Computing the Nearest Rank-Deficient Matrix Polynomial

Mark Giesbrecht
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
mwg@uwaterloo.ca

Joseph Haraldson
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
jharalds@uwaterloo.ca

George Labahn
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
glabahn@uwaterloo.ca

ABSTRACT

Matrix polynomials appear in many areas of computational algebra, control systems theory, differential equations, and mechanics, typically with real or complex coefficients. Because of numerical error and instability, a matrix polynomial may appear of considerably higher rank (generically full rank), while being very close to a rank-deficient matrix. "Close" is defined naturally under the Frobenius norm on the underlying coefficient matrices of the matrix polynomial. In this paper we consider the problem of finding the nearest rank-deficient matrix polynomial to an input matrix polynomial, that is, the nearest square matrix polynomial which is algebraically singular. We prove that such singular matrices at minimal distance always exist (and we are never in the awkward situation having an infimum but no actual matrix polynomial at minimal distance). We also show that singular matrices at minimal distance are all isolated, and are surrounded by a basin of attraction of non-minimal solutions. Finally, we present an iterative algorithm which, on given input sufficiently close to a rank-deficient matrix, produces that matrix. The algorithm is efficient and is proven to converge quadratically given a sufficiently good starting point. An implementation demonstrates the effectiveness and numerical robustness in practice.

ACM Reference format:
Mark Giesbrecht, Joseph Haraldson, and George Labahn. 2017. Computing the Nearest Rank-Deficient Matrix Polynomial. In *Proceedings of ISSAC '17, Kaiserslautern, Germany, July 25-28, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3087604.3087648

1 INTRODUCTION

We consider the problem of computing the nearest rank-deficient matrix polynomial to an input matrix polynomial in $\mathbb{R}[t]^{n\times n}$, i.e., finding the nearest matrix polynomial which is *singular*, with a determinant that is identically zero. In an exact setting, determining the rank or determinant of a matrix polynomial is more straightforward, and very efficient procedures are available [35]. However, in many applications, including control systems engineering, a transfer function is specified by a matrix polynomial with floating point coefficients. Due to input error and imprecise representations, most such matrix polynomials will be generically of full rank despite possibly being (numerically) very close to a singular one. Instead of computing the rank of the given matrix polynomial exactly, a more natural first question is to ask how far away it is from one that is rank-deficient, and then to find one at that distance. In the case of unstructured matrices with constant entries this problem is solved via the Singular Value Decomposition (SVD). However, in the case of matrix polynomials no equivalent rank revealing factorization has thus far been available. As a first step we consider the problem of finding a nearby singular matrix polynomial, which in many instances is expected to be a more appropriate object for algorithmic consideration.

To proceed, we need to define a reasonable metric on the space of matrix polynomials, for which we use the common Frobenius norm. For $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$, with (i,j) entry $A_{ij} \in \mathbb{R}[t]$, we define the *Frobenius* norm

$$\|\mathcal{A}\|^2 = \|\mathcal{A}\|_F^2 = \sum_{1\le i,j\le n} \|A_{ij}\|^2, \tag{1.1}$$

where, for $a \in \mathbb{R}[t]$, the coefficient 2-norm is defined by

$$a = \sum_{0\le i\le \deg a} a_i t^i,\ \|a\|^2 = \|a\|_2^2 = \sum_{0\le i\le \deg a} a_i^2. \tag{1.2}$$

In this paper we address the following question:

Main Problem: Nearest singular matrix polynomial. *Given $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ of full rank (non-zero determinant), compute $\Delta\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ with $\deg(\Delta\mathcal{A}_{ij}) \le \deg \mathcal{A}_{ij}$ (or similar degree constraints to be specified later), such that $\mathcal{A}+\Delta\mathcal{A}$ is singular (i.e., $\det(\mathcal{A}+\Delta\mathcal{A}) \equiv 0$) with $\|\Delta\mathcal{A}\|$ minimized.*

Note that the above statement is still somewhat ill-posed, and we refine the statement below.

The main results in this paper are:

1. We characterize the geometry of minimal solutions:
   (a) We show that minimal solutions exist. That is, there exists a $\Delta\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ of minimal norm such that $\det(\mathcal{A}+\Delta\mathcal{A}) \equiv 0$ and meeting the required degree constraints on perturbed coefficients.
   (b) We show that minimal solutions are isolated and are surrounded by a non-trivial open neighborhood of non-minimal solutions.
2. We provide efficient algorithms as follows:
   (a) On input $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ sufficiently close to a singular matrix polynomial, we give an iterative scheme which converges to a rank-deficient matrix polynomial at minimal distance, at a provably quadratic rate of convergence.
   (b) We provide a Maple implementation which demonstrates the convergence and numerical robustness of our iterative scheme.

ISSAC '17, Kaiserslautern, Germany

DOI: http://dx.doi.org/10.1145/3087604.3087648

1.1 Previous research

Much of the work in this area has often been done under the moniker of *matrix pencils*, which generally means matrix polynomials but restricted to the linear case. That is, matrix polynomials of the form $\mathcal{A} = A_0 + A_1 t$ for $A_0, A_1 \in \mathbb{R}^{n\times n}$. See [19] for an excellent overview. Non-singular/full rank square matrix polynomials are sometimes referred to as *regular matrix pencils*.

The problem of nearest rank-deficient or singular matrix pencil was posed for linear matrix pencils in [10] and followed up in [9]. The nearest singular matrix polynomial relates to the stability of linear time invariant systems and differential-algebraic equations studied subsequently in [23, 27]. For non-linear matrix polynomials/pencils, previous works rely on embedding a non-linear (degree greater than 1) matrix polynomial into a linear matrix polynomial of much higher order. Theorem 1.1 and Section 7.2 of [19] shows that any regular $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ of degree d, is equivalent to a linear matrix polynomial $\mathcal{B} = B_0 + tB_1$, for $B_0, B_1 \in \mathbb{R}^{nd\times nd}$. However, this equivalence is (obviously) not an isomorphism, nor is it distance preserving. Hence a nearby singular matrix polynomial to $\mathcal{B} \in \mathbb{R}[t]^{nd\times nd}$ (even when constrained to a degree one perturbation) almost certainly does not correspond to a nearby singular matrix polynomial to $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$. Moreover, even if one was to perturb to a rank-reduced matrix within the image of the linearization, the inverse image would not necessarily have reduced rank. In [28] a more sophisticated linearization with an eye towards ameliorating this is explored.

In the context of computer algebra the notion of symbolic-numeric algorithms for polynomials has been an active area of research for a number of years, and the general framework of finding nearby instances with a desired algebraic property is being thoroughly explored. Closest to our work here is work on approximate GCD [4, 5, 12], and especially optimization-based approaches employing the Structured Total Least Norm algorithm [25, 26, 29, 36] and Riemannian SVD [8]. More recently, we have explored computing the approximate GCRD of (non-commutative) differential polynomials [17, 18] and resolve similar issues.

The computer algebra community has made impressive progress on fast, exact algorithms for matrix polynomials, including nearly optimal algorithms for computing ranks, factorizations and various normal forms; see [24] and references therein for a recent overview. Part of our goal in this current paper is establish a basis for extending the reach of these symbolic techniques to matrices of polynomials with floating point coefficients.

