Approximate Local Rings and Local Solution of Nonlinear Systems

Preliminary Report*

Barry H. Dayton
Department of Mathematics,
Northeastern Illinois University,
Chicago, IL 60625
bhdayton@neiu.edu

Revised July 24, 2006

In a recent paper with Zhonggang Zeng [2], we showed how to compute the structure of the local ring of an isolated zero from an approximate estimate of the zero, adapting a method of Macaulay (1915) by using approximate rank revealing. This can be formalized by the concept of an approximate local ring. By further relaxing the tolerance, information can be obtained on a Euclidean neighborhood of the point. The existence of nearby points can be deduced and nearby points can be calculated by the eigenvalue method. Several examples are given.

1 Introduction

In 1915 Macaulay [7] introduced dialytic and inverse arrays which together implicitly described the structure of the local ring in the case of an isolated solution of a system of polynomial equations. Of course, this predated the actual introduction of local rings by Chevalley in 1943. Macaulay's arrays are reproduced by T. Mora, for example, in [9], see Figure 1 below.

Ninety years after Macaulay, Zhonggang Zeng and I [2] used similar arrays to find local information about an isolated zero given an approximate location of the zero and/or approximate equations. We called our arrays the *multiplicity* and *dual* matrices as in Figure 2 below.

In particular, in [2] the multiplicity of the zero was shown to be the approxi-dimension of the dual space, shown here as the number of rows of the dual matrix. Due to space limitations we were not able to include the graphics, in Figure 3, which show the calculated multiplicity

*Presented to the special session on Numerical Solution of Polynomial Systems at the AMS Notre Dame Meeting, April 8-9, 2006

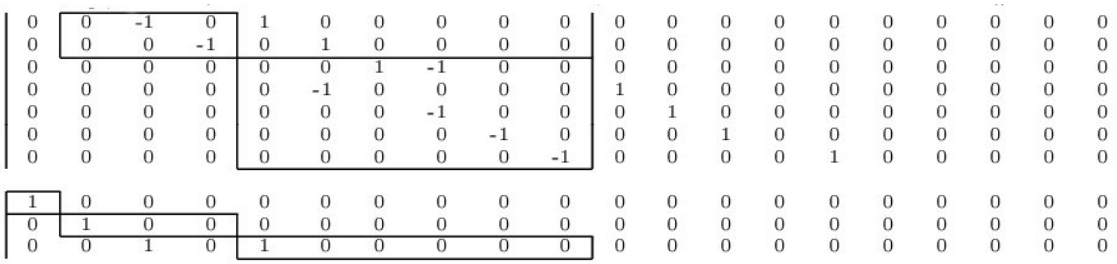


Figure 1: Macaulay's Inverse and Dialytic Arrays

multiplicity		$\|\mathbf{j}\|=0$	$\|\mathbf{j}\|=1$		$\|\mathbf{j}\|=2$			$\|\mathbf{j}\|=3$			
matrices ↘		∂_{00}	∂_{10}	∂_{01}	∂_{20}	∂_{11}	∂_{02}	∂_{30}	∂_{21}	∂_{12}	∂_{03}
	f_1	0	1	−1	1	0	0	0	0	0	0
S_1	f_2	0	1	−1	0	0	1	0	0	0	0
	x_1f_1	0	0	0	1	−1	0	1	0	0	0
	x_1f_2	0	0	0	1	−1	0	0	0	1	0
	x_2f_1	0	0	0	0	1	−1	0	1	0	0
S_2	x_2f_2	0	0	0	0	1	−1	0	0	0	1
	$x_1^2f_1$	0	0	0	0	0	0	1	−1	0	0
	$x_1^2f_2$	0	0	0	0	0	0	1	−1	0	0
	$x_1x_2f_1$	0	0	0	0	0	0	0	1	−1	0
	$x_1x_2f_2$	0	0	0	0	0	0	0	1	−1	0
	$x_2^2f_1$	0	0	0	0	0	0	0	0	1	−1
S_3	$x_2^2f_2$	0	0	0	0	0	0	0	0	1	−1

	bases for nullspaces (transposed as row vectors)									
$\mathcal{N}(S_0)$	1	0	0	0	0	0	0	0	0	0
$\mathcal{N}(S_1)$	0	1	1	0	0	0	0	0	0	0
$\mathcal{N}(S_2)$	0	−1	0	1	1	1	0	0	0	0
$\mathcal{N}(S_3)$										

Figure 2: The Multiplicity and Dual Matrices

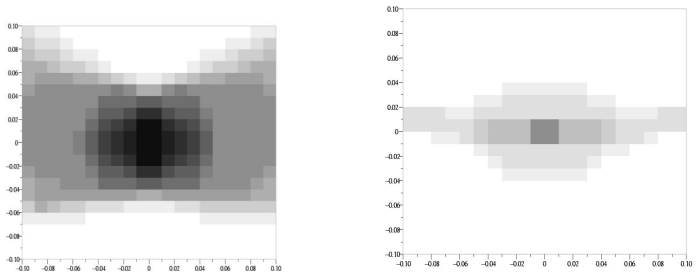


Figure 3: Loose Tolerance (left) and Tight Tolerance (right) for well isolated zero.

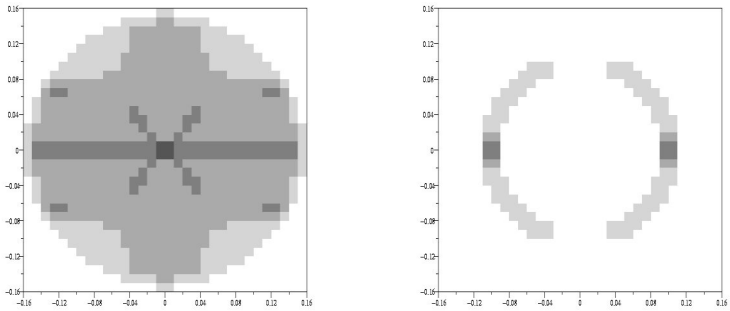


Figure 4: Loose (left) and tight (right) tolerance for two near zeros

for various approximations of the zero. For these pictures, generated by SCILAB, I calculated the Multiplicity matrix and its column rank deficiency for approximately 400 values of an approximate zero near an exact real zero of a two dimensional system. The darker colors represent higher multiplicity, so in the darkest neighborhood around the exact zero, calculating the Multiplicity matrices at a point here gives the correct multiplicity and other local information. In the next darker neighborhood if the approximation to the zero is located here the method indicates a multiple point but with lower multiplicity. And so on, until the white area where the point used to calculate the Multiplicity matrices is determined to not even be an approximate zero. The size of these *critical* neighborhoods depends on a tolerance, with a loose tolerance they were large, but with a tight tolerance they were smaller.

Thus our method worked reasonably well when the zero was well isolated, i.e. there was a large neighborhood of the point with only this one zero. However it appeared that an annoying problem with our method arose when there where distinct but close isolated zeros. In this case, for some choices of tolerance, an approximate zero might lie in the critical neighborhood of two or more of these points. In this case we would get strange, and often unexplained large, multiplitiicy values. Figure 4 shows an example with two close zeros.

The ALR method of this paper can be used to make sense of this loose tolerance case by separating out the various near zeros. Several examples are given, in particular I show, Example

6.2, that by starting LVZ deflation at a suspected multiple zero in a cluster one can get better results than by aiming deflation at a cluster as a whole. In Example 6.3 I suggest that this method may be quite useful in analyzing analytic systems where there may not be a method for finding all solutions.

In the theoretical, i.e. exact, case I show in §3 how to constructively obtain the local ring of an isolated zero, represented as an explicit matrix algebra, from the dialytic or multiplicity array using only one application of the reduced row echelon form algorithm. In §4 by using approximate linear algebra, specifically the *approximate reverse reduced row echelon form* (ARRREF), to obtain an *approximate local ring* (ALR). With accurate numerical data derived from an exact system and a tight tolerance this gives essentially the exact local ring. However by relaxing the tolerance one gets a structure which describes not just the isolated zero but a neighborhood of the isolated zero. In particular, there may be more than one point associated with the ALR, and these are readily calculated using the Möller-Stetter method.

