Nearest Multivariate System with Given Root Multiplicities

SCOTT R. POPE *, AGNES SZANTO †

Abstract
We present a symbolic-numeric technique to find the closest multivariate polynomial system to a given one which has roots with prescribed multiplicity structure. Our method generalizes the “Weierstrass iteration” defined by Ruatta to the case when the input system is not exact, i.e. when it is near to a system with multiple roots, but itself might not have multiple roots. First, using interpolation techniques, we define the “generalized Weierstrass map”, a map from the set of possible roots to the set of systems which have these roots with the given multiplicity structure. Minimizing the 2-norm of this map formulates the problem as an optimization problem over all possible roots. We use Gauss-Newton iteration to compute the closest system to the input with given root multiplicity together with its roots. We give explicitly an iteration function which computes this minimum. These results extends previous results of Zhi and Wu and results of Zeng from the univariate case to the multivariate case. Finally, we give a simplified version of the iteration function analogously to the classical Weierstrass iteration, which allows a component-wise expression, and thus reduces the computational cost of each iteration. We provide numerical experiments that demonstrate the effectiveness of our method.

1. Introduction
The paper considers systems of multivariate equations which are near systems with multiple roots. The motivation for this work comes from the fact that the main bottleneck of the most commonly used numerical solvers, for example the Newton iteration or the homotopy continuation method, is when the solver runs into a system

*Department of Mathematics, North Carolina State University, Raleigh NC 27695, USA. Research supported by NSF grants CCR-0306406 and CCR-0347506. srpope@ncsu.edu

†Department of Mathematics, North Carolina State University, Raleigh NC 27695, USA. Research supported by NSF grants CCR-0306406 and CCR-0347506. aszanto@ncsu.edu
that is close to one with multiple roots. Since the Jacobian matrix to be inverted
in the Newton iteration is close to being singular, these systems are ill-conditioned.
We combine multivariate interpolation techniques and the algebraic multiplicity
structure of the roots to construct an iterative method which is well defined for
systems with the given multiplicity structure, and we prove that it computes the
roots of the nearest system with the given multiplicity structure.

Our main tool is the multivariate interpolation method developed in [21]. Using
interpolation we were able to generalize to the multivariate case similar results in
the univariate case [34, 33] which use polynomial division. By close inspection of the
interpolation method, we could relate the distance of the input system from the set
of systems with a given multiplicity structure and the minimal 2-norm interpolant
Corresponding to the same multiplicity structure. This relationship resulted in a new
optimization formulation for the above distance, as well as an iterative method which
computes the roots of the nearest system with the given multiplicity structure. This
approach was presented in [26] for simple roots; in the present paper we generalize
it to the case of roots with multiplicities.

We start our discussion with the univariate case. We address the following problem
in the univariate case:

**Problem 1.1:** Given a polynomial \( f \in \mathbb{C}[x] \), a finite set of exponents \( E \subset \mathbb{N} \), the
number of roots \( m \in \mathbb{N} \), and the multiplicity structure \( l = (l_1, \ldots, l_m) \in \mathbb{N}^m \). Find
\( \tilde{f} \in \mathbb{C}[x] \) and \( z_1, \ldots, z_m \in \mathbb{C} \) such that \( \prod_{i=1}^m (x-z_i)^{l_i} \) divides \( \tilde{f} \), and \( f - \tilde{f} = \sum_{\alpha \in E} c_\alpha x^\alpha \)
with \( \sum_{\alpha \in E} |c_\alpha|^2 \) minimal.

Our first contribution in the univariate case is an explicit formula for the gradient
of the norm square function, extending the results of [34] to the \( m > 1 \) case. The
second contribution in the univariate case is a component-wise formula for the Gauss-
Newton iteration to find the optimum, given here explicitly for the first time.

We also present a simplified iteration formula, which may have independent inter-
est. We show that in the case where \( |E| = \sum_{i=1}^m l_i \), the Jacobian matrix used in the
Gauss-Newton iteration has only one non-zero entry per column if it is expressed in
terms of Hermite interpolation polynomials. This reduces the computations neces-
sary for each Gauss-Newton step. We must note that the simplified iteration does
not minimize \( \sum_{\alpha \in E} |c_\alpha|^2 \), but we do give a description of its fixed points.

Secondly, we present the extension of our results to the multivariate case. Instead
of multiplicities, we must consider a more complicated notion of root multiplicities.
A multiple root of the multivariate system \( (f_1, \ldots, f_N) \) is a root \( z \), along with “tan-
gential conditions” \( \Lambda = \{ \Lambda_1, \ldots, \Lambda_k \} \) where each \( \Lambda_i \) is a differential operator that
vanishes on the defining polynomials at \( z \), i.e. \( \Lambda_j(f_i)|_z = 0 \) for all \( i = 1, \ldots N \) and
\( k = 1, \ldots k \). The multiplicity of the root \( z \) is \( k = |\Lambda| \).

For the method presented in this paper we assume that a description of the multi-
licity structure of all or some of the roots of a nearby system to the input is known.
in advance, and this description is given as a set of vanishing tangential conditions as described above. We refer to the following papers on the subject of computation of the multiplicity structure of multivariate polynomial systems: the theoretical foundations in the exact case go back to the works of Macaulay [17] and Gröbner [7], more recently it was studied for example in [18, 31]; in the approximate case in [2] a method is presented to compute the multiplicity structure from an approximation of a multiple root.

We consider the following problem in the multivariate case:

Problem 1.2: Given $N \geq 1$, $f_1, \ldots, f_N \in \mathbb{C}[x_1, \ldots, x_n]$, $E_1, \ldots, E_N \subset \mathbb{N}^n$ finite sets of exponents, $m \in \mathbb{N}$ the number of roots, and the tangential conditions $\Lambda_1, \ldots, \Lambda_m$, each of them being a finite set of differential polynomials from $\mathbb{C}[\partial_{x_1}, \ldots, \partial_{x_n}]$. Find $\tilde{f}_1, \ldots, \tilde{f}_N \in \mathbb{C}[x_1, \ldots, x_n]$ and $z_1, \ldots, z_m \in \mathbb{C}^n$ such that $\Lambda_{r,j}(\tilde{f}_i)|_{z_r} = 0$ for all $i = 1, \ldots, N$, $r = 1, \ldots, m$, $\Lambda_{r,j} \in \Lambda_j$, moreover $f_i - \tilde{f}_i = \sum_{\alpha \in E_i} c_{\alpha} x^{\alpha}$ with $\sum_{i=1}^N \sum_{\alpha \in E_i} |c_{\alpha}|^2$ minimal.

Our first contribution in the multivariate case is to express the distance of the input from systems with given root multiplicity structure as the least squares norm of the multivariate Weierstrass map, a map from the set of possible roots to the set of systems which have these roots with the given multiplicity structure. Secondly, we give explicit formulas for the gradient of the square norm function. Thirdly, we also give explicit formulas for the Gauss-Newton iteration of the Weierstrass map, however, in the multivariate case these expressions are not given component-wise.

Finally, similarly to the univariate case, we give a simplified version of the iteration, which might be of independent interest. Analogously to the classical Weierstrass map, we use the multivariate Hermite interpolation polynomials in each Gauss-Newton iteration step to transform the Jacobian matrix as a block diagonal matrix. As a consequence, we get a simple component-wise formula for the iteration function. However, this iteration will not converge to the least squares minimum. As our numerical experiments indicate, the simplified method computes roots with the smallest residual value $\sum_{i=1}^N \sum_{r=1}^m |f_i(z_r)|^2$, compared to the non-simplified versions.

We also remark that our results have straightforward extension to a formulation of Problem 2 where $f_1, \ldots, f_N$ are analytic functions in $n$ variables and the sets $E_1, \ldots, E_N$ are finite sets of analytic functions so that the differences $f_j - \tilde{f}_j$ lie in $\text{span}_{\mathbb{C}}(E_j)$ for all $j = 1, \ldots, N$ (see [26] for a similar extension).