In a more general setting our problem can be formulated as a Structured Low Rank Approximation (SLRA) problem. A popular method to solve SLRA problems is the Structured Total Least Norm (STLN) approach [32, 33]. These are iterative methods and in general their convergence to stationary points is linear (first order), rather than quadratic, unless additional assumptions are made. In the event STLN converges to a solution, there may be other solutions arbitrarily nearby unless the Hessian is definite. The SLRA problem is a non-linear least squares problem and accordingly other techniques such as the Restricted and Riemannian SVD [13–15] provide general tools for solving such problems. Other heuristic tools applicable to our problem include variable projection [20, 21] and Newton's method [1]. We would expect these to perform very poorly in our case, as one can expect problems with large residuals to perform poorly and the rational function arising from variable projection can be too costly to deal with for modestly sized problems. The problem may also be considered as optimization on a manifold [2], however we do not explicitly consider this approach. For a detailed survey of linear structured low-rank approximation, see [30, 31].

Other methods for structured low-rank approximation involve the family of lift and project algorithms, with the best known being Cadzow's algorithm [11]. More recently [34] gives a sequence of alternating projections that provably converge quadratically to a fixed point. However, lift and project algorithms do not generally satisfy necessary first order optimality conditions, and while they may converge (quickly) to a fixed point, there is no guarantee that the fixed point is an optimal solution, though it is usually quite good. In any case, for specific problems such as ours, understanding the geometry of the minimal solutions (and hence the well-posedness of the problem) is key to effective algorithms for their computation.

A related but different problem is *Wilkinson's problem*. Given a linear matrix pencil $A_0 + tA_1$, it seeks the nearest matrix pencil which is *defective*, or such that it does not have complete basis of (generalized) eigenvectors. More generally, for a matrix polynomial $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ a defective matrix pencil is one whose Smith form is non-trivial, that is, does not equal $\mathrm{diag}(1, \ldots, 1, \det(\mathcal{A}))$. A number of techniques have been developed to find the nearest defective linear matrix pencil, see [3] for recent results and a survey. While this problem has a related (and somewhat more difficult) geometry, we hope that our techniques can be extended to this problem for non-linear matrix polynomials.

1.2 Outline

In Sections 2 and 3 we describe tools needed for our constructions and then explore the geometry of our problem. We show that the problem is locally well-posed. One cannot expect the nearest singular matrix polynomial to be unique. However under weak normalization assumptions, we show that solutions are locally unique in a closed-ball around them. To complement the separation of solutions, we also show that solutions corresponding to a different closed ball are separated by at least a constant amount independent of the dimension of the space.

In Section 4 we give an equality constrained variant of Newtons' method for computing via post-refinement the nearest rank-deficient matrix polynomial. The main idea is to compute an initial guess with a suitable first order or lift-and project method. We are able to prove that, with a suitable initial guess and regularity assumptions, our algorithm generally has local quadratic convergence except for degenerate cases. This is done by deriving closed-form expressions for the Jacobian of the constraints and the Hessian of the Lagrangian. We make explicit use of these closed-forms and prove that they have full rank in non-trivial open neighborhoods around solutions.

In Section 5 we describe our prototype implementation, including heuristics for starting points and other improvements. We discuss the numerical performance of the algorithm and give examples demonstrating convergence. The paper ends with a conclusion and topics for future research.

2 PRELIMINARIES AND GEOMETRY

In this section we will introduce some basic definitions and explore the numerical geometry of our problem. Canonically we will let

$$\mathcal{A} = \sum_{j=0}^{d} A_j t^j \in \mathbb{R}[t]^{n\times n}$$

be a matrix polynomial, with coefficients $A_0, \ldots, A_d \in \mathbb{R}^{n\times n}$. Throughout this paper we will present our results for square matrix polynomials, though much of the theory can be generalized in a straightforward manner for non-square matrix polynomials and complex valued ones. The *degree* $\deg \mathcal{A}$ of $\mathcal{A}$ is defined as d, assuming that $\mathcal{A}_d \not\equiv 0$.

We say that $\mathcal{A}$ is singular if $\det(\mathcal{A}) \equiv 0$ as a polynomial in $\mathbb{R}[t]$, or equivalently, that there is a non-trivial vector $b \in \mathbb{R}[t]^{n\times 1}$ such that $\mathcal{A}b \equiv 0$. The kernel of $\mathcal{A}$ is $\ker \mathcal{A} = \{b \in \mathbb{R}[t]^{n\times 1} : \mathcal{A}b \equiv 0\}$ and the rank of $\mathcal{A}$ is $n - \dim \ker \mathcal{A}$ (as a vector space over $\mathbb{R}(t)$). For $a \in \mathbb{R}[t]$, define

$$\phi(a) = \phi^{(n,d)}(a) = \begin{pmatrix} a_0 & & & \\ a_1 & a_0 & & \\ \vdots & & \ddots & \\ a_d & & & a_0 \\ & a_d & & a_1 \\ & & \ddots & \vdots \\ & & & a_d \end{pmatrix} \in \mathbb{R}^{(\mu+d)\times\mu}, \qquad (2.1)$$

where $\mu = nd + 1$. $\phi(a)$ is a Toeplitz matrix corresponding to polynomial multiplication of $a \cdot b$ where a has degree d and $b \in \mathbb{R}[t]$ has degree at most $\mu - 1$.

Definition 2.1. The $\mathbb{R}$-*embedding* of $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ is

$$\widehat{\mathcal{A}} = \begin{pmatrix} \phi(A_{1,1}) & \cdots & \phi(A_{1,n}) \\ \vdots & & \vdots \\ \phi(A_{n,1}) & \cdots & \phi(A_{n,n}) \end{pmatrix} \in \mathbb{R}^{n(\mu+d)\times n\mu}.$$

For $b \in \mathbb{R}[t]^{n\times 1}$ of degree $\mu - 1$ the $\mathbb{R}$-embedding of b is

$$\widehat{b} = (b_{1,0}, b_{1,1}, \ldots, b_{1,\mu-1}, \ldots, b_{n,0}, \ldots, b_{n,\mu-1})^T \in \mathbb{R}^{n\mu\times 1}.$$

Note that $\mathcal{A}b = 0$, for $b \in \mathbb{R}[t]$ of degree at most $\mu - 1$ if and only if $\widehat{\mathcal{A}}\widehat{b} = 0 \in \mathbb{R}^{n\mu\times 1}$.

For ease of notation we will take $N = n(\mu + d)$, $M = n\mu$ and $R \geq 1$ when dealing with $\mathbb{R}$-embeddings in subsequent sections. We note that $\widehat{\mathcal{A}}$ is a block-Toeplitz matrix, and as such one method of understanding the problem is to find the nearest structured rank deficient block-Toeplitz matrix, a typical structured low rank approximation problem.

Unlike the standard linearizations in [19] (Section 7.2) used to turn arbitrary degree matrix pencils into linear pencils, this $\mathbb{R}$-embedding is rank preserving for matrix polynomials of arbitrary degree. In particular, $b \in \ker \mathcal{A}$ with $\deg b \leq \mu$ implies $\widehat{b} \in \ker \widehat{\mathcal{A}}$. The $\mathbb{R}$-embedding is also quasi-distance preserving, since $\|\mathcal{A}\|_F^2 = \frac{\|\widehat{\mathcal{A}}\|_F^2}{\mu}$.