It should be mentioned here that my ALR method works only for zero dimensional points, i.e. is useful when only isolated zeros are nearby. At this point I am not able to get any useful information when positive dimensional components of the solution space are present.

2 The Multiplicity Matrix

Z. Zeng and I describe the multiplicity matrix in [2]. For the reader's convenience I will summarize that construction here. Let $f_1, f_2, \ldots, f_t$ be a list of t equations in s variables $x_1, x_2, \ldots, x_s$. To get isolated solutions it is necessary for $t \geq s$. By equation I mean polynomial or analytic functions, that is functions with partial derivatives of all orders and that are, at least, locally approximated by polynomials. Let $\hat{\mathbf{x}}$ be a point in $\mathbb{C}^s$. For a non-negative integer array $\mathbf{j} = [j_1, \ldots, j_s]$ I write $|\mathbf{j}| = j_1 + j_2 + \cdots + j_s$ and $\mathbf{x}^{\mathbf{j}} = x_1^{j_1} x_2^{j_2} \ldots x_s^{j_s}$. So $\mathbf{x}^{\mathbf{j}}$ has total degree $|\mathbf{j}|$. Assume a (local) degree ordering is given on the monomials. As in [2] I use the differentiation operator

$$\partial_{\mathbf{x}^{\mathbf{j}}} \equiv \partial_{x_1^{j_1} \cdots x_s^{j_s}} \equiv \frac{1}{j_1! \cdots j_s!} \frac{\partial^{j_1+\cdots+j_s}}{\partial x_1^{j_1} \cdots \partial x_s^{j_s}}.$$

I write $\partial_{\mathbf{x}^j}(g)(\hat{\mathbf{x}})$ to indicate that I am taking the partial derivative of function g and evaluating at $\hat{\mathbf{x}}$. Note that $\partial_1(g)(\hat{\mathbf{x}}) = \partial_{\mathbf{x}^{\mathbf{0}}}(g)(\hat{\mathbf{x}})$ is simply evaluation of g at $\hat{\mathbf{x}}$, i.e. $g(\hat{\mathbf{x}})$.

The *multiplicity matrix of degree k at $\hat{\mathbf{x}}$*, $\mathbf{M}(k, \hat{\mathbf{x}})$ is the $t\binom{k+s-1}{s} \times \binom{k+s}{s}$ matrix with columns indexed by the differentials $\partial_{\mathbf{x}^{\mathbf{j}}}$ for $\mathbf{j} \leq k$ or, if I am being sloppy, just by the $\mathbf{x}^{\mathbf{j}}$, where I will assume the columns indexed by $\mathbf{x}^{\mathbf{j}}$ of higher degree are to the right of those with lower degree. In particular the left hand column has index $\partial_1 = \partial_{x_1^0 \ldots x_s^0}$, the evaluation functional. The rows will be indexed by the functions $\mathbf{x}^{\mathbf{i}} f_\alpha$ for $|\mathbf{i}| < k$, $\alpha = 1, \ldots, t$. Again these will be grouped by $|\mathbf{i}|$ with smaller degree higher up. In particular the first t rows are indexed by $f_1, \ldots, f_t$. Whether within a degree you group by $\mathbf{x}^{\mathbf{i}}$ or by f_α is up to you.

The entry in the row indexed by $\mathbf{x}^{\mathbf{i}} f_\alpha$ and column indexed by $\mathbf{x}^{\mathbf{j}}$ is $\partial_{\mathbf{x}^{\mathbf{j}}}((\mathbf{x}^{\mathbf{i}} - \hat{\mathbf{x}}^{\mathbf{i}}) f_\alpha)(\hat{\mathbf{x}})$

See Figure 2 for an example when $s = t = 2, f_1 = x_1 - x_2 + x_1^2, f_2 = x_1 - x_2 + x_2^2$ at $\hat{\mathbf{x}} = (0, 0)$.

If you would rather use instead $\partial_{\mathbf{x^i}}(\mathbf{x^i} f_\alpha)(\hat{\mathbf{x}})$ for the entry that is fine by me also, your rows will just be linear combinations of the rows of my multiplicity matrix. But you may loose the block structure of the matrix, that is the zeros in the lower left. I don't use that structure in this paper except to assert existence of the block structure in Lemma 1 below and I have not noticed any difference in the numerical results.

If the $f_1, \ldots, f_t$ are exact polynomial equations and $\hat{\mathbf{x}}$ is an exact isolated zero then the nullity $H_0(k, \hat{\mathbf{x}})$ of $\mathbf{M}(k, \hat{\mathbf{x}})$, i.e. the dimension of the nullspace, as k increases will increase and stabilize at $k = m$ where m is the multiplicity at $\hat{\mathbf{x}}$. This was known to Macaulay and later algebraic geometers but anyway a proof is sketched in [2, Theorem 2]. In fact $H_0(k, \hat{\mathbf{x}})$ will increase in a very structured way as outlined in [2, Lemma 1 (iii),(iv)], but note that $H_0(k, \hat{\mathbf{x}}) = \sum_{\alpha=0}^{k} H(\alpha)$ where the latter $H(\alpha)$ is the Hilbert function of [2].

In the absence of polynomials and/or exactness for the f_α or $\hat{\mathbf{x}}$ then we will pick a tolerance ε and calculate $H_\varepsilon(k, \hat{\mathbf{x}})$ as the dimension of the approxi-nullspace of $\mathbf{M}(k, \hat{\mathbf{x}})$ with tolerance ε. In §5 of [2] there is a short discussion of approxi-nullspace, for full details see [6]. The approxi-nullspace of a matrix A with tolerance ε is essentially the nullspace of the matrix obtained by replacing the singular values of A which are less than ε by 0. Or, in other words, $H_\varepsilon(k, \hat{\mathbf{x}})$ is the number of columns of $\mathbf{M}(k, \hat{\mathbf{x}})$ minus the number of singular values greater than ε. This justifies using the notation $\varepsilon = 0$ to refer to the exact case.

The sequence $\{H_\varepsilon(k, \hat{\mathbf{x}})\}$ may be less well behaved than its exact counterpart, but as long as $\hat{\mathbf{x}}$ is far from a positive dimensional component of the solution set of $f_1 = 0, \ldots, f_t = 0$ then $H_\varepsilon(k, \hat{\mathbf{x}})$ should, but is not guarenteed to, again stabilize at some value m. If there exists integer d so that $H_\varepsilon(k, \hat{\mathbf{x}}) = m$ for all $k \geq d$ then I say that any $k > d$ is in the *stable range for* m.

Heuristically an m that has a stable range is the sum of the multiplicities of zeros of the system "near" $\hat{\mathbf{x}}$, but at this point I do not have a good idea as to what "near" actually is. In the next section I will show how one can attempt to find these m, not necessarily distinct, points.

3 Representation of the Local Rings in the Exact Case

As mentioned in the introduction, Macualay's dialytic arrays and our multiplication matrices give a computational way to get information normally associated with the local ring. It is a curious fact, to my knowledge so far unreported, that, in the theoretical, exact, case one can actually get a concrete representation of the local ring of an isolated zero as a matrix algebra from these matrices by using only one application of the RREF, reduced row echelon form, algorithm.

Consistent with the rest of this paper I work over $\mathbb{C}$, however the results of this section should work over any field, with the understanding that $\hat{\mathbf{x}}$ is an isolated zero means the local ring is a finite dimensional algebra algebra over that field. It is a very classical result, eg. [1], that any finite dimensional $\mathbb{C}$-algebra of $\mathbb{C}$-dimension m is isomorphic to an $m \times m$ matrix algebra. This isomorphism is given by assigning an element f to the multiplication matrix A_f, see eg. [1] or [10, Chap. 2]. I now show how to calculate the multiplication matrices A_{x_i}.