Also, our results can be easily extended to the problem of finding the closest system that satisfies a much broader range of constraints. We could find the closest system that share a common root, but each polynomial has a different multiplicity for the root. We could also find the closest system which has a point that obtains specified values on each polynomial (ex. $f_1(x) = 1$ and $f_2(x) = 45$) or has specific derivative values at each point (ex. $f_1'(x) = 2$, $f_2''(x) = -1$).
1.1. Related Work

The literature on solving polynomial systems and systems of analytic functions near singular systems is very rich. Below we give a brief summary of results that are the most related to our method. Without trying to give a complete survey of other results tackling related problems, instead we point to papers that contain good surveys and extended bibliography on these topics.

Our method is similar in spirit to a sequence of results computing the nearest systems which have certain prescribed properties: [8] computes the nearest univariate polynomial with constrained roots, [9] computes the nearest polynomial with real root, [13, 14] computes the nearest pair of polynomials with non-trivial GCD, [34] computes the nearest polynomial with a singular root, [33] computes the nearest polynomial with several singular roots.

In the univariate case, the least squares formulation given here is an extension of the ones in [34, 33] to the more general Problem 1. In [34] the authors consider the $m = 1$ case. They formulate the distance of the input to the set of monic degree $d$ polynomials with a root of multiplicity $l$ as a least squares problem, and they give explicit matrix formulation for the objective function and its gradient. In [33] the author considers the case when $\sum_{i=1}^{m} l_i = \deg(f)$, and the set $E$ is the support of $f$. He also formulates the problem as a least squares problem, although the objective function is not given explicitly. The minimum is computed using Gauss-Newton iteration. He also discusses the question of computing the multiplicity structure, which we do not consider in the present paper. Our subject in the univariate case can also be viewed as a special case of the approximate GCD problem (for a survey see [6, Section 2.12.3]), and in particular the present approach is an extension of the works in [1, 14, 26] where an approximate GCD is sought without multiple roots, while here we seek for a GCD which has roots with given multiplicities.

Another approach handling univariate polynomials near ones with multiple roots is the computation of zero clusters. In [10, 30] the authors compute the individual zeroes of the cluster if a nearby system with multiple roots is given: our method can be used to compute the nearby system with multiplicities. In the papers [32, 4] a complete complexity and convergence analysis is given for computing the center of a small disc containing the cluster. These papers also contain surveys and extended bibliography of the topic. Methods that do not require an initial approximation of the clusters include [27, 12, 33, 11], these methods are adaptations of the exact square-free factorization to the approximate case.

Considering the multivariate case, we are not aware of any work which considers computing the nearest multivariate system with given root multiplicity structure. In the works [24, 22, 23, 2? , 2, 16, 5] an incremental “deflation” algorithm is proposed to replace the original system which had a multiple root with another one which has the same root with multiplicity one. Lecerf [15] shows that the resulting iterator converges quadratically to the root in an appropriate non-Archimedean metric. Recently [5] gives a complete convergence analysis in terms of $\alpha$-theory of the deflation.
method, as well as criterions where to stop the iteration near the cluster. This paper also contains a good survey of numerical techniques to handle roots with clusters of roots. Global techniques – techniques where all roots of a polynomial system are encountered – include the homotopy continuation method of [20, 29, 28], or the approximate radical computation in [11], where an approximation to the center of gravity of every cluster is computed, without initial approximations. The method presented here has the property that it can compute several roots simultaneously, but does not need to compute all the roots. Ruatta in [21] defines a generalization of the Weierstrass iteration which computes several roots simultaneously with given multiplicity structure. As we mentioned earlier, our iteration is an adaptation of the generalized Weierstrass iteration in [21] for the case when the input might not have multiple roots but a nearby system has.

2. Notation

We will use the following notation throughout this paper:

1. We will use := to denote definitions.
2. Vectors will be denoted by \textbf{bold}, and considered column vectors. Ex. 
   \[ \mathbf{v} := (v_1, \ldots, v_n)^T \in \mathbb{C}^n. \]
3. Monomials will be expressed as a vector to a vector power. Ex. \( x^\alpha := \prod_{i=1}^n x_i^{\alpha_i} \) for \( \alpha \in \mathbb{N}^n. \)
4. Conjugates will be denoted by \( \overline{z} \). Adjoint matrices (i.e. conjugate transpose matrices) will be denoted by \( M^* \).
5. A support is a set of exponents \( E := \{\alpha_1, \ldots, \alpha_n\} \) where \( \alpha_i \in \mathbb{N}^n. \) We denote by \( \mathbb{C}[x_1, \ldots, x_n]_E \) to be the set of all polynomials in \( \mathbb{C}[x_1, \ldots, x_n] \) of the form \( \sum_{\alpha \in E} c_\alpha x^\alpha. \)
6. For a complex matrix \( A \), we will denote the Moore-Penrose pseudoinverse by \( A^\dagger. \) If \( A \) has more columns than rows and maximal rank, then \( A^\dagger = A^*(AA^*)^{-1}. \) If \( A \) has more rows than columns and maximal rank, then \( A^\dagger = (A^*A)^{-1}A^*. \)
7. If \( A \) is a matrix/vector with symbolic entries, then we will denote by \( \frac{\partial A}{\partial z} \) the matrix/vector obtained by applying \( \frac{\partial}{\partial z} \) to each entry of \( A. \)

3. Univariate Hermite Interpolation

We give a description of the Hermite interpolation method following the approach in [21]. While the univariate case is a special case of the multivariate case which we describe later in the paper, it is presented here for better understanding.

Root multiplicity structures are given in the form \( l = (l_1, \ldots, l_m) \) where \( l_i \) is the multiplicity of the \( i \)th root. For example, \( l = (3, 1) \) represents 2 roots, the first of
which has multiplicity 3 and the second is a simple root. In [34], they consider the case of the nearest polynomial with a single multiple root. In our notation, a single root of multiplicity $M$ would be represented by $(M)$.

In the next definition we introduce the ingredients of the univariate Hermite interpolation.

**Definition 3.1:** Let $\mathbf{z} := (z_1, \ldots, z_m) \in \mathbb{C}^m$ fixed. For a given multiplicity structure $\mathbf{l} := (l_1, \ldots, l_m)$ and a support $E := \{\alpha_1, \ldots, \alpha_t\} \subset \mathbb{N}$, we define a generalized Vandermonde matrix

$$V_{E, \mathbf{l}}(\mathbf{z}) := \begin{bmatrix}
  z_1^{\alpha_1} & \cdots & z_1^{\alpha_t} \\
  \alpha_1 z_1^{\alpha_1 - 1} & \cdots & \alpha_t z_1^{\alpha_t - 1} \\
  \vdots & \ddots & \vdots \\
  \frac{\alpha_1!}{(\alpha_1 - l_1 + 1)!} z_1^{\alpha_1 - l_1 + 1} & \cdots & \frac{\alpha_t!}{(\alpha_t - l_1 + 1)!} z_1^{\alpha_t - l_1 + 1} \\
  \vdots & \ddots & \vdots \\
  \frac{\alpha_1!}{(\alpha_1 - l_m + 1)!} z_m^{\alpha_1 - l_m + 1} & \cdots & \frac{\alpha_t!}{(\alpha_t - l_m + 1)!} z_m^{\alpha_t - l_m + 1}
\end{bmatrix}.$$  

If we denote $d := \sum_{i=1}^m l_i$ then $V_{E, \mathbf{l}}(\mathbf{z})$ has size $d \times t$. We assume that $t \geq d$ for the rest of the paper.

We define the set

$$\Delta_{E, \mathbf{l}} := \{\mathbf{z} \in \mathbb{C}^m \mid \text{rank } V_{E, \mathbf{l}}(\mathbf{z}) < d\}.$$  

If it is clear from the context, we will omit $\mathbf{z}$, $E$ and $\mathbf{l}$ from $V := V_{E, \mathbf{l}}(\mathbf{z})$ and $\Delta := \Delta_{E, \mathbf{l}}$.

