]	9	4	1	2 → 0	5.6e-10 → 1.8e-03	5 → 15
tangents [53]	6	4	1	2 → 0	2.6e-08 → 2.4e-02	7 → 14

Table 1: Numerical Results on a Collection of Test Systems. The dimension is listed under n, m is the multiplicity, and D is the number of deflations needed to restore quadratic convergence. The fifth column shows the decrease in the corank of the Jacobian matrix for all stages in the deflation. The second to last column contains the estimate for the inverse condition number of $A(\mathbf{x})$ at the start of the deflation to the end of the deflation for $\mathbf{x} \approx \mathbf{x}^*$. The last column lists the increase in the number of correct digits from the initial guess to the final approximation.

One of the interesting examples is taken from [48] and listed as "ojika3" in Table 1. This system has two isolated roots: one with multiplicity two, and the other one has multiplicity four. Both roots need only one deflation, but at the double root, the rank of the Jacobian matrix is two, while the rank is one at the other 4-tuple root. The program produces two different deflated systems: one with three multipliers (for the double root) and the other with two multipliers (for the 4-tuple root).

The high multiplicity 638 of one root of the system `kss3` in Table 1 looks spectacular, but since the defining equations are nice quadrics, one simple deflation suffices to compute the multiple root accurately, starting from any approximate root in the cluster of 638 solutions.

Observe the improved numerical conditioning in Table 1. This observation justifies the naming[4] of our method as a "re-conditioning" method.

6 Conclusions

Our modified deflation method works in general, is numerically stable, relatively simple to implement; and perhaps most importantly, a preliminary implementation on a wide class of examples performs quite well.

The doubling of the number of equations by the deflation has been addressed in [32] and [6] for the corank one case. Our method is numerically robust, depending primarily on a reliable determination of the numerical rank of a matrix, a well studied subject in numerical linear algebra (see e.g. [15] or [38]). Nevertheless, an alpha theoretic certificate [4] of the numerical rank might be desirable for a fully automatic computer implementation.

References

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

[2] G. Björck 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.

[3] G. Björck 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.

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

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

[4] We are grateful to Erich Kaltofen for this naming at the ISSAC'04 poster session.

[6] 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.

[7] D.W. Decker, H.B. Keller, and C.T. Kelley. Convergence rates for Newton's method at singular points. *SIAM J. Numer. Anal.*, 20(2):296–314, 1983.

[8] D.W. Decker, H.B. Keller, and C.T. Kelley. Broyden's method for a class of problems having singular Jacobian at the root. *SIAM J. Numer. Anal.*, 22:566–574, 1985.

[9] D.W. Decker and C.T. Kelley. Newton's method at singular points: II. *SIAM J. Numer. Anal.*, 17(3):465–471, 1980.

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

[11] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms.* Springer-Verlag, 2004.

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

[13] C. Durand. *Symbolic and Numerical Techniques for Constraints Solving.* PhD thesis, Purdue University, Department of Computer Science, 1998.

[14] Faugère, J.C. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999. Proceedings of MEGA'98, 22–27 June 1998, Saint-Malo, France.

[15] R.D. Fierro and P.C. Hansen. Utv expansion pack: Special-purpose rank-revealing algorithms. *Numerical Algorithms*, 40(1):47–66, 2005.

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

[17] M. Giusti, G. Lecerf, B. Salvy, and J.C. Yakoubsohn. On location and approximation of clusters of zeroes: case of embedding dimension one. Manuscript, 2004.

[18] M. Giusti, G. Lecerf, B. Salvy, and J.C. Yakoubsohn. On location and approximation of clusters of zeroes of analytic functions. Manuscript, 2004.

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

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

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

[22] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 2.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2001. `http://www.singular.uni-kl.de`.

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

[24] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM, 2000.

[25] T. Gunji, S. Kim, M. Kojima, A. Takeda, K. Fujisawa, and T. Mizutani. PHoM – a polyhedral homotopy continuation method for polynomial systems. *Computing*, 73(1):57–77, 2004.

[26] U. Haagerup. Orthogonal maximal abelian *-algebras of the n x n matrices and cyclic n-roots. In *Operator Algebras and Quantum Field Theory*, pages 296–322. International Press, 1996.

[27] R.B. Kearfott and J. Dian. Existence verification for higher degree singular zeros of nonlinear systems. *SIAM J. Number. Anal.*, 41(6):2350–2373, 2003.

[28] H. Kobayashi, H. Suzuki, and Y. Sakai. Numerical calculation of the multiplicity of a solution to algebraic equations. *Math. Comp.*, 67(221):257–270, 1998.

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

[30] 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.

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

[32] A. Leykin, J. Verschelde, and A. Zhao. Evaluation of Jacobian matrices for Newton's method with deflation to approximate isolated singular solutions of polynomial systems. In D. Wang and L. Zhi, editors, *SNC 2005 Proceedings. International Workshop on Symbolic-Numeric Computation. Xi'an, China, July 19-21, 2005*, pages 19–28, 2005.

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

[34] 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.

[35] T.Y. Li and X. Li. Finding mixed cells in the mixed volume computation. *Found. Comput. Math.*, 1(2):161–181, 2001.

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

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

[38] T.Y. Li and Z. Zeng. A rank-revealing method with updating, downdating and applications. *SIAM J. Matrix Anal. Appl.*, 26(4):918–946, 2005.

[39] F.S. Macaulay. *The Algebraic Theory of Modular Systems.* Cambridge University Press, 1916. Reissued with an Introduction by Paul Roberts in the Cambridge Mathematical Library 1994.

[40] M.G. Marinari, H.M. Moeller, and T. Mora. On multiplicities in polynomial system solving. *Trans. AMS*, 348(8):3283–3321, 1996.

[41] 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.

[42] T. Mora. Gröbner duality and multiple points in linearly general position. *Proceedings of the AMS*, 125(5):1273–1282, 1997.

[43] S. Moritsugu and K. Kuriyama. On multiple zeros of systems of algebraic equations. In S. Dooley, editor, *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (ISSAC 1999)*, pages 23–30. ACM, 1999.

[44] B. Mourrain. Isolated points, duality and residues. *Journal of Pure and Applied Algebra*, 117/118:469–493, 1997.

[45] 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.

[46] 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.

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

[48] 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.

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

[50] L.B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science*. Springer-Verlag, 1981.

[51] 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.

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

[53] F. Sottile and Th. Theobald. Lines tangents to $2n-2$ spheres in $\mathbb{R}^n$. *Trans. Amer. Math. Soc.*, 354:4815–4829, 2002. http://www.math.tamu.edu/∼sottile/pages/spheres/index.html offers a nice computer algebra supplement.

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

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

[56] 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`.

[57] J. Verschelde and A. Zhao. Newton's method with deflation for isolated singularities. Poster presented at ISSAC'04, 6 July 2004, Santander, Spain. Available at `http://www.math.uic.edu/~jan/poster.pdf` and at `http://www.math.uic.edu/~azhao1/poster.pdf`.

[58] T.Y. Ypma. Historical development of the Newton-Rapson method. *SIAM Review*, 37(4):531–551, 1995.