Suppose we have polynomial equations $f_1, \ldots, f_t \in \mathbb{C}[x_1, \ldots, x_s]$. For the rest of this section we will assume, for simplicity, that $\hat{\mathbf{x}} = \mathbf{0}$ is an exact isolated zero of the system. Let $T = [1, x_1, \ldots, x_s^k]$ be the vector of column indices for $\mathbf{M}(k, \hat{\mathbf{x}})$. Then I will view $F_k = \mathbf{M}(k, \hat{\mathbf{x}})T^\top$ as a new system of equations. These equations are just truncations of the equations $\mathbf{x}^\mathbf{i} f_\alpha$ with the terms of degree greater than k set to zero. If $\hat{\mathbf{x}}$ were not an exact zero F_k might be inconsistent, but if $\hat{\mathbf{x}}$ is an exact zero then at least $\hat{\mathbf{x}}$ is a zero. Note that if I had not assumed $\hat{\mathbf{x}} = \mathbf{0}$ then the equations would be Taylor expansions of $(\mathbf{x} - \hat{\mathbf{x}})^\mathbf{i} f_\alpha$ about $\hat{\mathbf{x}}$.

In the exact case the stable range must exist for an isolated zero $\hat{\mathbf{x}} = \mathbf{0}$ and the integer d in the definition of the stable range is the smallest integer d for which $M^{d+1}/M^{d+2} = \mathbf{0}$ where M is the maximal ideal of the local ring at $\hat{\mathbf{x}}$. See, for example [2] for a discussion of this fact. By Nakayama's lemma then actually $M^{d+1} = \mathbf{0}$ in the local ring. We used the unfortunate term, from a commutative algebra standpoint, *depth* in [2] for d, this has also gone under the name *nil-index* in the literature.

Lemma 1 *Let $\hat{\mathbf{x}} = \mathbf{0}$ be an exact isolated zero of the exact polynomial system $f_1, \ldots, f_t$ in $\mathbb{C}[\mathbf{x}] = \mathbb{C}[x_1, \ldots, x_s]$. Choose d as in the pararaph above. Then for $k > d$*

i) *The last $\binom{k+s-1}{k}$ columns of $\mathbf{M}(k, \hat{\mathbf{x}})$ are independent.*

ii) *The equations in the system $F_k = \mathbf{M}(k, \hat{\mathbf{x}})T^\top$ are actual elements of the ideal $\mathbb{C}[[\mathbf{x}]]\langle f_1, \ldots, f_t\rangle \subseteq \mathbb{C}[[\mathbf{x}]]$.*

Proof: i) The block structure used in my definition of $\mathbf{M}(k, \hat{\mathbf{x}})$ gives

$$\mathbf{M}(k, \hat{\mathbf{x}}) = \left[\begin{array}{c|c} \mathbf{M}(k-1, \hat{\mathbf{x}}) & A \\ \hline \mathbf{0} & B \end{array}\right]$$

where A, B have $\binom{k+s-1}{k}$ columns. Since the coranks, i.e. dimensions, of the nullspace of $\mathbf{M}(k-1, \hat{\mathbf{x}})$ and $\mathbf{M}(k, \hat{\mathbf{x}})$ are the same it follows that the matrix $\left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$ has independent columns.

ii) By the paragraph before the lemma $M^k = 0$ so the truncated terms of the $\mathbf{x}^\mathbf{i} f_\alpha$ were zero anyway. ∎

Armed with this lemma, given k in the stable range, preferably $k = d + 1$, we can apply the reverse reduced row echelon form algorithm, RRREF, to $\mathbf{M}(k, \hat{\mathbf{x}})$. By reverse I mean that we start pivoting at the lower right. Alternatively, we can physically rotate $\mathbf{M}(k, \hat{\mathbf{x}})$ 180°, apply the usual RREF algorithm, and then rotate back. One obtains

$$R = \left[\begin{array}{c|c} \cdots & \mathbf{0} \\ \hline \cdots & I \end{array}\right]$$

where I is a square identity matrix of size $\binom{k+s-1}{k}$, and R is row equivalent to $\mathbf{M}(k, \hat{\mathbf{x}})$. The m monomials $\mathbf{x}^{\beta_1}, \ldots, \mathbf{x}^{\beta_m}$, m is the multiplicity of zero $\hat{\mathbf{x}}$, corresponding to the non-pivot columns of R form a $\mathbb{C}$-basis for the algebra and the equations of $RT^\top$ are then of the form

$$c_1\mathbf{x}^{\beta_1} + \cdots + c_m\mathbf{x}^{\beta_m} + \mathbf{x}^\mathbf{j}$$

for some index **j**. Since all **j** of degree k are present,it is always possible to construct the multiplication matrices A_{x_i} which generate the matrix algebra.

Example 3.1 Consider the system $x_1 - x_2 + x_1^2, x_1 - x_2 + x_2^2$ with exact zero of multiplicity 3 at $\hat{\mathbf{x}} = (0,0)$. It is seen that $d = 2$ so $k = 3$ is in the stable range. The multiplicity matrix $\mathbf{M}(3, \hat{\mathbf{x}})$ has already been given in Figure 2. Applying RRREF to $\mathbf{M}(3, \hat{\mathbf{x}})$ I obtain

$$R = \begin{bmatrix} 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

One checks that the local ring has basis $\{I_3, A_{x_1}, A_{x_2}\}$ where I_3 is the 3×3 identity matrix and

$$A_{x_1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \end{bmatrix}, \quad A_{x_2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \end{bmatrix}$$

It is then easy to completely specify the multiplicative structure of this ring with equations $A_{x_1}^2 = A_{x_1}A_{x_2} = A_{x_2}^2 = A_{x_2} - A_{x_1}$ which can be checked by matrix multiplication or induced from R. Moreover one can check, directly or from R, $A_{x_1}^3 = A_{x_1}^2 A_{x_2} = A_{x_1} A_{x_2}^2 = A_{x_2}^3 = 0$ so $M^3 = \mathbf{0}$ where $M = \langle A_{x_1}, A_{x_2} \rangle$ is the maximal ideal. ∎

In summary, suppose the $\mathbb{C}$-basis is $A_{g_1}, \ldots, A_{g_m}$. The following identities must be satisfied:

$$A_{x_i}A_{x_j} - A_{x_j}A_{x_i} = \mathbf{0}, \qquad \text{all } i, j \tag{1}$$

$$A_{x_1}^{j_1} A_{x_2}^{j_2} \cdots A_{x_s}^{j_s} = \sum_{i=1}^{m} c_i A_{g_i} \qquad j_\alpha \geq 0, \text{ some } c_i \in \mathbb{C} \tag{2}$$

$$f_\alpha(A_{x_1}, \ldots, A_{x_s}) = 0, \qquad \alpha = 1, \ldots, t \tag{3}$$

$$\lambda_{i,1} = \cdots = \lambda_{i,m} = 0, \qquad \lambda_{i,j} \text{ eigenvalues of} A_{x_i} \tag{4}$$

It should be noted that (4) follows from the fact the $A_{x_i}^d = \mathbf{0}$ where d is the nil-index. Note also if $\hat{\mathbf{x}} \neq \mathbf{0}$ then (3) would need to be replaced by $f_\alpha(A_{x_1} + \hat{x}_1 I_m, \ldots, A_{x_s} + \hat{x}_s I_m) = 0$ where $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_s)$.

4 The inexact case

Now suppose $f_1, \ldots, f_t$ are approximate and/or analytic functions and/or $\hat{\mathbf{x}}$ is only an approximate zero. I will follow the method of the last section using approximate linear algebra to obtain basis and coordinate function matrices which approximate the local ring.