Note that if $\mathbf{z} \in \mathbb{C}^m - \Delta$ then the Moore-Penrose pseudoinverse of $V$ is given by

$$V^\dagger = V^\ast (VV^\ast)^{-1}.$$  

We define a set of unit column vectors $\mathbf{u}_{i,j}$ as follows:

$$\mathbf{u}_{i,j} := \begin{bmatrix}
  0 \cdots 0 \\
  \underbrace{1} \cdots \underbrace{0} \\
  \vdots \cdots \vdots \\
  \underbrace{0} \cdots \underbrace{0}
\end{bmatrix}^T.$$  

Each $\mathbf{u}_{i,j}$ consists of $m$ blocks. The $k$th block of $\mathbf{u}_{i,j}$ has $l_k$ rows. The $i$th block has a 1 in the $j$th position.

Define the vector of monomials $x_E := (x^{\alpha_1}, \ldots, x^{\alpha_t})^T$.

We define the Hermite basis polynomials by

$$h_{i,j}(\mathbf{z}, x) := x_E^T V^\dagger \mathbf{u}_{i,j}. \quad (1)$$

for all $1 \leq i \leq m$ and $1 \leq j \leq l_i$. 
The next proposition is a straightforward consequence of the definition of the Hermite basis polynomials.

**Proposition 3.1:** Given \( z = (z_1, \ldots, z_m) \in \mathbb{C}^m, l = (l_1, \ldots, l_m), E = (\alpha_1, \ldots, \alpha_t) \) and \( V = V_{E,1}(z) \) as in Definition 3.1. Assume that the matrix \( V \) has full rank. Then the Hermite basis polynomials defined in (1) satisfy:
\[
\frac{\partial^k h_{i,j}}{\partial x^k}(z, z_r) = \begin{cases} 
1, & \text{if } i = r \text{ and } j = k - 1; \\
0, & \text{otherwise.}
\end{cases}
\] (2)
for all \( 1 \leq i, r \leq m, 1 \leq j \leq l_i \) and \( 1 \leq k \leq l_r \).

Next we define the Hermite interpolation of a polynomial \( f \in \mathbb{C}[x] \) with respect to a fixed multiplicity structure \( l \) and support \( E \).

**Definition 3.2:** Let \( l = (l_1, \ldots, l_m), E = (\alpha_1, \ldots, \alpha_t) \) and \( z = (z_1, \ldots, z_m) \in \mathbb{C}^m - \Delta \) be given as in Definition 3.1. Let \( h_{i,j}(z, x) \) be as in (1). For a given \( f \in \mathbb{C}[x] \) we define the Hermite interpolation polynomial
\[
F(z, x) := \sum_{i=1}^{m} \sum_{j=1}^{l_i} f^{(j-1)}(z_i) h_{i,j}(z, x).
\]

The following proposition gives the main properties of the Hermite interpolation polynomials.

**Proposition 3.2:** With \( f, m, l, E, z, \) and \( F(z, x) \) defined as above, the following properties hold:
1. \( F^{(j)}(z, z_i) = f^{(j)}(z_i) \) for all \( 1 \leq i \leq m \) and \( 0 \leq j \leq l_i - 1 \).
2. \( F(z, x) \) is a polynomial with support \( E \).
3. \( \|F(z, x)\|_2 \) is minimal among all polynomials with properties 1 and 2.
4. A polynomial having properties 1, 2 and 3 is uniquely determined and is equal to \( F(z, x) \).

**Proof:** 1. Let \( F \in \mathbb{C}^t \) be the coefficient vector of \( F \) in the monomial basis \( \{x^{\alpha_1}, \ldots, x^{\alpha_t}\} \). Define also the evaluation vector \( \mathbf{f} \) for \( f \), namely
\[
\mathbf{f} := (f(z_1), \ldots, f^{(l_1-1)}(z_1), f(z_2), \ldots, f^{(l_m-1)}(z_m))^T,
\] (3)
and the vector of monomials \( \mathbf{x}_E = (x^{\alpha_1}, \ldots, x^{\alpha_t}) \) as before. It suffices to note that \( V_{E,1}(z)\mathbf{F} = \mathbf{f} \), so \( F = \mathbf{x}_E\mathbf{F} \) has the desired property.
2. Since each \( h_{i,j} \) has support \( E \), \( F \) also has support \( E \).
3. Note that \( \mathbf{F} = V^\dagger \mathbf{f} \), and the claim is a basic property of the Moore-Penrose pseudoinverse (see [19]).
4. The uniqueness of least-squares solutions is proved in [19]. \( \square \)
4. Univariate Weierstrass Map

In this section, we introduce the univariate Weierstrass map, originally defined in [21], using the Hermite interpolation polynomials defined in the previous section. We prove that the minimal 2-norm of the Weierstrass map equals the distance of the input polynomial from the set of polynomials with the given root multiplicity structure. As we mentioned in the introduction, the Weierstrass map considered here is a generalization of the map defined in [33] in that we allow for the perturbation of coefficients corresponding to an arbitrary set of monomials, while in [33] only perturbation of all the non-leading coefficients of \( f \) is allowed.

**Definition 4.1:** Let \( f, l, E, F, \Delta, \) and \( m \) be as in Definition 3.2. We define the univariate Weierstrass map as

\[
W : \mathbb{C}^m - \Delta \to \mathbb{C}[x]_E \\
z \mapsto F(z, x)
\]

The following proposition gives the main properties of the Weierstrass map.

**Proposition 4.1:** Let \( f, l, E, V, F, \Delta, \) and \( m \) be as above. The univariate Weierstrass map \( W \) defined in Definition 4.1 has the following properties:

1. \( W(z) = 0 \) for some \( z = (z_1, \ldots, z_m) \in \mathbb{C}^m - \Delta \) if and only if \( z_1, \ldots, z_m \) are roots of \( f \) with multiplicities \( l_1, \ldots, l_m \), respectively.
2. \( \|W(z)\|_2^2 = f^* M^{-1} f \) where \( M := V V^* \) and \( f \) is defined in (3).
3. If \( \min_{z \in \mathbb{C}^m - \Delta} \|W(z)\|_2 \) exists then it is equal to the distance of \( f \) from the set of polynomials \( \tilde{f} \) such that \( \tilde{f} \) has the multiplicity structure \( l \) and is obtained by perturbation of the coefficients of \( f \) corresponding to the support \( E \).

**Proof:** We prove this in a more general setting in Proposition 6.1 below. \( \square \)

We follow the approach of Zhi and Wu [34] and express the roots of the gradient of the function

\[
\|W\|^2 : \mathbb{C}^m - \Delta \to \mathbb{R}
\]

If \( z_i = a_i + i b_i \) for \( i = 1, \ldots, m \), then \( \|W\|^2 \) can be considered a real function \( \|W\|^2 : (\mathbb{R}^2)^m - \Delta \to \mathbb{R} \), and its gradient is a \( 1 \times 2m \) vector. However, since one can formally define (see, for instance, [?])

\[
\frac{\partial \|W\|^2}{\partial z_i} = \frac{1}{2} \left( \frac{\partial \|W\|^2}{\partial a_i} - i \frac{\partial \|W\|^2}{\partial b_i} \right),
\]

\[
\frac{\partial \|W\|^2}{\partial \bar{z}_i} = \frac{1}{2} \left( \frac{\partial \|W\|^2}{\partial a_i} + i \frac{\partial \|W\|^2}{\partial b_i} \right)
\]

we can also consider the roots of \( \frac{\partial \|W\|^2}{\partial z_i} \) for \( i = 1, \ldots, m \).