Problem 2.2. ***Refined Main Problem:***
Given $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ non-singular of degree d as above, determine $\Delta\mathcal{A} \in \mathbb{R}[t]^{n\times n}$, with $\deg(\Delta\mathcal{A})_{ij} \leq \deg \mathcal{A}_{ij}$ for all $1 \leq i, j \leq n$, and $b \in \mathbb{R}[t]^{n\times 1}$, such that $\|\Delta\mathcal{A}\|$ is (locally) minimized, subject to the constraints that $(\mathcal{A} + \Delta\mathcal{A})b = 0$ and $\|b\| = 1$.

Note that this is minimizing a convex objective function subject to non-convex constraints. However, the equality constraints are linear in each argument. It is still not clear that Problem 2.2 is well-posed in the current form. We will prove that solutions exist, that is, there is an attainable global minimum value and not an infimum.

Lemma 2.3. *$\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ is singular if and only if there exists a $b \in \mathbb{R}[t]^{n\times 1}$ with $\deg b \leq nd = \mu - 1$ such that $\mathcal{A}b = 0$.*

Proof. Suppose that $\mathcal{A}$ has rank $r < n$. By permuting rows and columns we may assume without loss of generality that the leading $r \times r$ submatrix of $\mathcal{A}$ is non-singular. There is a unique vector of the form $c = (b_1/\gamma, \ldots, b_r/\gamma, -1, 0, \ldots, 0)$ from Cramer's rule such that $\mathcal{A}c = 0$, where $\gamma \in \mathbb{R}[t]$ is the determinant of the leading $r \times r$ minor of $\mathcal{A}$, and all of $b_1, \ldots, b_r, \gamma \in \mathbb{R}[t]$ have degree at most $rd \leq nd$. Multiplying through by γ, we find that $b = \gamma c$ satisfies the requirements of the lemma. □

See [6, Corollary 5.5] for an alternative proof.

Lemma 2.4. *$\mathcal{A}$ is singular if and only if $\widehat{\mathcal{A}}$ does not have full column rank.*

Proof. If $\mathcal{A}$ is rank deficient then there exists $b \in \mathbb{R}[t]^{n\times 1}$ with $\deg b \leq \mu - 1$ such that $\mathcal{A}b = 0$. $\widehat{\mathcal{A}}$ has a non-trivial kernel and, $\widehat{b} \in \ker \widehat{\mathcal{A}}$ by construction. Conversely, suppose that $\mathcal{A}$ has full rank. Then for all $b \in \mathbb{R}[t]^{n\times 1}$ we have $\mathcal{A}b \neq 0$ which implies that $\widehat{\mathcal{A}}\widehat{b} \neq 0$ or $\ker \widehat{\mathcal{A}}$ is trivial. □

We recall the Singular Value Decomposition as the primary tool for finding the distance to the nearest *unstructured* rank deficient matrix over $\mathbb{R}$ or $\mathbb{C}$.

Definition 2.5. A Singular Value Decomposition (SVD) of $C \in \mathbb{R}^{N\times M}$ is given by $C = Q \cdot \Sigma \cdot P^T$, where $Q \in \mathbb{R}^{M\times M}, P^T \in \mathbb{R}^{N\times N}$ are orthogonal matrices and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_M) \in \mathbb{R}^{M\times N}$ is a diagonal matrix consisting of the singular values of C in descending order of magnitude. See [22].

The following fact is a standard motivation for the SVD.

Fact 2.6 (Eckart and Young 16). *Suppose $C = Q\Sigma P^T \in \mathbb{R}^{N\times M}$ as above has full column rank, with $N \geq M$. Then the matrix $\Delta C = Q \operatorname{diag}(0, \ldots, 0, -\sigma_M)P^T$ is such that $C + \Delta C$ has column rank at most $M - 1$, $\|\Delta C\|_F = \sigma_M$, and ΔC is a perturbation of minimal Frobenius norm which reduces the column rank of C.*

Lemma 2.7. *Given a non-singular $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$, and $\Delta\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ such that $\mathcal{B} = \mathcal{A} + \Delta\mathcal{A}$ is singular, it is the case that $\|\widehat{\Delta\mathcal{A}}\| \geq \sigma_{n\mu}(\widehat{\mathcal{A}})$.*

Proof. By Lemma 2.4 above, $\widehat{\mathcal{B}}$ is not of full column rank. Thus, by Fact 2.6 $\|\widehat{\Delta\mathcal{A}}\|_F \geq \sigma_{n\mu}(\widehat{\mathcal{A}})$. □

Corollary 2.8. *The set of non-singular matrices over $\mathbb{R}[t]^{n\times n}$ of degree at most d is open, or equivalently, the set of all rank deficient matrices over $\mathbb{R}[t]^{n\times n}$ of degree at most d is closed.*

Theorem 2.9 (Existence of Solutions). *The minimization posed in Problem 2.2 has an attainable global minimum if* $\deg \Delta\mathcal{A}_{i,j} \leq \deg \mathcal{A}_{i,j}$ *for all* $1 \leq i,j \leq n$.

Proof. Let

$$S = \left\{\mathcal{B} \in \mathbb{R}[t]^{n\times n} \mid \operatorname{rank}\mathcal{B} \leq n-1 \wedge \deg\mathcal{B} \leq d\right\} \cap \left\{\mathcal{B} \in \mathbb{R}[t]^{n\times n} \mid \|\mathcal{B}\|_F^2 \leq \|\mathcal{A}\|_F^2\right\}.$$

S is the intersection of a closed and bounded set and a closed set, hence S is closed and bounded. S is isomorphic to some closed and bounded subset of Euclidean space, hence by the Heine-Borel theorem, S is compact. To show the set is non-empty, we note that, by the degree assumption on $\Delta\mathcal{A}$, $\Delta\mathcal{A} = -\mathcal{A}$ is a feasible point independent of rank.

Let $\mathcal{B} \in S$ then $\|\mathcal{A} - \mathcal{B}\|_F^2 = \|\Delta\mathcal{A}\|_F^2$ is a continuous function over a compact set. By Weierstrass' theorem it has an attainable global minimum. □

It is important not to over-constrain the problem with a choice of $\Delta\mathcal{A}$, since otherwise the feasible set might be empty. Another reasonable choice of $\Delta\mathcal{A}$ which we can handle, is that the perturbation has the same coefficient structure/support as $\mathcal{A}$, that is, zero terms in polynomial entries are preserved.

We note that this result says nothing about uniqueness or separation of solutions or any local properties. All that has been shown is that if the perturbations are in the same space as the input, and one seeks a rank at-most approximation, then there is an attainable global minimum value, i.e. not an infimum. If one wants a minimal solution with the rank being exactly r, then there is no guarantee that there is an attainable global minimum to Problem 2.2.