There are several issues to deal with. First, Lemma 1 will not hold, statement ii) does not really apply and, generally, when viewing the rows of $\mathbf{M}(k, \hat{\mathbf{x}})$ as equations these equations are often inconsistent. Also i) may be false when $\hat{\mathbf{x}}$ is not an exact zero. It is no longer clear as to which k to use in $\mathbf{M}(k, \hat{\mathbf{x}})$. RREF is notoriously unstable with approximate data. I will deal with these problems in the following subsections.

4.1 Step 1: Choosing $\hat{\mathbf{x}}, k$ and Computing $\mathbf{M}(k, \hat{\mathbf{x}})$

In principle one could input just about any point $\hat{\mathbf{x}}$ but obviously one will not get much out of the algorithm unless $\hat{\mathbf{x}}$ is close to a zero, better yet a multiple zero or suspected zero cluster. A good choice in the latter case is the output of the LVZ algorithm [5, 2] as this algorithm tends to find a point maximizing the approximate multiplicity m.

The choice of ε will depend on your objective. If you want to search for nearby zeros then a large value of ε is desirable, say 0.1 or even 0.5 The idea is to get as large a value m with a stable range. See Examples 6.1 – 6.3 below. On the other hand, if one suspects that one has an approximate multiple point of multiplicity m and would like confirmation that it is not a cluster of distinguishable near points then one should use as small an ε as possible to get a stable range for m. In either case trial and error will probably be necessary.

I have found that to calculate $\mathbf{M}(k, \hat{\mathbf{x}})$ a straighforward calculation of $\partial_{\mathbf{x^j}}((\mathbf{x^i} - \hat{\mathbf{x}}^{\mathbf{i}})f_\alpha)(\hat{\mathbf{x}})$ for each entry of $\mathbf{M}(k, \hat{\mathbf{x}})$ seems to work as fast as anything when using CAS such as MAPLE. A trick in looking for the stable range is to calculate $\mathbf{M}(k, \hat{\mathbf{x}})$ for as large a k as possible given your system (the number of entries is on the order of $\mathcal{O}(t\, k^{2s})$). Then use the fact that $\mathbf{M}(k-1, \hat{\mathbf{x}})$ is a submatrix of $\mathbf{M}(k, \hat{\mathbf{x}})$. There are also strategies that can be used to compute $\mathbf{M}(k, \hat{\mathbf{x}})$ recursively, using the fact that many computations are essentially repeated (see Figure 2, for example) but my experience is that the bookkeeping necessary takes more time than the original computations. If you are implementing this on a platform such as MATLAB/SCILAB you have to use your own routines for calculating the partial derivatives and the bookkeeping may be significantly faster.

To find $H_\varepsilon(k, \hat{\mathbf{x}})$ one needs to calculate an approximate rank. If one has an implementation of the Li-Zeng rank revealing algorithm then that would be the most efficient way to proceed. Otherwise one can make do with the Singular Value algorithm: count singular values greater than ε. Unlike the exact case where one can use the smallest k in the stable range, I find from experiments that using as large a k as you can handle gives more accurate results. Also, since Lemma 1(i) may fail, this condition could be checked directly although one can go to the next step and inspect the output.

4.2 Step 2: ARRREF

In the exact case one can rotate the matrix and apply a standard RREF algorithm and then rotate back. The big problem in the inexact case is that the standard RREF algorithm is not robust enough for approximate numerical work. Therefore I propose the following *approximate reverse reduced row echelon form* algorithm which has worked reasonably well in my examples.

Algorithm ARRREF

- **Input:** $m \times n$ matrix M, tolerance ε.
- Apply the SVD decomposition to get $M = Q_1 \Sigma Q_2^\top$. Calculate the approximate rank r by counting the number of diagonal entries of Σ which are greater than ε. The approximate row space of M is then spanned by the first r rows of $Q_2^\top$, call the $r \times n$ matrix of these rows A.
- We can assume the right hand, n^{th} column of M, hence A, is not $\mathbf{0}$. Initialize a matrix B to be the n^{th} column. Starting with the $j = n-1^{\text{st}}$ column and proceeding leftward we can check, using SVD or rank revealing, that the j^{th} column is approximately independent of the space spanned by columns $j+1, \ldots, n$, i.e. the approximate rank increases by 1 when this column is added. If the j^{th} column is approximately independent then add this column as the first (leftmost) column of B. When done B will consist of independent columns of A. B should be reasonably well conditioned, check. Note: this step may work better with a slightly tighter tolerance, I use $.5 * \varepsilon$ in rank revealing.
- **Output:** $R = B^{-1}A$. (B^{-1} is just the usual inverse.) and a list of indicies of the pivot columns.

Note that if some column is a *pivot column*, i.e. one of the columns placed in B, then all dependence relations are with columns to the left. Thus, in R, all entries above and to the right of the pivots should be numerically zero. This algorithm could fail, so the results should be checked, a small adjustment of the tolerance may help.

Assuming the input to ARRREF is $\mathbf{M}(k, \hat{\mathbf{x}})$ then the number of rows of R should be $\binom{k+s}{s} - m$. Thus there should be m non-pivot columns. The list of pivot columns given in the output could be used to check condition i) in Lemma 1. It will be awkward if the first column is a pivot column, this shouldn't happen if $\hat{\mathbf{x}}$ is an approximate zero, a loosening of the tolerance may help.

4.3 Step 3: Extract Multiplication Matrices

The point here is to deliberately ignore the fact that the sytems determined by the matrix $\mathbf{M}(k, \hat{\mathbf{x}})$ and its ARRREF form R may be inconsistent. Since ARRREF outputs a list of the pivot columns it is straightforward symbolic programming to extract the multiplication matrices A_{x_i} which are guarenteed to exist if condition i) of Lemma 1 holds. The $\mathbb{C}$-basis matrices A_{g_i} may be immediately extracted or can be calculated from the A_{x_i} since the basis monomials are known from the missing indices.

5 The Approximate Local Ring and calculation of near points

If the steps in the previous section all work well then one should get $m \times m$ $\mathbb{C}$-basis matrices $A_{g_1}, \ldots, A_{g_m}$ and coordinate matrices $A_{x_1}, \ldots, A_{x_s}$. If these satisfy the following analogs of

equations (1) – (4) of §3 then I call this system an *approximate local ring* or ALR.

$$A_{x_i}A_{x_j} - A_{x_j}A_{x_i} \approx \mathbf{0}, \qquad \text{all } i,j \tag{5}$$

$$A_{x_1}^{j_1}A_{x_2}^{j_2}\cdots A_{x_s}^{j_s} \approx \sum_{i=1}^{m} c_i A_{g_i} \qquad j_\alpha \geq 0, \text{ some reasonable } c_i \in \mathbb{C} \tag{6}$$

$$f_\alpha(A_{x_1} + \hat{x}_1 I_m, \ldots, A_{x_s} + \hat{x}_s I_m) \approx 0, \qquad \alpha = 1, \ldots, t \tag{7}$$

$$\lambda_{i,1}, \ldots, \lambda_{i,m} \text{ are small}, \qquad \lambda_{i,j} \text{ eigenvalues of} A_{x_i} \tag{8}$$

By "reasonable" in (6) I mean not very large in magnitude. In this preliminary report I will not attempt to further define the notion of "approximate", "reasonable" or "small", obviously this would be desirable in a final paper. For now the reader will have to accept the examples in §6 as evidence that this concept is useful.

In practice one can work with these matrices as if they do define an m-dimensional matrix subalgebra of the $m \times m$ matrices even though such a algebra will not exactly exist. For example, multiplying the matrix monomials by a large coefficient may no longer satisfy (6).

Condition (8) is the justification for the term "local". For then one would expect the matrix monomials $A_{x_1}^{j_1}A_{x_2}^{j_2}\cdots A_{x_s}^{j_s}$ to be small for $j_1 + \cdots + j_s$ large and hence a reasonable polynomial made up of these monomials with non-zero constant term should be invertible in some sense.