The next theorem and corollaries are needed to give explicit formulas for the gradient of \( \|W\|^2 \), as well as for the Jacobian matrix of the Weierstrass map, which we will use later.
Theorem 4.2: Let $f$, $E$, $m$, $l$, $d$, $z$, $h_{i,j}$, $V$ and $F$ be as in Definitions 3.1. Let $F$ be the coefficient vector of $F$, and $f$ be the evaluation vector of $f$. If $V$ is of maximal rank, then

$$\frac{\partial F}{\partial z_i} = V^\dagger \left( \frac{\partial f}{\partial z_i} - \frac{\partial V}{\partial z_i} F \right) + \left( I - V^\dagger V \right) \left( \frac{\partial V^*}{\partial z_i} (VV^*)^{-1} f \right)$$

Proof: By definition, $F$ is the least squares solution to

$$VF = f.$$  \hfill (4)

We can solve this system using the Moore-Penrose pseudoinverse of $V$, namely $V^\dagger = V^* (V V^*)^{-1}$. So equation (4) is actually solved in two steps. First we find $g$ such that

$$VV^* g = f$$ \hfill (5)

then we compute $F$ as

$$F = V^* g.$$ \hfill (6)

From equation (6) we have

$$\frac{\partial F}{\partial z_i} = \left( \frac{\partial V^*}{\partial z_i} \right) g + V^* \left( \frac{\partial g}{\partial z_i} \right)$$ \hfill (7)

From equation (5) we have

$$\frac{\partial g}{\partial z_i} = (V V^*)^{-1} \left( \frac{\partial f}{\partial z_i} - \left( \frac{\partial V}{\partial z_i} \right) V^* g - V \left( \frac{\partial V^*}{\partial z_i} \right) g \right)$$ \hfill (8)

Combining equations (8) and (7) and simplifying gives

$$\frac{\partial F}{\partial z_i} = V^\dagger \left( \frac{\partial f}{\partial z_i} - \frac{\partial V}{\partial z_i} F \right) + \left( I - V^\dagger V \right) \left( \frac{\partial V^*}{\partial z_i} (VV^*)^{-1} f \right)$$

\hfill \square

Corollary 4.1: Let $f$, $E$, $m$, $l$, $d$, $z$, $h_{i,j}$ and $F$ be as in Definitions 3.1 and 3.2. The entries of $V$ are analytic in $z_i$ therefore

$$\frac{\partial F}{\partial z_i} = V^\dagger \left( \frac{\partial f}{\partial z_i} - \frac{\partial V}{\partial z_i} F \right)$$

Proof: This follow immediately from theorem (4.2) the fact that $\frac{\partial V^*}{\partial z_i} = 0$ if $V$ is analytic in $z_i$. \hfill \square

Corollary 4.2: Let $f$, $E$, $m$, $l$, $d$, $z$, $h_{i,j}$ and $F$ be as in Definitions 3.1 and 3.2. Let $\pi$ be the projection

$$\pi : \mathbb{C}[x]_E \rightarrow \text{span}_\mathbb{C}\{h_{i,j}\}_{i=1,j=1}^m$$

Then

$$\pi \frac{\partial F}{\partial z_i} = \left( f^{(l_i)}(z_i) - F^{(l_i)}(z, z_i) \right) h_{i,l_i}(z, x)$$ \hfill (9)
Proof: This follows from the fact that

\[ V \frac{\partial F}{\partial z_i} = \frac{\partial f}{\partial z_i} - \frac{\partial V}{\partial z_i} F. \]

Because of property 1 of proposition (3.2) and with \( u_{i,j} \) as defined in Definition 3.1,

\[ V \frac{\partial F}{\partial z_i} = \left( f^{(l_i)}(z_i) - F^{(l_i)}(z, z_i) \right) u_{i,l}. \]

Equivalently,

\[ \pi \frac{\partial F}{\partial z_i} = \left( f^{(l_i)}(z_i) - F^{(l_i)}(z, z_i) \right) h_{i,l}(z, x). \]

The next proposition extends the results of [34, Theorem 1] to the case when \( m > 1 \). Before stating the proposition we need some definitions.

**Definition 4.3:** Let \( E = \{\alpha_1, \ldots, \alpha_t\}, 1, z, V, M, f \) and \( F \) be as in Definition 3.1 and Proposition 4.1. For \( i = 1, \ldots, m \) we define

- \( V_i \) to be the matrix obtained from \( V \) by deleting the row \( \frac{\partial^{l_i - 1}}{\partial z_i^{l_i - 1}}(z_i^{\alpha_1}, \ldots, z_i^{\alpha_t}) \), i.e. \( V_i \) is the Vandermonde matrix for the multiplicity structure \( (l_1, \ldots, l_i - 1, \ldots, l_m) \);
- \( f_i \) to be the vector obtained from \( f \) by deleting the entry \( f^{(l_i - 1)}(z_i) \);
- \( M_i := V_i V_i^* \);
- \( F_i(z, x) \) to be the polynomial with coefficient vector \( F_i := V_i f_i \), i.e. \( F_i \) interpolates \( f \) in \( z \) with respect to the multiplicity structure \( (l_1, \ldots, l_i - 1, \ldots, l_m) \).

**Proposition 4.2:** Using the notation of Definition 4.3, for each \( i = 1, \ldots, m \) the partial derivative \( \frac{\partial \|W\|^2}{\partial z_i} \) is equal to

\[ \frac{\det M_i}{\det M} \left( f^{(l_i)}(z_i) - F^{(l_i)}(z_i) \right) \left( f^{(l_i - 1)}(z_i) - F^{(l_i - 1)}(z_i) \right). \] (10)

Proof: The proof is similar to the one in [34, Theorem 1], so we will skip some of the details. By Proposition 4.1 we have that \( \|W\|^2 = f^* M^{-1} f \). Let \( D \) be the \( d \times m \) matrix with entries defined by

\[ D_{j,k} = \begin{cases} 
   f^{(l_j)}(z_i) - F^{(l_j)}(z, z_i) & \text{if } j = \sum_{t=1}^k l_t \\
   0 & \text{otherwise.} 
\end{cases} \] (11)

and let \( D_{+,i} \) be the \( i \)-th column of \( D \). Let \( g \) be defined as in the proof of Theorem 4.2. We have

\[ \frac{\partial \|W\|^2}{\partial z_i} = \frac{\partial f^* M^{-1} f}{\partial z_i} = \frac{\partial f^* g}{\partial z_i}. \] (12)
Since \( f \) is analytic, \( \frac{\partial f}{\partial z_i} = 0 \) and we have (12) is equal to
\[
f^* \frac{\partial g}{\partial z_i} = f^* (VV^*)^{-1} \left( \frac{\partial f}{\partial z_i} - \frac{\partial V}{\partial z_i} \right) F
\]
or equivalently
\[
\frac{\partial f^* M^{-1} f}{\partial z_i} = f^* M^{-1} D_{*,i}.
\]
Since the only non-zero entry in \( D_{*,i} \) is in row \( j_i := \sum_{r=1}^l \tau_r \) and is equal to \( f(l_i - 1) - F(l_i)(z_i) \), it suffices to show that the \( j_i \)-th entry in \( f^* M^{-1} \) is equal to the conjugate of \( \frac{\det M_i}{\det M} (f(l_i - 1)(z_i) - F(l_i)(z_i)) \).

But this follows from the fact that \( F(l_i - 1)(z_i) = v_{j_i} V_i^* M_i^{-1} f_i = m_{j_i} M_i^{-1} f_i \), where \( v_{j_i} \) is the \( j_i \)-th row of \( V \) and \( m_{j_i} \) is the \( j_i \)-th row of \( M_i \). The rest of the proof is the same as in the proof of [34, Theorem 1].

Finally, we give a conjecture, which was proved in the \( m = 1 \) case in [34].

**Conjecture 4.1:** If \( \tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_m) \in \mathbb{C}^m - \Delta \) is a local minimum of \( \|W\|^2 \) then \( f(l_i - 1)(\tilde{z}_i) - F(l_i)(\tilde{z}_i) = 0 \) and \( f(l_i)(\tilde{z}_i) - F(l_i)(\tilde{z}_i) \neq 0 \) for all \( i = 1, \ldots, m \).

Note that if \( d = |E| \) then the condition \( z \in \mathbb{C}^m - \Delta \) implies that \( V = V_{E,1}(z) \) is a square invertible matrix. In this case the Weierstrass map is a complex analytic function, and we can use the approach of [33] applying the Gauss-Newton iteration to get an algorithm which computes a local minimum of the Weierstrass map. If \( |E| > d \) and \( z \in \mathbb{C}^m - \Delta \), then \( V \) has more columns than rows and has full rank.

We next make an observation that is useful in determining the local convergence property of a Gauss-Newton map applied to \( W \).