3 RANK FACTORIZATION

A natural formulation of the problem that encompasses the rank implicitly is to perform a rank factorization and write $\mathcal{A} + \Delta\mathcal{A} = \mathcal{U}\mathcal{V}$ for $\mathcal{U} \in \mathbb{R}[t]^{n\times r}$ and $\mathcal{V} \in \mathbb{R}[t]^{r\times m}$. Here $\mathcal{U}\mathcal{V}$ is subject to some constraints that preserve the structure of $\Delta\mathcal{A}$ (i.e., that we do not perturb any coefficients we are not allowed to, typically that $\deg\Delta\mathcal{A}_{ij} \leq \deg\mathcal{A}_{ij}$, but possibly also preserving the zero coefficients and not introducing a larger support). This is a non-linear least squares problem. However solutions are not unique. Indeed, if $\mathcal{Z} \in \mathbb{R}[t]^{r\times r}$ is unimodular (i.e., $\det(\mathcal{Z}) \in \mathbb{R}^*$), then $\mathcal{U}\mathcal{Z}$, $\mathcal{Z}^{-1}\mathcal{V}$ is another rank r factorization, and we obtain an infinite family. While normalizing over matrix polynomial rank-factorizations is difficult, it is much easier to exploit the quasi-distance preserving property of $\|\cdot\|_F$ and look at rank-factorizations of $\widehat{\mathcal{A}}$, that do not necessarily correspond to $\mathcal{U}$ and $\mathcal{V}$.

3.1 Embedded Rank Factorization

Definition 3.1. Let $N = (\mu + d)n$, $M = n\mu$ and $R > 0$. A rank factorization of $\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}}$ is given by writing $\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}} = UV$ where $U \in \mathbb{R}^{N\times R}$ and $V \in \mathbb{R}^{R\times M}$ are arbitrary (unstructured) matrices over $\mathbb{R}$.

Our goal is to find U, V with shape as above which minimize

$$\|\Delta\widehat{\mathcal{A}}\| = \|\widehat{\mathcal{A}} - UV\|$$

and such that $\Delta\widehat{\mathcal{A}}$ has the correct Toeplitz-block structure (i.e., it is an $\mathbb{R}$-embedding of a matrix polynomial). This is a problem with a non-convex objective function (that is convex in each argument) and non-convex constraints. We note that U, V have no direct connection with $\mathcal{U}, \mathcal{V} \in \mathbb{R}[t]^{n\times n}$.

One may always write $\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}}$ this way via the SVD for fixed $\widehat{\mathcal{A}}$ and $\widehat{\Delta\mathcal{A}}$, so in particular the optimal solution can be written as a rank factorization. The problem $\min \|\widehat{\mathcal{A}} - UV\|^2$ such that UV has the same structure as $\widehat{\Delta\mathcal{A}}$ is generally ill-posed and needs to be constrained to do any meaningful analysis, as there are numerous degrees of freedom. At first glance, optimizing over rank factorizations appears to be a harder problem than the original. However it is helpful to perform analysis on this formulation. In particular, we are able to prove that optimal values of $\widehat{\Delta\mathcal{A}}$ that satisfy first order conditions (which we will show contains all useful perturbations) are separated by a constant amount, and that equivalence classes of solutions are isolated.

We next need to demonstrate that the condition that the matrix $\Delta\widehat{\mathcal{A}} = \widehat{\mathcal{A}} - UV$ is the $\mathbb{R}$-embedding of some matrix polynomial $\Delta\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ can be phrased as a single polynomial being zero. Clearly each entry in $\Delta\widehat{\mathcal{A}}$ is a bilinear function of the U_{ij} and V_{ij}. Also, each entry $\Delta\mathcal{A}_{ij}$ in $\Delta\mathcal{A}$ is mapped to a Toeplitz block $\phi(\Delta\widehat{\mathcal{A}}_{ij})$ in $\Delta\widehat{\mathcal{A}}$; see (2.1).

- Let T_1 be the sum of the squares of all the entries required to be zero in $\Delta\widehat{\mathcal{A}} - UV$.
- Let T_2 be the sum the squares of the differences of all pairs of entries required to be equal in $\Delta\widehat{\mathcal{A}} - UV$.

We then define the structural enforcement function $\Gamma : \mathbb{R}^{N\times R} \times \mathbb{R}^{R\times M} \to \mathbb{R}$ as $\Gamma(\Delta\widehat{\mathcal{A}}) = T_1 + T_2$. It is easily observed that Γ is a polynomial of degree 4 in the U_{ij} and V_{ij}, that it is always non-negative, and that $\Gamma(\Delta\widehat{\mathcal{A}}) = 0$ if an only if $\Delta\widehat{\mathcal{A}}$ is the $\mathbb{R}$-embedding of a matrix polynomial.

Problem 3.2. *With $\widehat{\mathcal{A}}, U, V$ as above, the constrained $\mathbb{R}$-embedded rank factorization problem consists of computing* $\min \|\widehat{\mathcal{A}} - UV\|_F^2$ *subject to the constraints that* $U^TU - I = 0$ *and* $\Gamma(U,V) = 0$. *If* $R = M - 1$, *then this encodes all rank deficient matrix polynomials.*

It is still not clear that Problem 3.2 is well-posed, as there are many degrees of freedom in V, and this matrix can have arbitrary rank. The enforcement of U as an orthogonal matrix ($U^TU - I = 0$) is allowed for without loss of generality. Informally then we are looking at all rank factorizations where where U is orthogonal and $\Gamma(U,V) = 0$, that is, the product satisfies the block-Toeplitz structure on $\widehat{\Delta\mathcal{A}}$.

We employ the machinery of non-linear optimization to describe the geometry of the minimal solutions, and hence the nearest appropriately structured matrices. See [7] for an excellent overview.

Fact 3.3 (Bertsekas 7, Section 3.1.1). *For a sufficiently large penalty term $\rho > 0$, one has that the unconstrained optimization problem of computing*

$$\Phi(U,V) = \min_{U,V} \|\widehat{\mathcal{A}} - UV\|_F^2 + \rho\|\Gamma(U,V)\|_F^2 + \rho\|U^TU - I\|_F^2$$

is equivalent to Problem 3.2.

All the solutions to the minimization of $\Phi(U,V)$ occur at stationary points. The first-order necessary condition (on V) of gradients vanishing gives us

$$\nabla_V \left(\|\widehat{\mathcal{A}} - UV\|_F^2 + \rho\|\Gamma(U,V)\|_F^2) + \rho\|U^TU - I\|_F^2 \right) = 0$$
$$\iff U^T(\widehat{\mathcal{A}} - UV) + \left(\frac{\partial}{\partial V}\Gamma(U,V)^T\right)\rho\Gamma(U,V) = 0.$$

If we assume that the constraints are active, that is U is orthogonal and that $\Gamma(U,V) = 0$, then we have $U^T\widehat{\mathcal{A}} - V = 0$. Of course, there is the other first order necessary condition requiring that

$$\nabla_U \left(\|\widehat{\mathcal{A}} - UV\|^2 + \rho\|\Gamma(U,V)\|^2 + \rho\|U^TU - I\|^2 \right) = 0.$$

However, we do not need to employ this explicitly in the following.