5.1 The Theorem

From (5) one might expect the existence of almost common eigenvectors of the A_{x_i}, this suggests

Theorem 1 *Let $A_{x_1}, \ldots, A_{x_s}$ be the coordinate matrices of an ALR for the system $f_1, \ldots, f_t$ at $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_s)$. If $\mathbf{v}$ is an approximate common eigenvector of the A_{x_i} with $A_{x_i}\mathbf{v} \approx \lambda_i \mathbf{v}$ then $(\lambda_1 + \hat{x}_1, \ldots, \lambda_s + \hat{x}_s)$ is an approximate zero of the system $f_1, \ldots, f_t$.*

A heuristic argument might go as follows. From $A_{x_i}\mathbf{v} \approx \lambda_i \mathbf{v}$ one gets $A_{x_1}^{j_1}A_{x_2}^{j_2}\cdots A_{x_s}^{j_s}\mathbf{v} \approx \lambda_1^{j_1}\cdots\lambda_s^{j_s}\mathbf{v}$ hence summing up gives $f_\alpha(\lambda_1, \ldots, \lambda_s)\mathbf{v} \approx f_\alpha(A_{x_1}, \ldots, A_{x_s})\mathbf{v} \approx 0$ from (7). Since I can assume $\mathbf{v}$ to be normalized to have a component of 1 it would follow that $f_\alpha(\lambda_1, \ldots, \lambda_s) \approx 0$. Of course one worries that the small errors over the possibly many terms of f_α add up. Here (8) may help as the terms $A_{x_1}^{j_1}A_{x_2}^{j_2}\cdots A_{x_s}^{j_s}$ are small so the small relative errors in $A_{x_i}\mathbf{v} \approx \lambda_i\mathbf{v}$ give smaller absolute errors in $A_{x_1}^{j_1}A_{x_2}^{j_2}\cdots A_{x_s}^{j_s}\mathbf{v}$.

To counter this argument one could argue that since, by (8), the λ_i are small the point $(\lambda_1, \ldots, \lambda_s)$ is close to a presumably multiple approximate zero $\hat{\mathbf{x}}$ so that alone could explain $f_\alpha(\lambda_1, \ldots, \lambda_s)$ being small. Thus a believable proof of Theorem 1 along these lines would show that $f_\alpha(\lambda_1, \ldots, \lambda_s)$ is smaller than expected for a random point close to $\hat{\mathbf{x}}$. Example 6.3a below suggests looking for a completely different argument.

At this point the theorem must be considered vacuous as a definition of the not yet precise notion of "approximate zero." A number of examples suggest that there is a more precice notion

of approximate zero making Theorem 1 true. In practice it may not matter if this Theorem is proven or even precise since the points obtained can be refined by various methods. The utility of this method will be justified if, in practice, we can determine and classify m points near $\hat{\mathbf{x}}$ as to multiplicity.

5.2 Calculation of common eigenvectors

Standard eigenvector software seems sufficient for the purposes of this method, the goal is to obtain at least m independent usable eigenvectors as initial values for the algorithm of this section. Usable means that the first entry should be normalized, or normalizable, to 1, assuming the first column of R is a pivot column.

There are likely more sophisticated methods available but I have found the following to be sufficient for my purposes.

Common Eigenvector Calculation (CEC): Let $A_{x_1}, \dots A_{x_m}$ be the multiplication matrices. For random floating point numbers $r_1, \dots, r_m$ let $A = r_1 A_{x_1} + \cdots + r_m A_{x_m}$. Use the eigenvectors of M, rather than any A_{x_i}. (This is apparently a well known trick, I learned it from [4]).

For each normalized eigenvector $\mathbf{v} = [1, v_2, \dots, v_m]^\top$ of M solve the quadradic least squares problem

$$\begin{bmatrix} A_{x_1}\mathbf{u} - \mu_1\mathbf{u} \\ A_{x_2}\mathbf{u} - \mu_2\mathbf{u} \\ \vdots \\ A_{x_s}\mathbf{u} - \mu_s\mathbf{u} \\ u_1 - 1 \end{bmatrix} = \mathbf{0}$$

in variables $u_1, \dots, u_m, \mu_1, \dots, \mu_s$ where $\mathbf{u} = [u_1, u_2, \dots, u_m]^\top$. This seems best solved by one or more iterations of Gauss-Newton using inital values $\mathbf{u} = \mathbf{v}$, μ_i the first coordinate of $A_{x_i}\mathbf{v}$, $i = 1..m$.

The desired zero is given by $\hat{\mathbf{x}} = [\mu_1, \dots \mu_s]^\top$. ∎

The method outlined in the last two sections for finding approximations for zeros predicted by the multiplicity matrices in the inexact case will be called the *ALR method* in the remainder of this paper.

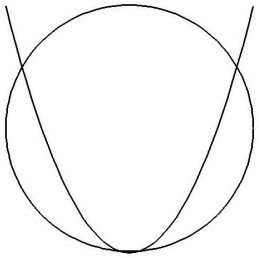
6 Examples

In this section we look at some examples.

6.1 A simple Example

Example 6.1

$$f_1 = x^2 + y^2 - 1$$
$$f_2 = y - 2x^2 + 1.01$$



By inspection of the graph I see the possiblity of several/multiple zeros near $(0, -1)$. I calculate $\mathbf{M}(k, (0, -1))$ for $k = 0, 1, 2, \ldots, 5$, find the approxi-rank with a loose tolerance of 0.1 and see that $H_{0.1}(k, (0, -1)) = \{1, 2, 2, \ldots\}$ so the stable range for 2 is $k > 1$. For better accuracy, I pick $k = 3$ in the stable range and find

$$\mathbf{M}(3, (0, -1)) = \begin{bmatrix} 0.0 & 0.0 & -2.0 & 1.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.01 & 0.0 & 1.0 & -2.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & -2.0 & 0.0 & 1.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.01 & 0.0 & 0.0 & 1.0 & 0.0 & -2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -2.0 & 0.0 & 1.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.01 & 0.0 & 0.0 & 1.0 & 0.0 & -2.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -2.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.010 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -2.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.01 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -2.0 \\ 0.0 & 0.0 & 0.0 & -0.0 & 0.0 & 0.010 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Applying ARRREF I obtain

$$R = \begin{bmatrix} -0.0033253 & -0.0 & 1.0 & -0.0 & -0.0 & -0.0 & 0.0 & -0.0 & 0.0 & -0.0 \\ -0.0066624 & -0.0 & 0.0 & 1.0 & 0.0 & 0.0 & -0.0 & 0.0 & -0.0 & 0.0 \\ 0.0 & -0.0033287 & 0.0 & 0.0 & 1.0 & 0.0 & -0.0 & -0.0 & 0.0 & -0.0 \\ 0.0 & -0.0 & -0.0 & -0.0 & -0.0 & 1.0 & 0.0 & 0.0 & -0.0 & -0.0 \\ -0.0 & -0.0066643 & -0.0 & -0.0 & -0.0 & -0.0 & 1.0 & -0.0 & 0.0 & 0.0 \\ 0.0 & -0.0 & -0.0 & -0.0 & -0.0 & -0.0 & 0.0 & 1.0 & 0.0 & -0.0 \\ 0.0 & 0.0 & -0.0 & -0.0 & -0.0 & 0.0 & -0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & -0.0 & 0.0 & 0.0 & 0.0 & -0.0 & 0.0 & -0.0 & 0.0 & 1.0 \end{bmatrix}$$

Note the $\mathbb{C}$-basis consists of $1, x$. From this I can calculate

$$A_x = \begin{bmatrix} 0 & 1 \\ 0.0066624 & 0 \end{bmatrix} \text{ and } A_y = \begin{bmatrix} 0.0033253 & 0 \\ 0 & 0.0033287 \end{bmatrix}$$