Let \( J(z) \) be the Jacobian of \( W \) and let \( J_{*,i} \) be the \( i \)-th column of \( J(z) \) (derived in theorem 4.2). We have
\[
J_{*,i}^* W = \left( \frac{\partial f}{\partial z_i} - \frac{\partial V}{\partial z_i} \right)^* (VV^*)^{-1} V F
\]
\[
\frac{\partial \|W\|^2}{\partial z_i}
\]
\[
(\partial \|W\|^2)_{*,i} = (\partial \|W\|^2)_{*,i} = (\partial \|W\|^2)_{*,i}.
\]

If \( \tilde{z} \) is a local minimum of \( \|W(z)\|^2 \) then \( \frac{\partial \|W\|^2}{\partial z_i} = 0 \) for \( i = 1, \ldots, m \). Therefore \( J(\tilde{z})^* W(\tilde{z}) = 0 \). (see also [33, Lemma 2.5]). The Gauss-Newton iteration for \( W \) is
\[
z_{new} = z - J(\tilde{z}) W(z)
\]
and locally converges to \( \tilde{z} \) satisfying \( J(\tilde{z})^*W(\tilde{z}) = 0 \), assuming that \( J(\tilde{z}) \) has full rank (see [3]). Note that by Proposition 4.2 we have that the entries of \( J(\tilde{z})^*W(\tilde{z}) \) are equal to the conjugate of the expression in (10), and if \( J(\tilde{z}) \) has full rank then \( f^{(l)}(\tilde{z}_i) - F^{(l)}(\tilde{z}_i) \neq 0 \) for all \( i = 1, \ldots, m \).

The next theorem is the main result of this section, giving a component-wise expression for the Gauss-Newton iteration in (16).

**Theorem 4.4:** Let \( f, m, l = (l_1, \ldots, l_m), \Delta, F, M \) and \( W \) be as above. For \( z = (z_1, \ldots, z_m) \in C^m - \Delta \) assume that the Jacobian \( J(z) \) of \( W \) has full rank. Then the Gauss-Newton iteration for \( W \) has the following component-wise expression:

\[
z_{j}^{\text{new}} = z_{j} - \frac{\det(N_j)}{\det(M) \det(N)(f^{(l_j)}(z_j) - F^{(l_j)}(z_j))}
\]

where \( N \) is the \( m \times m \) submatrix of \( M^{-1} \) with rows and columns indexed by \( \sum_{s=1}^{i} l_s \) for \( i = 1, \ldots, m \), and \( N_j \) is the matrix obtained from \( N \) by replacing its \( j \)-th column by the vector

\[
P := \left[ \det(M_i) \left( f^{(l_i-1)}(z_i) - F^{(l_i-1)}(z_i) \right) \right]_{i=1,\ldots,m}^T.
\]

Here \( M_i \) and \( F_i \) are defined in Definition 4.3.

**Proof:** Using the definition of \( M \) and \( F \) and Corollary 4.2 and also the matrix \( D \) defined in equation (11) we get that (16) is equal to

\[
z^{\text{new}} = z - (D^*M^{-1}D)^{-1}D^*M^{-1}f
\]

where \( D \) is the matrix defined in (11) and \( f \) was defined (3). Let \( \tilde{D} \) be the \( m \times m \) diagonal matrix consisting of the non-zero rows of \( D \). Then we have

\[
D^*M^{-1}D = \tilde{D}^*N\tilde{D}
\]

and the condition that the Jacobian \( J(z) \) has full rank is equivalent to both \( \tilde{D} \) and \( N \) being invertible. Therefore,

\[
(D^*M^{-1}D)^{-1}D^*M^{-1}f = \frac{1}{\det(M)} \tilde{D}^{-1}N^{-1}P.
\]

Note that we used the fact that the entry of \( M^{-1}f \) indexed by \( j_i := \sum_{s=1}^{i} l_s \) is equal to \( \frac{\det(M_i)}{\det(M)} \left( f^{(l_i-1)}(z_i) - F^{(l_i-1)}(z_i) \right) \) (see the proof of Proposition 4.2). The claim follows from Cramer's rule applied for the \( j \)-th entry of \( \tilde{D}^{-1}N^{-1}P \). \( \square \)

**Remark:** Because of the multiple determinate calculations in the component wise formula, it is actually more efficient to compute the Jacobian for the iteration using Theorem (4.2). If we use the assumption that the number of support elements and constraints are equal, Corollary (4.2) gives us an even more efficient way to construct the Jacobian.
4.1. Simplified Iteration

The classical Weierstrass iteration is the Newton iteration applied to the corresponding Weierstrass map, described in [25]. The Jacobian of the Weierstrass map is diagonal if one expresses it in terms of the standard Lagrange polynomial basis at the iteration point. Doing so allows for a simple, componentwise iteration formula.

Analogously to the classical Weierstrass iteration, we can use the fact that the Jacobian of \( W \) is easily expressed in the basis of Hermite interpolation polynomials to simplify the Gauss-Newton iteration. (See Corollary 4.2). Also, in the basis of Hermite polynomials, the coefficient vector \( F \) is the evaluation vector \( f \) (since \( VF = f \)). The Jacobian is the matrix with \( j \)th column

\[
D_j = (f^{(l,j)}(z_j) - F^{(l,j)}(z, z_j)) u_{j,l}
\]

So, in the Hermite basis the iteration is

\[
\begin{align*}
    z_{\text{new}} &= z - D^\dagger f. \\
    \text{(17)}
\end{align*}
\]

We simplify the iteration even further if we choose exactly the number of support elements to make the Vandermonde system square. We test the simplified iteration in the Numerical Results section under this restriction to the number of support elements.

We must note that by changing the polynomial basis, we have changed the norm that we are minimizing. In general, the iteration in (17) will converge to a different point than the iteration of (16). In the exact case, however, the fixed points of (17) are the same as the fixed points of (16) since \( \|W(z)\|^2 = 0 \) if and only if all entries of \( W \) are zero, regardless of the norm used.

The following proposition describes the fixed points of simplified iteration.

**Proposition 4.3:** Let \( f, m, l, z, \Delta \) and \( F(z, x) \) be defined as in Proposition 3.1. If there exists an open neighborhood, \( U \subset \mathbb{C}^m - \Delta \), about the point \( z \in \mathbb{C}^m - \Delta \) such that for all \( \tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_m) \in U \)

\[
\sum_{i=1}^{m} |F^{(s-1)}(z, \tilde{z})|^2 \leq \sum_{s=1}^{l} |F^{(s-1)}(\tilde{z}, \tilde{z})|^2
\]

then the iteration function (17) has a fixed point at \( z \).

Even though the simplified iteration does not minimize the 2-norm of the monomial coefficient vector, it is still ”minimal” in the sense of Proposition 4.3.

5. Multivariate Interpolation

In the multivariate case the multiplicity structure of roots of a system can be described by the vanishing of certain partial derivatives of the defining polynomials in the given roots. A theory of the multiplicity structure in the language of dual algebras can be found in [18, 31, 21]. Here we adopt the notation in [21].
NOTATION 5.1: Let $x = (x_1, \ldots, x_n)$ be $n$ variables.

- We use $\Lambda(\partial)$ to denote a polynomial in the partial derivatives $\partial x_i$, i.e. $\Lambda(\partial) \in \mathbb{C}[\partial x_1 \ldots \partial x_n]$.

- For $p \in \mathbb{C}[x_1, \ldots, x_n]$ and $z \in \mathbb{C}^n$, we use $\langle \Lambda(\partial), p \rangle_z$ to denote $\Lambda(\partial)$ applied to the polynomial, $p$, and the result evaluated at $z$.

In the first definition we give the main ingredients of a multivariate Hermite interpolation.

DEFINITION 5.2: Given

- Support $E = \{\alpha_1, \ldots, \alpha_t\} \subset \mathbb{N}^n$;

- $\bar{z} = (z_1, \ldots, z_m) \in (\mathbb{C}^n)^m$;

- The tangential conditions $\Lambda = (\Lambda_1, \ldots, \Lambda_m)$, where each $\Lambda_i = \{\Lambda_{i,1}, \ldots, \Lambda_{i,l_i}\} \subset \mathbb{C}[\partial x_1 \ldots \partial x_n]$ has cardinality $l_i$. We denote $d := \sum_{i=1}^m l_i$.