Theorem 3.4 (Strong Separation of Objective). *Suppose $\widehat{\Delta\mathcal{A}}$ and $\widehat{\Delta\mathcal{A}}^\star$ are distinct (local) optimal solutions to Problem 3.2 that satisfy first order necessary conditions. Then $\|\widehat{\Delta\mathcal{A}} - \widehat{\Delta\mathcal{A}}^\star\|_2 \geq \sigma_{\min}(\widehat{\mathcal{A}})$.*

Proof. From the previously discussed necessary first order condition we have that

$$\|\widehat{\Delta\mathcal{A}} - \widehat{\Delta\mathcal{A}}^\star\|_2 = \|UV - U^\star V^\star\|_2 = \|UU^T\widehat{\mathcal{A}} - U^\star U^{\star T}\widehat{\mathcal{A}}\|_2.$$

From this we can obtain the sequence of lower bounds

$$\|UU^T\widehat{\mathcal{A}} - U^\star U^{\star T}\widehat{\mathcal{A}}\|_2 \geq \|UU^T - U^\star U^{\star T}\|_2\sigma_{\min}(\widehat{\mathcal{A}})$$
$$= \|I - U^TU^\star U^{\star T}U\|_2\sigma_{\min}(\widehat{\mathcal{A}}) \geq \sigma_{\min}(\widehat{\mathcal{A}}).$$

The symmetric matrix $W = U^TU^\star U^{\star T}U$ is a product of matrices whose non-zero eigenvalues have magnitude 1. Symmetric matrices have real eigenvalues, and their non-zero eigenvalues are ± 1, since

$$\|W\|_2 \geq \sigma_{\min}(U)\sigma_{\min}(U^T)\sigma_{\min}(U^{\star T})\sigma_{\min}(U^\star),$$
$$\|W\|_2 \leq \sigma_{max}(U)\sigma_{max}(U^T)\sigma_{max}(U^{\star T})\sigma_{max}(U^\star),$$

which gives us $1 \leq \|W\|_2 \leq 1$.

W must have at least one negative eigenvalue or non-trivial 0 eigenvalue by the orthogonality assumption, since $W \neq I$. It follows that $\|I - W\|_2 \geq 1$ and the theorem follows. □

We note that when U and $U^\star$ have the same dimension, then $\operatorname{rank} U = \operatorname{rank} U^\star$ and so W has full rank. Since W has full rank, it follows that $\|I - W\|_2 \geq 2$ and the lower-bound can be improved. While the separation bound exploited properties of the rank factorization, these bounds hold for all formulations of the problem.

Corollary 3.5. *All locally optimal solutions satisfying first order necessary conditions are isolated modulo equivalence classes.*

Proof. Suppose the contrary, that is that (U,V) is a solution corresponding to $\widehat{\Delta\mathcal{A}}$ and $(U^\star,V^\star)$ is a solution corresponding to $\widehat{\Delta\mathcal{A}}^\star$. The objective function and constraints are locally Lipschitz continuous, so let $s > 0$ be a Lipschitz constant with respect to $\|\cdot\|_F$ in some open neighborhood.

If we take $0 < \varepsilon < \dfrac{\sigma_{\min}(\widehat{\mathcal{A}})}{s}$ then we have

$$\sigma_{\min}(\widehat{\mathcal{A}}) \leq \|\widehat{\Delta\mathcal{A}} - \Delta\widehat{\mathcal{A}}^\star\|_2$$
$$\leq s\left\|\begin{pmatrix}U\\V\end{pmatrix} - \begin{pmatrix}U^\star\\V^\star\end{pmatrix}\right\|_F$$
$$< \sigma_{\min}(\widehat{\mathcal{A}}),$$

which is a contradiction to Theorem 3.4. □

While there are too many degrees of freedom to easily obtain a (locally) quadratically convergent minimization over the rank factorization, the rank factorization does yield non-trivial insights into the geometry of the solution space. In particular, the isolation of solutions indicates first order (gradient) methods will perform well on the problem. In the next section we will introduce a locally quadratically convergent algorithm for an equivalent form of Problem 2.2 that reduces each equivalence class of solutions to a single solution.

4 ITERATIVE ALGORITHM

In this section we propose an iterative algorithm to solve Problem 2.2 based on Newton's method for constrained optimization. Sufficient conditions for quadratic convergence are that the Jacobian of the residuals has full rank, the Hessian matrix of the objective function is positive definite in a neighborhood around a solution [7] and local Lipschitz continuity of the objective and constraints. We ensure this by augmenting the objective function without changing the solution and working on a restricted space of minimal $\mathbb{R}$-embeddings that removes degrees of freedom.

4.1 Augmented System

In order to find the nearest rank deficient matrix, we consider the modified objective function

$$\Psi = \|\widehat{\Delta\mathcal{A}}\|_F^2 + \|\widehat{\mathcal{b}}\|_F^2 - 1,$$

subject to the constraints $(\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{\mathcal{b}} = 0$ and $\widehat{\mathcal{b}}^T\widehat{\mathcal{b}} - 1 = 0$ for $\widehat{\mathcal{b}} \in \mathbb{R}[t]^{M\times 1}$. One notes that this is essentially a trick, because we already have the equality condition that $\|\widehat{\mathcal{b}}\| = 1$. Thus the new formulation is completely equivalent to Problem 2.2 as their solutions are the same. However, the change to the objective function ensures that the Hessian matrix of the objective function is not singular, that is, it is positive definite. In this case the Hessian matrix of Ψ is given by $\nabla^2\Psi = \operatorname{diag}(2,\ldots,2)$.

Definition 4.1 (Minimal $\mathbb{R}$-Embedding). Suppose $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ with $\mathbb{R}$-embedding $\widehat{\mathcal{A}}$. The vector $\mathcal{b} \in \mathbb{R}[t]^{n\times 1}$, with $\mathbb{R}$-embedding $\widehat{\mathcal{b}}$, is said to be *minimally $\mathbb{R}$-embedded* in $\widehat{\mathcal{A}}$ if $\ker\widehat{\mathcal{A}} = \langle\widehat{\mathcal{b}}\rangle$ (i.e., a dimension 1 subspace). We say that $\widehat{\mathcal{b}}$ is *minimally degree $\mathbb{R}$-embedded* in $\widehat{\mathcal{A}}$ if (1) $\widehat{\mathcal{b}}$ is minimally $\mathbb{R}$-embedded in $\widehat{\mathcal{A}}$ and (2) $\mathcal{b} \in \mathcal{B}$ for a given column echelon reduced basis $\mathcal{B}$ of $\ker\mathcal{A}$, where each column of the basis is primitive, that is, $\gcd(\mathcal{B}_{1,j},\ldots,\mathcal{B}_{n,j}) = 1$.

We note that this definition ensures minimally $\mathbb{R}$-embedded vectors are unique, or that $(\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{\mathcal{b}} = 0$ has a (locally) unique solution for fixed $\Delta\widehat{\mathcal{A}}$. If $\widehat{\mathcal{b}}$ is not minimally degree $\mathbb{R}$-embedded in

$\widehat{\mathcal{A}}$, then the kernel of $\widehat{\mathcal{A}}$ will typically consist of the desired vector, cyclic shifts of this vector and multiples of other vectors.

There are two degrees of freedom for elements of the kernel of $\mathcal{A} + \Delta\mathcal{A}$. The first is the degree of entries, that is, polynomial multiples of b are in the kernel of $\mathcal{A}$. The second degree of freedom is that a linear combination of elements of the kernel is also contained in the kernel. The minimal degree $\mathbb{R}$-embedding removes both degrees of freedoms (after normalization). The column reduced echelon constraint ensures that the constraint is represented with the minimum number of equations.