Calculating the eigenvalues and eigenvectors for A_x we get (rounded)

$$\lambda_1 = .0816236, \mathbf{v}_1 = \begin{bmatrix} 0.99669 \\ 0.081353 \end{bmatrix}, \lambda_2 = -.0816236, \mathbf{v}_2 = \begin{bmatrix} -0.99669 \\ 0.081353 \end{bmatrix}$$

Applying the common eigenvector algorithm with inital value $\lambda_1, \mathbf{v}_1$ gives eigenvector $\mathbf{u}_1$ and eigenvalues μ_1, μ_2

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ .081624 \end{bmatrix}, \quad \hat{\mathbf{y}}_1 = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 0.0816235161 \\ 0.0033253476 \end{bmatrix}$$

Likewise I get

$$\hat{\mathbf{y}}_2 = \begin{bmatrix} -0.0816235161 \\ 0.0033253476 \end{bmatrix}$$

Remembering that I used $\hat{\mathbf{x}}_0 = [0, -1]^\top$ the approximate solutions are

$$\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_0 + \hat{\mathbf{y}}_1 = \begin{bmatrix} 0.0816235161 \\ -0.9966746524 \end{bmatrix} \text{ and } \hat{\mathbf{x}}_2 = \hat{\mathbf{x}}_0 + \hat{\mathbf{y}}_2 = \begin{bmatrix} -0.0816235161 \\ -0.9966746524 \end{bmatrix}$$

Note that $|f_1(\hat{\mathbf{x}}_1)| \approx .2 * 10^{-4}, |f_2(\hat{\mathbf{x}}_1)| \approx .5 * 10^{-6}$, these values are about two orders of magnitude smaller than these functions evaluated at a random point equivalently near $(0, -1)$. Using Newton iteration to refine these values further gives

$$\hat{\mathbf{x}} = \begin{bmatrix} \pm 0.08167243611 \\ -0.9966592261 \end{bmatrix}$$

which are good approximations of the actual zeros. The smallest singular value of the Jacobian matrix of Newton Iteration at these points is approximately 0.2. Thus we should reasonably view these zeros as two distinct zeros of the system f_1, f_2.

Note that A_x, A_y do generate an approximate local ring as in the definition of §4. $\|A_x A_y - A_y A_x\| \approx .3 * 10^{-5}$. I note that from the first row of $\mathbf{M}(3, (0, -1))$ I can read off the equations after change of coordinates bringing $\hat{\mathbf{x}}$ to $\mathbf{0}$ $\tilde{f}_1 = -2y + x^2 + y^2, \tilde{f}_2 = .01 + y - 2x^2$, so I can calculate $\|\tilde{f}_1(A_x, A_y)\| = \|-2A_y + A_x^2 + A_y^2\| \approx .2 * 10^{-4}$ and $\|\tilde{f}_2(A_x, A_y)\| = \|.01 I_2 + A_y - 2A_x^2\| \approx .4 * 10^{-5}$ The eigenvalues of A_x were on the order of 0.08 while those of A_y were around .003 so are certainly "small". Unexpected, however is the fact that $A_y \approx 0.00332702 I_2$ so this ALR is no local ring. ∎

6.2 A more complicated example

Consider the following two dimensional system $H := [h_1, h_2]^\top$ with 49 total zeros.

$$\begin{aligned}
h_1 = {} & 0.053293293\,x^7 - 1.1191592\,x^5y^2 + 1.8652653\,x^3y^4 - 0.37305305\,xy^6 - 0.40449609\,x^6 \\
& + 1.1336937\,x^5y + 6.0674414\,x^4y^2 - 3.7789790\,x^3y^3 - 6.0674414\,x^2y^4 + 1.1336937\,xy^5 \\
& + 0.40449610\,y^6 - 0.15826346\,x^5 - 2.7078499\,x^4y + 1.5826346\,x^3y^2 + 5.4156997\,x^2y^3 \\
& - 0.79131732\,xy^4 - 0.54156997\,y^5 + 16.895936\,x^4 - 34.728465\,x^3y - 101.375612\,x^2y^2 \\
& + 34.728465\,xy^3 + 16.895936\,y^4 - 30.136281\,x^3 + 108.15682\,x^2y + 90.408842\,xy^2 \\
& - 36.052275\,y^3 + 7.1139288\,x^2 - 4.7385010\,xy - 7.1139288\,y^2 - 0.40902330\,x \\
& - 0.07700852\,y + 0.00670127 \\
h_2 = {} & 0.37305305\,x^6y - 1.8652653\,x^4y^3 + 1.1191592\,x^2y^5 - 0.0532932\,y^7 - 0.18894894\,x^6 \\
& - 2.4269766\,x^5y + 2.8342342\,x^4y^2 + 8.0899219\,x^3y^3 - 2.8342342\,x^2y^4 - 2.4269766\,xy^5 \\
& + 0.18894895\,y^6 + 0.54156997\,x^5 - 0.79131732\,x^4y - 5.4156997\,x^3y^2 + 1.5826346\,x^2y^3 \\
& + 2.7078499\,xy^4 - 0.15826346\,y^5 + 8.6821163\,x^4 + 67.583744\,x^3y - 52.092698\,x^2y^2 \\
& - 67.583744\,xy^3 + 8.6821163\,y^4 - 36.052275\,x^3 - 90.408842\,x^2y + 108.15682\,xy^2 \\
& + 30.136280\,y^3 + 2.3692505\,x^2 + 14.227858\,xy - 2.3692505\,y^2 + 0.077008522\,x \\
& - 0.4090233\,y - 0.00497924
\end{aligned}$$

The equations here have been rounded a few digits for display purposes from the actual ones used in the calculation of this example.

Example 6.2a In this example I find and classify the nine zeros (counted by multiplicity) near the origin. With $\hat{\mathbf{x}} = (0,0) = \mathbf{0}$ I find, with a tolerance of .001, a stable range for $m = 9$ starting at $k = 9$ so I calculate $\mathbf{M}(10, \hat{\mathbf{x}})$, a 110×66 matrix with approxi-rank 57. A calculation of the ARRREF form leads to the multipliciation matrices with commutator norm $5 * 10^{-9}$. The eigenvalues for A_x are close to, but not identical to full precision, two values, however MAPLE normalizes all the eigenvectors with first entry close to 1. Using the CEC I do get 9 points as listed in Figure 5 below. I compare this with 9 solutions given by PHCpack [11] with default settings in blackbox mode.

The results of my method and PHCpack are quite similar, a difference being that the 9 near zeros are scattered through the 49 zeros given by PHCpack.

Refinement by Newton's method, no deflation, suggests that zeros 1 and 3 are the same double zero as are zeros 2 and 4. Zero 5 is a simple zero, but zeros 6 through 9 all appear to refine to the same solution of multiplicity 4. Now deflation can be used on the multiple zeros and I