DEFINITION 5.3: The generalized Vandermonde matrix corresponding to $E$ and $\Lambda$ is the following $d \times t$ matrix:

$$V_{E,\Lambda}(\bar{z}) := \begin{bmatrix}
\langle \Lambda_{1,1}, x_{\alpha_1} \rangle_{z_1} & \cdots & \langle \Lambda_{1,1}, x_{\alpha_t} \rangle_{z_1} \\
\vdots & \ddots & \vdots \\
\langle \Lambda_{1,l_1}, x_{\alpha_1} \rangle_{z_1} & \cdots & \langle \Lambda_{1,l_1}, x_{\alpha_t} \rangle_{z_1} \\
\vdots & \ddots & \vdots \\
\langle \Lambda_{m,1}, x_{\alpha_1} \rangle_{z_m} & \cdots & \langle \Lambda_{m,1}, x_{\alpha_t} \rangle_{z_m} \\
\vdots & \ddots & \vdots \\
\langle \Lambda_{m,l_m}, x_{\alpha_1} \rangle_{z_m} & \cdots & \langle \Lambda_{m,l_m}, x_{\alpha_t} \rangle_{z_m}
\end{bmatrix}.$$  

We assume that $t \geq d$ and define the set

$$\Delta_{E,\Lambda} := \{\bar{z} \in (\mathbb{C}^n)^m : \text{rank} V_{E,\Lambda}(\bar{z}) < d\}.$$  

We may omit $E$, $\Lambda$ and $\bar{z}$ from the notation of $\Delta := \Delta_{E,\Lambda}$ and $V := V_{E,\Lambda}(\bar{z})$.

We define $u_{i,j}$ to be the unit vector associated with $\Lambda_{i,j}$ by

$$u_{i,j} := \begin{bmatrix} 0 \cdots 0 & \frac{1}{l_1} \cdots \frac{1}{l_i} \cdots 0 & \frac{1}{l_{m}} \cdots 0 \end{bmatrix}^T.$$  

For fixed $\bar{z} \in (\mathbb{C}^n)^m - \Delta$ the multivariate Hermite basis polynomials are defined by

$$h_{i,j}(\bar{z}, x) := \bar{x}_E V^T u_{i,j}$$  

for $i = 1, \ldots, m$, $j = 1, \ldots, l_i$, and $\bar{x}_E := (x_{\alpha_1}, \ldots, x_{\alpha_t})$. 


Remark: As we mentioned in the introduction, we only consider here the case where $f$ is a polynomial and the set $E$ correspond to monomials. However, all the propositions are true if we replace the monomials and polynomials with analytic functions. For the notation in the general case see [?].

Remark: Some care must be used to choose support elements that ensure $V$ has maximal rank.

The next proposition is a straightforward generalization of the univariate case, so we omit the proof.

**Proposition 5.1:** Let $E$, $\Lambda$, $\bar{z} \in (\mathbb{C}^n)^m - \Delta$ and $h_{i,j}$ be as in definition 5.2. Then we have

- $h_{i,j} \in \mathbb{C}[x_1, \ldots, x_t]_E$.
- $\langle \Lambda_{r,s}, h_{i,j} \rangle_{\bar{z}} \begin{cases} 1, & \text{if } i = r \text{ and } j = s; \\ 0, & \text{otherwise}. \end{cases}$  \hspace{1cm} (19)

for all $i, r = 1, \ldots, m$, $j = 1, \ldots l_i$ and $\Lambda_{r,s} \in \Lambda_r$.

Next we define the Hermite interpolation of a polynomial $f \in \mathbb{C}[x]$.

**Definition 5.4:** Fix $\bar{z} = (z_1, \ldots, z_m) \in (\mathbb{C}^n)^m - \Delta$, $f \in \mathbb{C}[x]$, support $E \subset \mathbb{N}^n$ and multiplicity structure $\Lambda = (\Lambda_1, \ldots, \Lambda_m)$. We define the multivariate Hermite interpolation polynomial

$$F(\bar{z}, x) := \sum_{r=1}^m \sum_{s=1}^{l_r} \langle \Lambda_{r,s}, f \rangle_{\bar{z}} h_{r,s}(\bar{z}, x)$$ \hspace{1cm} (20)

where $\Lambda_{r,s} \in \Lambda_r$ and $h_{r,s}$ was defined in Definition 5.2).

The next proposition is a straightforward consequence of the definition, so we omit the proof.

**Proposition 5.2:** Let $\bar{z}$, $f$, $E$, $\Lambda$ and $F$ be as in Definition 5.4. Then

1. $F \in \mathbb{C}[x]_E$;
2. $\langle \Lambda_{r,s}, f - F \rangle_{\bar{z}} = 0$ for all $r = 1, \ldots, m$ and $\Lambda_{r,s} \in \Lambda_r$;
3. $F$ is the minimal 2-norm polynomial satisfying 1 and 2.
4. $\|F\|_2^2 = f^* M^{-1} f$ where $M := VV^*$ and

$$f = \left( \langle \Lambda_{1,1}, f \rangle_{z_1}, \ldots, \langle \Lambda_{m,l_m}, f \rangle_{z_m} \right)^T.$$

\hspace{1cm} (21)
We finish this section with a generalization of Theorem (4.2) and Corollary (4.2) to the multivariate case. The proof is generally the same as the univariate case, and we omit it.

**Theorem 5.5:** Let $f, m, \Lambda, z, h_{i,j}, V$ and $F(\vec{z}, x)$ be as in Definition 5.4. Let $\mathbf{F}$ be the coefficient vector of $F$, and $\mathbf{f}$ be the evaluation vector of $f$. If $V$ is of maximal rank and has analytic entries, then

$$
\frac{\partial \mathbf{F}}{\partial z_{i,j}} = V^\dagger \left( \frac{\partial \mathbf{f}}{\partial z_{i,j}} - \frac{\partial V}{\partial z_{i,j}} \mathbf{F} \right)
$$

**Proposition 5.3:** Let $f, \Lambda, E$ and $F(\vec{z}, x)$ be as in Definition 5.4. Let $h_{i,s}$ be the Hermite basis polynomials defined in Definition 5.2 and define the projection

$$
\pi : \mathbb{C}[x]_E \rightarrow \text{span}_C \{h_{i,j}\}_{i=1}^m.
$$

Denote $\vec{z} = (z_1, \ldots, z_m)$ and $z_i = (z_{i,1}, \ldots, z_{i,n})$. Then

$$
\pi \frac{\partial F(\vec{z}, x)}{\partial z_{i,j}} = \sum_{s=1}^{l_i} \langle \partial_x \Lambda_{i,s}, f - F \rangle_{z_i} h_{i,s}(\vec{z}, x)
$$

where $\partial_x \Lambda_{i,s}$ is the multiple of $\partial_x$ and $\Lambda_{i,s}$ in $\mathbb{C}[\partial x_1 \ldots \partial x_n]$.

6. Multivariate Weierstrass Map

In this section we generalize the univariate Weierstrass map to the case when we are given $N \geq 1$ polynomials $f_1, \ldots, f_N \in \mathbb{C}[x_1, \ldots, x_n]$. Together with the polynomials we are also given $m$ systems of tangential conditions $\Lambda = \{\Lambda_1, \ldots, \Lambda_m\}$, one for each desired common root. Furthermore, we have $N$ sets of supports $E_1, \ldots, E_N$. The goal is to find the minimal 2-norm perturbation polynomials $F_1, \ldots, F_N$ such that there exists an $m$-tuple $\vec{z} = (z_1, \ldots, z_m)$ of points in $\mathbb{C}^n$ so that for each $i = 1, \ldots, N$, $F_i$ is the Hermite interpolant with respect to $f_i, \Lambda, E_i$ and $\vec{z}$, defined in the previous section.

**Notation 6.1:** For each $j = 1, \ldots, m$ we denote

$$
\Lambda_j = \{\Lambda_{j,1}, \ldots, \Lambda_{j,l_j}\},
$$

where $l_j := |\Lambda_j|$ and $\Lambda_{j,k} \in \mathbb{C}[\partial x_1 \ldots \partial x_n]$. We define

$$
d := \sum_{j=1}^m l_j.
$$
For the rest of the paper we assume that

\[ d < \min_i |E_i| \]

which implies that we have to solve an under-constrained system of equations, so it makes sense to talk about the nearest system which satisfies the tangential conditions.