Throughout the rest of this section we will assume that $\widehat{\mathcal{A}}$ and $\widehat{b}$ are minimal degree $\mathbb{R}$-embeddings of $\mathcal{A}$ and b. It can be taken without any loss of generality that a minimal degree $\mathbb{R}$-embedding also removes rows from $\widehat{\mathcal{A}}$ and $\widehat{b}$ that correspond to trivial equations, i.e. $0 = 0$, $A_{i,j} = 0$ or $\widehat{b}_i = 0$.

4.2 Lagrange Multipliers

Definition 4.2. The vectorization of $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$ of degree at most d is defined as

$$\text{vec}(\mathcal{A}) = (\mathcal{A}_{1,1,0}, \ldots, \mathcal{A}_{1,1,d}, \ldots, \mathcal{A}_{n,n,0}, \ldots \mathcal{A}_{n,n,d})^T.$$

Let $\ell = (\text{vec}((\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{b}), \widehat{b}^T\widehat{b} - 1)^T \cdot \lambda$, where $\lambda = (\lambda_1, \ldots, \lambda_\nu)$ with ν the number of non-trivial constraints. Their Lagrangian is $L = \Psi + \ell$, and the Hessian of the Lagrangian is $\nabla^2 L$.

4.2.1 The Jacobian.

Definition 4.3. The matrix $\psi(\widehat{b})$ is an alternative form of $(\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{b} = 0$ that satisfies $\psi(\widehat{b})\text{vec}(\mathcal{A} + \Delta\mathcal{A}) = 0$. That is, $\psi(\widehat{b})$ satisfies

$$\psi(\widehat{b}) \cdot \text{vec}(\mathcal{A} + \Delta\mathcal{A}) = 0 \iff (\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{b} = 0.$$

Here we use the bilinearity of the system $(\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{b} = 0$ to write the same system using a matrix with entries from $\widehat{b}$ instead of $\text{vec}(\mathcal{A} + \Delta\mathcal{A})$.

The closed-form expression for the Jacobian is given by

$$J = \begin{pmatrix} \nabla_{\widehat{\Delta\mathcal{A}}} \left((\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{b} \right) \\ \nabla_{\widehat{b}} \left((\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}})\widehat{b} + \widehat{b}^T\widehat{b} \right) \end{pmatrix} = \begin{pmatrix} \psi(\widehat{b}) & \widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}} \\ 0 & 2\widehat{b}^T \end{pmatrix}.$$

If the Jacobian has full rank at a solution, then it necessarily has full rank in an open neighborhood around that solution. We will show that J and $\nabla^2\Psi$ have full rank so $\nabla^2 L$ has full rank as well.

Theorem 4.4. *Suppose $\widehat{b}$ is minimally degree $\mathbb{R}$-embedded in $\widehat{\mathcal{A}}$, then J has full (row) rank when $\nabla L = 0$.*

Proof. We show that J full row rank by contradiction. If this matrix was rank deficient, then one row is a linear combination of the others. This means that one of the equations in the constraints is trivial or the solution is not regular (see [7, Section 3.1]). As we are only concerned about regular solutions, this contradicts the minimal degree $\mathbb{R}$-embedding. □

4.3 Iterative Post-Refinement

Newton's method for equality constrained minimization problems can be interpreted as solving the non-linear system of equations $\nabla L = 0$. Let $x = (\text{vec}(\Delta\mathcal{A})^T, \widehat{b}^T)^T$. Then Newton's method is based on the iterative update scheme

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k + \Delta x^k \\ \lambda^k + \Delta\lambda^k \end{pmatrix} \text{ such that } \nabla^2 L \begin{pmatrix} \Delta x \\ \Delta\lambda \end{pmatrix} = -\nabla L. \tag{4.1}$$

Since $\nabla^2 L$ has full rank, the iteration is well defined by matrix inversion. This system can be solved in several different ways, although the most straightforward is matrix inversion. This method clearly has quadratic convergence as follows.

Fact 4.5 (Bertsekas [7, Prop. 4.4.3]). *Let $x^\star$ be a strict local minimum where the Jacobian of the constraints has full rank, with corresponding Lagrange multiplier $\lambda^\star$ such that*

$$\nabla L(x^\star, \lambda^\star) = 0 \text{ and } y^T \nabla^2_{xx} L y > 0$$
$$\text{for } y \neq 0 \text{ such that } J^T y = 0.$$

Then $(x^\star, \lambda^\star)$ is a point of attraction to the Newton iteration, and if $(x, \lambda) \to (x^\star, \lambda^\star)$ then the convergence rate is quadratic if the objective and constraints have a locally Lipschitz Hessian around $x^\star$.

Since our objective function and constraints are quadratic functions we trivially have the required Lipschitz continuity. It is noted that one can always make $\nabla^2_{xx} L$ positive definite by re-scaling the objective function without changing the optimal values of x.

Theorem 4.6. *The previously described Newton method converges quadratically with a suitable initial guess.*

Corollary 4.7. *Solutions to Problem 2.2 are isolated solutions in the minimal $\mathbb{R}$-embedding. That is, if x is an optimal solution, then there is a non-trivial open neighborhood around x where x is the only (locally) optimal solution.*

Although we only optimize over a single vector in the kernel, if $\widehat{\mathcal{A}} + \widehat{\Delta\mathcal{A}}$ is rank deficient and our initial guess is sufficiently close to the optimal solution, then it does not matter which element of the kernel we choose to use in the optimization. Note that this gives us the nearest rank deficient matrix, not the nearest matrix of prescribed rank. Performing an iteration on the basis of the entire kernel is left as future work.

4.4 Computing an initial guess

There are a variety of interesting first-order or other methods that can be used to obtain an initial guess. The most intuitive method to obtain an initial guess is to do a a lift and project using the SVD [11]. Another approach is to use a first-order method such as STLS or RSVD and then improve upon the answer with post-refinement when the convergence becomes too slow. Once a suitable initial guess is obtained, the next question is how to determine the Lagrange multipliers. Since J has full rank this reduces to solving a linear system of equations by substituting the initial guess into the gradient of L and solving for λ. There will always be a solution when first order conditions are satisfied since J has full row rank. Since initial guesses will generally not satisfy first order conditions, one can solve a linear least squares problem to approximate λ instead.

In general, there is a large selection of different methods available to compute an initial guess. A single iteration of lift and project is reasonably fast, and when the minimum residual is sufficiently

small the iteration will converge to the optimal solution, as the problem is well-posed. In instances of a large residual any method can be used and a locally-optimal solution can be computed by post-refinement.

5 DESCRIPTION OF IMPLEMENTATION

In this section we discuss implementation details and demonstrate our implementation for computing the nearest rank deficient matrix polynomial. All algorithms are implemented in Maple 2016. Examples are constructed by taking a singular matrix polynomial, then perturbing entries by a scaled amount of noise. Noise is constructed by generating a polynomial of prescribed degree structure with coefficients chosen uniformly at random from $(0, 1)$, then scaled appropriately. We do not perturb high-order or low-order coefficients that are set to zero and restrict ourselves to relatively large amounts of noise where it generally takes at least three steps to converge. The matrix polynomials are generated by rank-factorization with coefficients generated by `randpoly`. The left rank factor has coefficients of degree $\lfloor d/2 \rfloor$ and the right rank factor has coefficients of degree $\lceil d/2 \rceil$. All experiments are done using quad precision floating point arithmetic, with about 35 decimal digits of accuracy.