Figure 5: Ex. 6.2a results compared with PHCpack

eigenvalue number	ALR method, no refinement	PHCpack no refinement	PHC solution number
1	$\begin{bmatrix} 0.045001 + 0.022728\,i \\ -0.022728 + 0.45562\text{E-}03\,i \end{bmatrix}$	$\begin{bmatrix} 0.044999 + 0.022727\,i \\ -0.022727 + 0.45336\text{E-}03\,i \end{bmatrix}$	1
2	$\begin{bmatrix} 0.045001 - 0.022728\,i \\ -0.022728 - 0.45336\text{E-}03\,i \end{bmatrix}$	$\begin{bmatrix} 0.045000 - 0.022728\,i \\ -0.022728 - 0.45470\text{E-}03\,i \end{bmatrix}$	34
3	$\begin{bmatrix} 0.044999 + 0.022727\,i, \\ -0.022727 + 0.45317\text{E-}03\,i \end{bmatrix}$	$\begin{bmatrix} 0.044999 + 0.022727\,i \\ -0.022727 + 0.45363\text{E-}03\,i \end{bmatrix}$	11
4	$\begin{bmatrix} 0.044999 - 0.022726\,i \\ -0.022726 - 0.45348\text{E-}03\,i \end{bmatrix}$	$\begin{bmatrix} 0.045000 - 0.022728\,i \\ -0.022728 - 0.45409\text{E-}03\,i \end{bmatrix}$	31
5	$\begin{bmatrix} 0.044545 \\ 0.595\text{E-}10 \end{bmatrix}$	$\begin{bmatrix} 0.044545 \\ 0.476\text{E-}10 \end{bmatrix}$	28
6	$\begin{bmatrix} 0.045495 \\ -0.045454 \end{bmatrix}$	$\begin{bmatrix} 0.045454 \\ -0.045455 - 0.127\text{E-}06\,i \end{bmatrix}$	12
7	$\begin{bmatrix} 0.045434 + 0.348\text{E-}05\,i \\ -0.045455 - 0.753\text{E-}07\,i \end{bmatrix}$	$\begin{bmatrix} 0.045453 - .329\text{E-}06\,i \\ 0.045455 - 0.631\text{E-}06\,i \end{bmatrix}$	18
8	$\begin{bmatrix} 0.04543 - 0.348\text{E-}05\,i \\ -0.045455 + 0.753\text{E-}07\,i \end{bmatrix}$	$\begin{bmatrix} 0.045453 - 0.149\text{E-}05\,i \\ -0.045456 + 0.770\text{E-}07\,i \end{bmatrix}$	30
9	$\begin{bmatrix} 0.045455 \\ -0.036394 \end{bmatrix}$	$\begin{bmatrix} 0.045454 + 0.192\text{E-}06\,i \\ -0.045456 - 0.115\text{E-}05\,i \end{bmatrix}$	47

conclude the zeros of this sytem near the origin, with multiplicities are

$$\hat{\mathbf{x}}_1 = \begin{bmatrix} 0.045000000 + 0.0227272727\,i \\ -0.027272727 + 0.000454545\,i \end{bmatrix} \qquad m = 2$$

$$\hat{\mathbf{x}}_2 = \begin{bmatrix} 0.045000000 - 0.0227272727\,i \\ -0.027272727 - 0.000454545\,i \end{bmatrix} \qquad m = 2$$

$$\hat{\mathbf{x}}_3 = \begin{bmatrix} .0445454545 \\ 0 \end{bmatrix} \qquad m = 1$$

$$\hat{\mathbf{x}}_4 = \begin{bmatrix} .0454545455 \\ -.0454545455 \end{bmatrix} \qquad m = 4$$

∎

Example 6.2b: Of the 49 zeros of the example H of this subsection PHCpack identifies two pairs of clustered zeros. In fact, there are more, such as the ones above. In this example I analyze one of the identified clusters, PHCpack solutions 5 and 6.

```
solution 5 :    start residual :  1.821E-14   #iterations : 1   success
t :  1.00000000000000E+00   0.00000000000000E+00
m : 5
the solution for t :
 x :  1.20462113364280E+00   6.13535853028936E-01
 y :  5.68282329516424E-01  -1.15901522965219E+00
== err :  1.061E-10 = rco :  4.049E-06 = res :  8.882E-15 = complex regular ==
solution 6 :    start residual :  1.155E-14   #iterations : 1   success
t :  1.00000000000000E+00   0.00000000000000E+00
m : 6
the solution for t :
 x :  1.20462113364444E+00   6.13535853038710E-01
 y :  5.68282329506649E-01  -1.15901522965055E+00
== err :  9.042E-11 = rco :  4.049E-06 = res :  1.021E-14
= complex clustered : 5 ==
```

Starting with solution 5, I deflate using PHCpack with tolerance on numerical rank set at $1.0\mathrm{E}-03$. The result is

$$\tilde{\mathbf{x}}_5 = \begin{bmatrix} 1.20461321453912 + 0.602692583945157\,i \\ 0.579125598432050 - 1.15902314734559\,i \end{bmatrix}$$

This is suprisingly far from both solutions 5 and 6.

The reason is $H_{.01}(k, \tilde{\mathbf{x}}_5)$ has a stable range for $m = 3$, not $m = 2$, beginning at $k = 4$. So there is another near zero affecting the deflation. Since I only have 2 variables to get increased accuracy I use $\mathbf{M}(5, \tilde{\mathbf{x}}_5)$ and directly, without further refinement, produce the following three

zeros

$$\hat{\mathbf{y}}_1 = \begin{bmatrix} 1.20444079658657555 + .613795791160425930\,i \\ .568022391235419422 - 1.15919556593683582\,i \end{bmatrix}$$
$$\hat{\mathbf{y}}_2 = \begin{bmatrix} 1.20465162929802649 + .613476260252905803\,i \\ .568197556363474355 - 1.15898311383042696\,i \end{bmatrix}$$
$$\hat{\mathbf{y}}_3 = \begin{bmatrix} 1.20408938594172610 + .590909745035199552\,i \\ .590908437362156924 - 1.15954697659234785\,i \end{bmatrix}$$

Note $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2$ are close to the clustered solutions 5,6 but $\hat{\mathbf{y}}_3$ is near PHCpack solution 7.

With tolerance set at 0.005 I find the approxi-rank of the Jacobian matrix of ordinary Newton's method (no deflation) is 2 at $\hat{\mathbf{y}}_1$ which suggests that it is a simple zero, while the Jacobian matrix has approxi-rank 0 at the other two points, suggesting that they are approximations of a double zero. Using PHCpack to do deflation, still with tolerance on numerical rank at $1.0\mathrm{E}-05$, from $\hat{\mathbf{y}}_1$ gives

$$\tilde{\mathbf{y}}_1 = \begin{bmatrix} 1.20454545800870 + 0.613636354355985\,i \\ 0.568181828189372 - 1.15909090528630\,i \end{bmatrix}$$

Finally note, see below, that the deflated answer $\tilde{\mathbf{y}}_1$ agrees with an expected exact solution to 9 decimal places. It important to use an initial point of deflation which avoids unwanted points of the cluster. ∎

For these examples I can verify the conclusions of this analysis since I know the construction of the system H. I started with a complex univariate polynomial $f(z)$ of degree 7 with zeros $z_1 = z_2 = \frac{1}{22} - \frac{1}{22}i, z_3 = \frac{49}{1100}$ and the other zeros random larger rational complex numbers. Repacing $z = x + iy$ and taking real and imaginary parts and approximating by floating point numbers gives the two real equations h_1, h_2. A necessary condition for $\hat{\mathbf{x}} = [x, y]^\top$ to be a solution to this system is that $z = x + iy$ is a solution to $f(z) = 0$. The real solutions to $H = \mathbf{0}$ correspond to the complex solutions to $f(z) = 0$, but in Example 6.2a the multiplicities were different. There are complex solutions to H as well. Since H is a real system, the conjugate of any complex solution is also a solution so it is actually necessary to not only have $z = x + iy$ be a solution of f but also $z = \bar{x} + i\bar{y}$. Note that for $\hat{\mathbf{y}}_3$ one has $\bar{x} + i\bar{y} = 0.044545425 + .8085 * 10^{-6}i \approx z_3$ while for $\hat{\mathbf{y}}_1$, $\bar{x} + i\bar{y} = 0.045670135 - 0.0454230700i \approx z_1 = z_2$ and similarly for $\hat{\mathbf{y}}_2$. Thus $\hat{\mathbf{y}}_3$ definitely represents a different solution to the original exact system than $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2$ which appear to represent the same solution. That it is a multiple solution makes sense since z_1 was a multiple solution of $f(z)$.

6.3 An analytic Example

Although the motivation for my method comes from the study of polynomial systems, this method may have its most significant application in analytic systems where it is often not possible to have a solution method which gives all solutions. This example is due to Griewank and Osborne [3] in 1983.