**Definition 6.2:** Let \( f_1, \ldots, f_N, \Lambda \) and \( E_1, \ldots, E_N \) be as above. We define the multivariate Weierstrass map as

\[
W : (\mathbb{C}^n)^m - \Delta \rightarrow \bigoplus_{i=1}^N \mathbb{C}[x]_{E_i}
\]

\[
\vec{z} \mapsto (F_1, \ldots, F_N)
\]

where \( F_i \) is the Hermite interpolation polynomial corresponding to \( f_i, \Lambda, E_i \) and \( \vec{z} \), as defined in Definition 5.4. Here

\[
\Delta := \bigcup_{i=1}^N \Delta_{E_i, \Lambda}
\]

using the notation of Definition 5.2.

The next proposition summarizes the main properties of the Weierstrass map.

**Proposition 6.1:** The Weierstrass map \( W \), as defined in Definition 6.2, has the following properties:

1. \( W(\vec{z}) = 0 \) for some \( \vec{z} = (z_1, \ldots, z_m) \in (\mathbb{C}^i)^m - \Delta \) if and only if

\[
\langle \Lambda_{j,k}, f_i \rangle_{z_j} = 0
\]

for all \( 1 \leq i \leq N, 1 \leq j \leq m \) and \( 1 \leq k \leq l_{i,j} \).

2. \( \|W(\vec{z})\|^2_2 = \sum_{i=1}^N f_i^* M_i^{-1} f_i \) where \( M_i \) and \( f_i \) are defined as in Proposition 5.2(4) for \( f_i, V_{E_i, \Lambda}(\vec{z}) \) and \( E_i \) in the place of \( f, V \) and \( E \).

3. If

\[
\min_{\vec{z} \in (\mathbb{C}^n)^m - \Delta} \|W(\vec{z})\|^2_2
\]

exists then it is equal to the distance of \((f_1, \ldots, f_N)\) from the set of systems \((\tilde{f}_1, \ldots, \tilde{f}_N)\) such that for some \( \vec{\zeta} \in (\mathbb{C}^n)^m - \Delta \) each \( \tilde{f}_i \) satisfies the tangential conditions \( \Lambda \), and is obtained by a perturbation of the coefficients of \( f_i \) corresponding to the support \( E_i \).
Proof: Properties (1) and (2) follow from Proposition 5.2. Property (3) follows from the fact that if the minimum is taken at $\vec{\zeta} \in (\mathbb{C}^n)^m - \Delta$ then the corresponding Weierstrass image $W(\vec{\zeta}) = (F_1(\vec{\zeta}, x), \ldots, F_N(\vec{\zeta}, x))$ will be the perturbations to obtain the nearest system

$$\left(\hat{f}_1, \ldots, \hat{f}_N\right) := \left(f_1 - F_1(\vec{\zeta}, x), \ldots f_N - F_N(\vec{\zeta}, x)\right)$$

satisfying the conditions in the claim. \qed

First we give explicitly the gradient of the map

$$\|W\|^2 : (\mathbb{C}^n)^m - \Delta \rightarrow \mathbb{R}.$$  

Similarly as in the univariate case we can consider the vanishing of the partial derivatives $\frac{\partial \|W\|^2}{\partial z_{i,j}}$ instead of the partial derivatives by the real and imaginary part of $z_{i,j}$.

In order to give explicitly the gradient of the map $\|W\|^2$, we use Proposition 5.3 and the following definition:

**Definition 6.3:** Given $f_1, \ldots, f_N$, $\Lambda_1, \ldots, \Lambda_m$ and $E_1, \ldots, E_N$ as above. Let $F_1, \ldots, F_N$ be as in Definition 6.2. Define the $N d_i \times mn$ matrix

$$D := \begin{bmatrix} D_1 \\ \vdots \\ D_N \end{bmatrix}$$

where $D_i$ has size $d \times mn$ and has the form

$$D_i := \begin{bmatrix} D_{i,1} \\ \vdots \\ D_{i,m} \end{bmatrix}$$

each block $D_{i,j}$ has size $l_{i,j} \times n$ and defined as

$$D_{i,j} := \left[\langle \partial_{x_k} \Lambda_{j,s}, f_i - F_i \rangle_{\mathbb{R}^n}\right]_{1 \leq s \leq l_{i,j}, 1 \leq k \leq n} \quad (22)$$

Let $M_1, \ldots, M_N$ be as in Proposition 6.1. For each $i = 1, \ldots, N$, $j = 1, \ldots, m$ and $s = 1, \ldots, l_{i,j}$ define $F_{i,j,s}$ and $M_{i,j,s}$ to be the modification of $F_i$ and $M_i$ using the modified multiplicity structure

$$(\Lambda_1, \ldots, \Lambda_j - \{\Lambda_{j,s}\}, \ldots, \Lambda_m).$$

Now we are ready to give an explicit formula for the gradient of the map $\|W\|^2$, which is a generalization of Proposition 4.2.
Proposition 6.2: Using the notation of Definition 6.3, the partial derivatives \( \frac{\partial \|W\|^2}{\partial z_{j,k}} \) are equal to
\[
\sum_{i=1}^{N} \sum_{s=1}^{l_{i,j}} \frac{\det M_{i,j,s}}{\det M_i} \langle \partial_{z_k}A_{j,s}, f_i - F_i \rangle_{\bar{z}_j} \cdot (A_{j,s}, f_i - F_{i,j,s})_{\bar{z}_j},
\]
for \( j = 1, \ldots, m \) and \( k = 1, \ldots, n \)

Proof: Similarly to the univariate case, Proposition 5.3 implies that
\[
M_i \frac{\partial M_i^{-1} f_i}{\partial z_{j,k}} = V_i \frac{\partial F_i}{\partial z_{j,k}} = [D_{i,j}]_{*,k}
\]
where \( f_i \) was defined in Proposition 6.1(2), \( F_i \) is the coefficient vector of \( F_i \), \( V_i := V_{E_i,A}(\bar{z}) \) is the Vandermonde matrix and \([D_{i,j}]_{*,k}\) denotes the \( k \)-th column of the matrix \( D_{i,j} \) defined in (22). Therefore
\[
\frac{\partial \sum_{i=1}^{N} f_i^* M_i^{-1} f_i}{\partial z_{j,k}} = \sum_{i=1}^{N} f_i^* M_i^{-1} [D_{i,j}]_{*,k}
\]
and the rest of the argument is the same as in the univariate case. \( \square \)

Next we give methods to find the least squares solution of the multivariate Weierstrass map.

Since the matrix \( D \) of Definition 6.3 is not as simple as in the univariate case, we cannot exploit its structure to give component-wise formula for the Gauss-Newton iteration of the multivariate Weierstrass map. Therefore we only give an explicit formula for the Gauss-Newton iteration in terms of the matrices \( D_i \) and \( M_i \).

Proposition 6.3: Let \( f_1, \ldots, f_N \) and \( W \) be as above. Using the notation above, for \( \bar{z} \in (\mathbb{C}^n)^m - \Delta \) the Gauss-Newton iteration for \( W \) is given by
\[
\bar{z}^{new} = \bar{z} - \left( \sum_{i=1}^{N} D_i^* M_i^{-1} D_i \right)^{-1} \left( \sum_{i=1}^{N} D_i^* M_i^{-1} f_i \right)
\]
where \( M_i \) and \( f_i \) are defined in 6.1(2) and \( D_i \) is defined in Definition 6.3.

Proof: Note that, using Proposition 5.3, we get that the Jacobian matrix of \( W \) is given by
\[
J := \begin{bmatrix}
V_1^T D_1 \\
\vdots \\
V_N^T D_N
\end{bmatrix}
\]
where \( V_i = V_{E_i,A}(\bar{z}) \) is the Vandermonde matrix for \( i = 1, \ldots, N \). Then the claim follows from the definitions of \( J^T, W, M_i \) and \( F_i \). \( \square \)
6.1. Simplified Iteration

Just as in the univariate case, we can consider a simplified version of the Gauss-Newton map given in Proposition 6.3.