To compute the approximate kernel vector, first we use the SVD to compute an approximate kernel of an $\mathbb{R}$-embedded rank deficient matrix polynomial. Next we use structured orthogonal elimination RQ (LQ) decomposition to produce a minimally (degree) $\mathbb{R}$-embedded vector from the kernel.

5.1 Description of Algorithm

Algorithm 1 : `Iterative Post-Refinement`

Input:

- Full rank matrix polynomial $\mathcal{A} \in \mathbb{R}[t]^{n\times n}$
- Rank deficient matrix polynomial $\mathcal{C} \in \mathbb{R}[t]^{n]\times n}$
- Approximate kernel vector $c \in \mathbb{R}[t]^{n\times 1}$ of the desired degree structure
- Structure matrix $\Delta\mathcal{A}$ to optimize over

Output:

- Singular matrix $\mathcal{A} + \Delta\mathcal{A}$ with $\mathcal{b} \in \ker(\mathcal{A} + \Delta\mathcal{A})$ or an indication of failure.

1: $\mathbb{R}$-Embed $\mathcal{A}, \mathcal{C}, c$ and $\Delta\mathcal{A}$.
2: Compute Lagrangian L from Section 4.
3: Initialize λ via linear least squares from $\nabla L|_x = 0$.
4: Compute $\begin{pmatrix} x + \Delta x \\ \lambda + \Delta\lambda \end{pmatrix}$ by solving (4.1) until $\left\| \begin{pmatrix} \Delta x \\ \Delta\lambda \end{pmatrix} \right\|_2$ is sufficiently small or divergence is detected.
5: Return the locally optimal $\Delta\mathcal{A}$ and $\mathcal{b}$ or an indication of failure.

In our implementation we compute Δx and $\Delta\lambda$ using an iterative linear least squares method instead of inverting the Hessian of the Lagrangian directly. Improvements can be made in the conditioning of the values of x by exploiting the block structure of the Hessian since the Jacobian of the constraints can have a large condition number. The size of $\nabla^2 L$ is $O(n^4 d^2)$ and accordingly each iteration has a cost of $O(n^{12} d^6)$ flops using standard matrix multiplication.

5.2 Experiments

Small Example. Consider the singular (up to 2 decimal points of precision) matrix polynomial $\mathcal{A}$:

$$\begin{pmatrix} -.038t^2 - .21t + .026 & .12t^2 + .38t + .048 & .28t^2 - .14t + .11 \\ -.013t^2 + .18t - .17 & -.15t^2 - .16t + .31 & -.28t^2 + .48t - .19 \\ .051t^2 - .11t - .045 & .15t^2 + .20t + .038 & .22t^2 - .15t - .090 \end{pmatrix},$$

and approximate kernel vector $\mathcal{b}$,

$$\begin{pmatrix} .15t^2 - .40t + .77 & -.19t^2 + .10t + .23 & .10t^2 + .34t - .28 \end{pmatrix}^T.$$

Given the matrix polynomial $\mathcal{B}$

$$\begin{pmatrix} -.03760t^2 - .2122t + .0278 & .107t^2 + .363t + .0563 & .293t^2 - .1385t + .1141 \\ .003t^2 + .18027t - .1758 & -.14914t^2 - .1510t + .327 & -.2859t^2 + .469t - .173 \\ .0577t^2 - .1060t - .056 & .1455t^2 + .212t + .0321 & .231t^2 - .1514t - .075 \end{pmatrix},$$

the goal is to find a singular matrix polynomial $\mathcal{B} + \Delta\mathcal{B}$ where $\|\Delta\mathcal{B}\|_F^2$ is minimized. Using $\mathcal{A}$ and $\mathcal{b}$ as an initial guess for our algorithm, the relative (non-squared) error is $\frac{\|\mathcal{B}-\mathcal{A}\|_F}{\|\mathcal{B}\|_F} \approx .0499$. A table showing the convergence table is given in Figure 1.

Figure 1: Example of Convergence

iteration	$\|x_{i-1} - x_i\|_2$
1	1.3074e-1
2	2.0941e-2
3	3.8330e-4
4	1.4141e-7
5	2.4169e-14
6	4.7003e-28
7	0

We indeed see that each iteration is converging (quadratically) and the size of the perturbation that the algorithm converges to is $\|\Delta\mathcal{B}\|_F \approx 0.026604$. The solution is a unique local minimizer.

Performance on Randomly Generated Examples. In the experiments presented in Figure 2 all matrices are $n \times n$ of degree at most d in all entries. We restrict ourselves to modest values of n and d, given the expensive per-iteration cost. The computed matrices are rank deficient by at least 1 in all experiments.

Algorithm 1 typically fails because the initial guess is not sufficiently close, that is the input matrix is too far away or the approximate kernel vector produces a large residual. Failures to converge can also occur due to encountering a singular Jacobian (computed kernel vector is not minimally $\mathbb{R}$-embedded), as is the case in the examples with $n = 6, d = 6$ and $n = 9, d = 2$ with $\frac{\|A-A_{init}\|}{\|A\|} \approx 1e-2$. Singular Jacobians can be solved in practice by re-initializing the iteration with a new approximate kernel vector that is minimially $\mathbb{R}$-embedded. Our approximate kernel code relies on user specified parameters to determine numerical error terms and to compute the rank numerically of the $\mathbb{R}$-embedded matrix. If the rank computation fails then the kernel vector returned may not be minimally $\mathbb{R}$-embedded or $\|A_{init} b_{init}\|$ may be large.

6 CONCLUSIONS AND FUTURE WORK

We have shown that finding the nearest singular matrix polynomial can be set as a numerically well-posed problem, and is amenable to first-order optimization methods. It is demonstrated that minimal solutions exist and are well separated by non-trivial open neighborhoods. We also provide a theory for a second order method that

obtains quadratic convergence and discuss corresponding implementation details.

In the immediate future we will consider finding more specific rank approximations. While we currently only optimize over a single kernel vector, we would like to optimize over a basis of the kernel in order to achieve a "rank at most r" approximation as opposed to a rank deficient approximation. Optimizing over a minimally $\mathbb{R}$-embedded kernel should be a relatively straight forward generalization of the results presented here.

These results can also be generalized to obtain a quadratically convergent algorithm for a broader class of linearly structured STLS problems that exploit first order necessary conditions and second order sufficient conditions.

We also regard this current paper as a first step towards a formally robust approach to non-linear matrix polynomials, in the spirit of recent work with symbolic-numeric algorithms for polynomials. Problems such as approximate matrix polynomial division, GCRD and factorization all have applications which can benefit from these modern tools.