I study the sytem $G = [g_1, g_2, g_3]^\top$ given by

$$\begin{aligned} g_1 &= 0.5x^2\cos(10x) + x\sin(10y) + 10x\sinh(z) \\ g_2 &= \cos(5x) + \sin(5y) - 1 \\ g_3 &= \cosh(5x) + 5\sinh(z) - 1 \end{aligned}$$

Example 6.3a Griewank and Osborne were specifically interested in minimization problems and, hence, real solutions. They identified the origin $(0, 0, 0)$ as a multiple isolated solution with a nearby simple zero at approximately $(.12, .033, -.035)$. In fact my method enabled me to find 4 imaginary solutions close by that I had not previously known. In the interest of full disclosure I must mention that I do much of my work with MAPLE 9 which, using standard procedures was not helpful in finding zeros of this system. These complex zeros were quite easily found with the `FindRoot` function of MATHEMATICA.

Applying the Li-Zeng rank revealing method I find a stable range for $m = 7$ starting at $k = 7$ for $\hat{\mathbf{x}} = \mathbf{0} = (0, 0, 0)$. Applying ARRREF to $\mathbf{M}(7, \mathbf{0})$, a 252×120 matrix, returns a 113×120 matrix R which consists of a 113×7 matrix appended with the identity I_{113}. It is then easy to extract the 7×7 multiplication matrices A_x, A_y and A_z for $\mathbb{C}$-basis $1, x, y, z, x^2, xy, xz$.

One calculates the commutators $\|A_iA_j - A_jA_i\| < .09$ for $1 \le i < j \le 3$. With more difficulty one can estimate $\|g_i(A_x, A_y, A_z)\|$ but these are not particularly small numbers. This suggests that Theorem 1 may require a different approach for its proof.

However, A_x has a good set of eigenvectors, especially given the fact that it is known that $\mathbf{0}$ is a multiple zero: two of the eigenvectors, according to MAPLE, are $[1, 0, \ldots, 0]^\top$ with common eigenvalues 0. Thus one can read from the $\mathbb{C}$-basis that $x = y = z = 0$. The other eigenvalue-eigenvector pairs can be fed to the CEC from which the following 5 points can be extracted (data rounded for display):

$$\begin{bmatrix} .06376 \pm .25626i \\ -.15067 \pm .13560i \\ .13980 \mp .06117i \end{bmatrix}, \quad \begin{bmatrix} .11601 \\ .03286 \\ -.03459 \end{bmatrix}, \quad \begin{bmatrix} .12962 \mp .08056i \\ .02669 \pm .04985i \\ -.02404 \mp .05445i \end{bmatrix}$$

with residues $.013, .11 * 10^{-5}, .39 * 10^{-3}$ respectively.

Newton refinement gives the following values (rounded) with residues $< 10^{-9}$:

$$\begin{bmatrix} .06534 \pm .26056i \\ -.15336 \pm .14012i \\ .14398 \mp .06352i \end{bmatrix}, \quad \begin{bmatrix} .11591 \\ .03281 \\ -.03453 \end{bmatrix}, \quad \begin{bmatrix} -.12972 \pm .08182i \\ .02732 \mp .05071i \\ -.02350 \pm .05531i \end{bmatrix}$$

with smallest singular value of the Newton Method Jacobian being $1.1, 0.10, 0.16$ respectively. Thus we can conclude that these 5 zeros are simple zeros and the zero at the origin is of multiplicity 2. ∎

Example 6.3b By inspection it is easily seen that $\hat{\mathbf{x}} = [0, \frac{\pi}{5}, 0]^\top$ is also a solution to the system G. A calculation using $\mathbf{M}(k, \hat{\mathbf{x}})$ for various k, ε shows that 5 is in the stable range for $m = 3$ for even tight tolerance, eg. $\varepsilon = .5 * 10^{-5}$. One is tempted to call $\hat{\mathbf{x}}$ a zero of multiplicity

3. However, applying ARRREF to $\mathbf{M}(5, .6283185)$, $\varepsilon = .5 * 10^{-3}$ gives 3×3 multiplication matrices

$$B_x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -0.40008 \\ 0 & 0 & 0.00995 \end{bmatrix}, \; B_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0.00995 \\ 0 & 0 & -.000247 \end{bmatrix}, \; B_z = \begin{bmatrix} 0 & 0 & 1.0004 \\ 0 & 0 & 0.00996 \\ 0 & 0. & -.000247 \end{bmatrix}$$

A calculation gives $\|B_x B_\alpha - B_\alpha B_x\| < .5 * 10^{-7}$ for $\alpha = y, z$ while $\|B_y B_z - B_z B_y\| \approx .01$. This last number is suprising, relative to the first, since B_y, B_z are almost identical matrices.

According to MAPLE, B_x has a repeated eigenvector $[1, 0, 0]^\top$ with eigenvalue 0, suggesting that $\hat{\mathbf{x}}$ is a zero of multiplicity 2. Note the $\mathbb{C}$-basis is $1, x, y$ so one should check z. In fact, by inspection, $[1, 0, 0]^\top$ is an eigenvector for eigenvalue 0 of B_z.

The third eigenvalue-eigenvector pair for B_x gives, under the common eigenvector algorithm, the common eigenvector $\mathbf{v}$, and eigenvalue vector μ

$$\mathbf{v} = \begin{bmatrix} 1 \\ .00995 \\ -.00025 \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \begin{bmatrix} .0099504963 \\ -.00024748 \\ -.00024758 \end{bmatrix}$$

This suggests a possible zero at

$$\hat{\mathbf{y}} = \hat{\mathbf{x}} + \mu = \begin{bmatrix} .00995050 \\ .62807102 \\ -.00024758 \end{bmatrix}$$

This point already has a residue of $.2 * 10^{-6}$ and is almost indistinguishable from its refinement which has Jacobian with smallest singular value about .005, still large enough to accept as a separate zero. Thus I am tempted to conclude that $\hat{\mathbf{x}}$ is a zero of multiplicity 2 with nearby zero $\hat{\mathbf{y}}$ of multiplicity 1.

Because the system and zero $(0, \pi/5, 0)$ are exact, the multiplicity matrices can be calculated exactly and an exact calculation is possible of their rank. The conclusion above agrees with the exact multiplicity 2 of this zero. It should also be noted that MATHEMATICA easily finds the zero $\hat{\mathbf{y}}$ as distinct from zero $\hat{\mathbf{x}}$. ∎

References

[1] G. Birkhoff, S. Mac Lane *A survey of Modern Algebra*, Macmillan, 1941.

[2] B. H. Dayton and Z. Zeng,*Computing the Multiplicity Structure in Solving Polynomial systems*, Proc. of ISSAC '05, ACM Press, pp. 116–123.

[3] A. Griewank and M.R. Osborne, *Analysis of Newton's Method at irregular singularities*. SIAM J. Numer. Anal. 20(4), pp. 747-773, 1983.

[4] I. Janovitz-Freireich, L. Ronyai, A. Szanto, *Approximate Radical of Ideals with Clusters of Roots*, Proc of ISSAC '07, ACM press.

[5] A. Leykin, J. Verschelde, and A. Zhao, *Newton's method with deflation for isolated singularities of polynomial systems*. Preprint, (2004).

[6] T. Y. Li and Z. Zeng, *A rank-revealing method with updating, downdating and applications*. to appear: SIAM J. Matrix Anal. Appl.

[7] F. S. Macaulay, *The Algebraic Theory of Modular Systems*, Cambridge Univ. Press, 1916.

[8] H. M. Möller and H. J. Stetter, *Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems*, Numer. Math., 70 (1995), pp. 311–329.

[9] T. Mora, *Solving Polynomial Equation Systems II*. manuscript.

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

[11] J. Verschelde, *Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Softw., 25(2) pp 251–276, 1999. Software available at `http://www.math.uic.edu/~jan`. (July 2005 version used in this paper.)