From Theorem 5.5 and Proposition 5.3, we know that the Jacobian matrix of our map $\mathcal{W}$ is given by $D$ as defined in Definition 6.3.

If we consider using as our support the Hermite interpolation polynomials with nodes being the current root approximations, we get the following simplified iteration (identical to univariate case):

$$ z^{\text{new}} = z - D^T f. \quad (24) $$

Note that we also pick only enough support polynomials to ensure that the corresponding Vandermonde systems are square.

By changing the support polynomials at each point in the iteration, we do not minimize the perturbations to the coefficients of the monomials with this simplified iteration.

**Proposition 6.4:** Let $f, m, n, l, z, \Lambda, \Delta$ and $F_1(z, x), \ldots, F_N(z, x)$ be defined as in Definition 5.2. If there exists an open neighborhood, $U \subset (\mathbb{C}^n)^m - \Delta$, about the point $z \in (\mathbb{C}^n)^m - \Delta$ such that for all $\tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_m) \in U$

$$ \sum_{i=1}^{N} \sum_{k=1}^{m} \sum_{s=1}^{l} \sum_{j} \left| F_i^{(s-1)}(z, \tilde{z}_k) \right|^2 \leq \sum_{i=1}^{N} \sum_{k=1}^{m} \sum_{s=1}^{l} \sum_{j} \left| F_i^{(s-1)}(\tilde{z}_1, \tilde{z}_k) \right|^2 $$

then the iteration function (24) has a fixed point at $z$.

6.2. Further Generalizations

We must note that our method can easily be modified to handle a much broader collection of problems.

For example, the only assumption that was used on the entries of $V$ was that they were analytic. So one can choose any set of linearly independent analytic functions to serve as the support for the interpolation. Furthermore, the tangential conditions given for each desired common root do not have to the same. Instead of asking what the closest system with a common root of multiplicity 2, we can also also ask what the closest system is that a common root with multiplicity 1 for the first polynomial, but multiplicity 2 for the second.

7. Numerical Results

We generated random test cases in the following way:

1. Two hundred random systems of three polynomials in three variables where generated.
2. Random roots, $x_1, \ldots, x_k$, were generated with $0 \leq \Re(x_i) \leq 1$ and $0 \leq \Im(x_i) \leq 1$. We considered cases where $k = 2$ and $k = 3$.

3. A generalized Vandermonde system was solved with minimal support to ensure the system had the common roots with multiplicity.

4. The resulting system was then perturbed by a random polynomial with known 2-norm, $\sqrt{3} \times 10^{-6}$.

5. Three iteration methods were applied to each system.
   
   (a) The method with complete support. In this case we use the iteration defined in Proposition 6.3. The support consists of all monomials of total degree less than or equal to the total degree of the input polynomials.
   
   (b) The method with minimal support. In this case we again use the iteration from Proposition 6.3. However, the number of support elements is equal to the number of constraints, making the Vandermonde matrices square.
   
   (c) The simplified iteration. In this case we use the iteration defined in equation (24). The number of support elements is equal to the number of constraints, making the Vandermonde matrices square.

6. The initial iterate was chosen to be 0.4 from the original root of the non-perturbed system.

7. If the iteration did not converge, a new initial iterate half the distance from the original root was chosen, and the algorithm was repeated.

8. After 10 tries, if the iteration still failed, then that case was labeled divergent.

To explain our results, the following definitions based on the list above are useful:

- $\mathbf{F}$ is the coefficient vector of the polynomial system that has the desired root stucture.
- $\mathbf{F}_p$ is the coefficient vector of the perturbed system.
- $z$ are the common roots of $\mathbf{F}$.
- $f(x)$ is the evaluation vector for a given polynomial system, $f$ at a set of roots $x$. (see equation (21)).
- $\tilde{z}$ are the computed roots.
- $\tilde{\mathbf{F}}$ is the coefficient vector of the computed solution system.

The columns of figures (7, 7, 7, 7) are defined as follows:

- Method refers to the type of iteration used.
- The ending residual (End Res.) is defined as $\|\mathbf{F}_p(\tilde{z})\|_2$.
- The distance from original is defined as $\|\mathbf{F} - \tilde{\mathbf{F}}\|_2$.
- The average iterations (Avg. Iterations) is defined as the average number of iterations needed. The iteration terminates when an iteration step fails to reduce the residual by at least 1%.


**Figure 1:** 3 polynomials, total degree 5, in 3 unknowns. Looking for closest system with 2 roots of multiplicity 3 each. Test cases were generated by perturbing a system with known root structure by $1.74 \times 10^{-4}$. The residual of the known roots in the perturbed system was $6.288 \times 10^{-4}$ for the complete support case and $4.356 \times 10^{-4}$ for the other cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>End Res. $(10^{-4})$</th>
<th>Distance $(10^{-4})$ from Original</th>
<th>Avg. Iterations</th>
<th>Avg. Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>4.731</td>
<td>1.694</td>
<td>20.82</td>
<td>53.54</td>
</tr>
<tr>
<td>Simplified</td>
<td>2.632</td>
<td>13.34</td>
<td>6.53</td>
<td>6.762</td>
</tr>
<tr>
<td>Square</td>
<td>3.846</td>
<td>1.200</td>
<td>6.18</td>
<td>6.081</td>
</tr>
</tbody>
</table>

**Figure 2:** 3 polynomials, total degree 5, in 3 unknowns. Looking for closest system with 3 roots of multiplicity 3 each. Test cases were generated by perturbing a system with known root structure by $1.74 \times 10^{-4}$. The residual of the known roots in the perturbed system was $7.867 \times 10^{-4}$ for the complete support case and $6.250 \times 10^{-4}$ for the other cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>End Res. $(10^{-4})$</th>
<th>Distance $(10^{-4})$ from Original</th>
<th>Avg. Iterations</th>
<th>Avg. Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>6.351</td>
<td>1.719</td>
<td>24.19</td>
<td>105.3</td>
</tr>
<tr>
<td>Simplified</td>
<td>3.829</td>
<td>33.03</td>
<td>7.665</td>
<td>13.37</td>
</tr>
<tr>
<td>Square</td>
<td>5.832</td>
<td>1.171</td>
<td>6.995</td>
<td>12.41</td>
</tr>
</tbody>
</table>

**Figure 3:** 3 polynomials, total degree 10, in 3 unknowns. Looking for closest system with 2 roots of multiplicity 3 each. Test cases were generated by perturbing a system with known root structure by $1.74 \times 10^{-4}$. The residual of the known roots in the perturbed system was $1.530 \times 10^{-3}$ for the complete support case and $1.732 \times 10^{-4}$ for the other cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>End Res. $(10^{-4})$</th>
<th>Distance $(10^{-4})$ from Original</th>
<th>Avg. Iterations</th>
<th>Avg. Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>11.35</td>
<td>1.721</td>
<td>5.84</td>
<td>216.3</td>
</tr>
<tr>
<td>Simplified</td>
<td>3.152</td>
<td>13.78</td>
<td>7.575</td>
<td>23.02</td>
</tr>
<tr>
<td>Square</td>
<td>4.221</td>
<td>1.016</td>
<td>7.560</td>
<td>22.77</td>
</tr>
</tbody>
</table>

- The average time (Avg. Time) is the average time in seconds it took the given method to complete.
- In the caption of each figure we note the beginning residual. The beginning residual is defined as $\|F_p(z)\|_2$.

In all the test runs, over 70% of test cases converged with an initial iterate that shared only one significant digit with the roots of the original system. The only divergent cases observed occurred in the complete support case of figure (7). In this instance 3.5% of cases were divergent.
Figure 4: 3 polynomials, total degree 10, in 3 unknowns. Looking for closest system with 3 roots of multiplicity 3 each. Test cases were generated by perturbing a system with known root structure by $1.74 \times 10^{-4}$. The residual of the known roots in the perturbed system was $1.725 \times 10^{-3}$ for the complete support case and $6.242 \times 10^{-4}$ for the other cases.

References