Figure 2: Convergence of Random Examples

n	d	iterations	$\frac{\|\mathcal{A}-\mathcal{A}_{init}\|_F}{\|\mathcal{A}\|_F}$	$\frac{\|\Delta\mathcal{A}\|_F}{\|\mathcal{A}\|_F}$	Status
3	2	4	2.273085e-06	6.448939e-07	
3	2	4	1.711447e-04	6.329353e-05	
3	2	6	1.536637e-02	4.607345e-03	
3	6	4	1.815279e-06	5.667709e-07	
3	6	2	1.736938e-04	1.910097e-04	FAIL
3	6	7	1.549175e-02	4.201870e-03	
3	10	3	1.805139e-06	5.115107e-07	
3	10	4	1.516551e-04	5.043466e-05	
3	10	6	1.528087e-02	5.461671e-03	
6	2	4	1.795881e-06	3.144274e-07	
6	2	5	1.678479e-04	3.179304e-05	
6	2	7	1.649747e-02	3.290890e-03	
6	6	4	1.627576e-06	4.013569e-07	
6	6	6	1.582963e-04	3.401362e-05	
6	6	1	1.514245e-02	6.858211e-03	FAIL
6	10	4	1.718920e-06	3.557049e-07	
6	10	6	1.851868e-04	4.166933e-05	
6	10	2	1.703748e-02	2.988045e-02	FAIL
9	2	4	1.739717e-06	3.594439e-07	
9	2	5	1.696604e-04	3.759553e-05	
9	2	1	1.668151e-02	6.114100e-03	FAIL
9	4	4	1.727764e-06	4.331134e-07	
9	4	6	1.664166e-04	2.987824e-05	
9	4	9	1.685704e-02	2.292579e-03	
12	2	4	1.748623e-06	3.280337e-07	
12	2	5	1.754237e-04	2.210751e-05	
12	2	9	1.673896e-02	2.282716e-03	

REFERENCES

[1] T. Abatzoglou, J. Mendel, and G. Harada. 1991. The constrained total least squares technique and its applications to harmonic superresolution. *IEEE Transactions on Signal Processing* 39, 5 (1991), 1070–1087.

[2] P-A Absil, R. Mahony, and R. Sepulchre. 2009. *Optimization algorithms on matrix manifolds.* Princeton University Press.

[3] S. Ahmad and R. Alam. 2015. On Wilkinson's problem for matrix pencils. *Electronic Journal of Linear Algebra* 30 (2015), 632–648.

[4] B. Beckermann and G. Labahn. 1998. A fast and numerically stable Euclidean-like algorithm for detecting relatively prime numerical polynomials. *Journal of Symbolic Computation* 26, 6 (1998), 691–714.

[5] B. Beckermann and G. Labahn. 1998. When are two numerical polynomials relatively prime? *J. Symb. Comp.* 26 (1998), 677–689.

[6] B. Beckermann, G. Labahn, and G. Villard. 2006. Normal forms for general polynomial matrices. *Journal of Symbolic Computation* 41, 6 (2006), 708–737.

[7] D. Bertsekas. 1999. *Nonlinear programming.* Athena Scientific.

[8] B. Botting, M. Giesbrecht, and J.P. May. 2005. Using the Riemannian SVD for Problems in Approximate Algebra. In *Proc. Workshop on Symbolic-Numeric Comp.* 209–219.

[9] R. Byers, C. He, and V. Mehrmann. 1998. Where is the nearest non-regular pencil? *Linear algebra and its applications* 285, 1 (1998), 81–105.

[10] R. Byers and N. Nichols. 1993. On the stability radius of a generalized state-space system. *Linear Algebra and Its Applications* 188 (1993), 113–134.

[11] J. Cadzow. 1988. Signal enhancement – a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 36, 1 (1988), 49–62.

[12] R. M. Corless, P. Gianni, B. Trager, and S. Watt. 1995. The singular value decomposition for polynomial systems. In *Proc. Int. Symp. Symb. Algebraic Computation (ISSAC).* 96–103.

[13] B. De Moor. 1993. Structured total least squares and L_2 approximation problems. *Linear algebra and its applications* 188 (1993), 163–205.

[14] B. De Moor. 1994. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing* 42, 11 (1994), 3104–3113.

[15] B. De Moor. 1995. The Riemannian singular value decomposition. In *Signal Processing, III: Algorithms, Architectures and Applications.* Elsevier, 61–78.

[16] G. Eckart and G. Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrica* 1 (1936), 211–218.

[17] M. Giesbrecht and J. Haraldson. 2014. Computing GCRDs of Approximate Differential Polynomials. In *Proc. Workshop on Symbolic-Numeric Computing (SNC'14).* 78–87.

[18] M. Giesbrecht, J. Haraldson, and E. Kaltofen. 2016. Computing Approximate Greatest Common Right Divisors of Differential Polynomials. (2016). Submitted for publication, 41 pages.

[19] I. Gohberg, P. Lancaster, and L. Rodman. 2009. *Matrix polynomials.* SIAM.

[20] G. Golub and V. Pereyra. 1973. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis* 10, 2 (1973), 413–432.

[21] G. Golub and V. Pereyra. 2003. Separable nonlinear least squares: the variable projection method and its applications. *Inverse problems* 19, 2 (2003), R1.

[22] G. Golub and C. Van Loan. 2012. *Matrix Computations.* Vol. 3. JHU Press.

[23] N. Guglielmi, C. Lubich, and V. Mehrmann. 2016. On the nearest singular matrix pencil. *Preprint* 12 (2016).

[24] E. Kaltofen and A. Storjohann. 2015. The complexity of computational problems in exact linear algebra. In *Encyclopedia of Applied and Computational Mathematics.* Springer, 227–233.

[25] E. Kaltofen, Z. Yang, and L. Zhi. 2005. Structured Low Rank Approximation of a Sylvester Matrix. In *Proc. Int. Workshop on Symbolic-Numeric Computation (SNC 2005).* 188–201.

[26] E. Kaltofen, Z. Yang, and L. Zhi. 2006. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In *Proc. 2006 Int. Symp. Symbolic Algebraic Comput.* ACM Press, 169–176.

[27] D. Kressner and M. Voigt. 2015. Distance Problems for Linear Dynamical Systems. In *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory.* Springer, 559–583.

[28] P. W. Lawrence and R. M. Corless. 2015. Backward Error of Polynomial Eigenvalue Problems Solved By Linearization of Lagrange Interpolants. *SIAM J. Matrix Analysis* 36, 4 (2015), 1425–1442.

[29] B. Li, Z. Yang, and L. Zhi. 2005. Fast Low Rank Approximation of a Sylvester Matrix by Structure Total Least Norm. *J. Japan Soc. Symbolic and Algebraic Computation* 11 (2005), 165–174.

[30] I. Markovsky. 2008. Structured low-rank approximation and its applications. *Automatica* 44, 4 (2008), 891–909.

[31] I. Markovsky. 2011. *Low rank approximation: algorithms, implementation, applications.* Springer Science & Business Media.

[32] B. Rosen, H. Park, and J. Glick. 1996. Total least norm formulation and solution for structured problems. *SIAM J. Matrix Anal. Appl.* 17, 1 (1996), 110–126.

[33] B. Rosen, H. Park, and J. Glick. 1998. Structured total least norm for nonlinear problems. *SIAM J. Matrix Anal. Appl.* 20, 1 (1998), 14–30.

[34] É. Schost and P.-J. Spaenlehauer. 2016. A quadratically convergent algorithm for structured low-rank approximation. *Foundations of Computational Mathematics* 16, 2 (2016), 457–492.

[35] A. Storjohann and G. Villard. 2005. Computing the rank and a small nullspace basis of a polynomial matrix. In *Proc. Int. Symp. Symb. Alg. Comp. (ISSAC).* ACM Press, 309–316.

[36] L. Zhi. 2007. Numerical Optimization in Hybrid Symbolic-numeric Computation. In *Proc. 2007 International Workshop on Symbolic-Numeric Computation.* 33–35.