Numerical Linear System Solving with Parametric Entries by Error Correction[*]

Brice Boyer
Department of Mathematics,
North Carolina State University
Raleigh, North Carolina 27695-8205, USA
bbboyer@ncsu.edu
www4.ncsu.edu/~bbboyer

Erich L. Kaltofen
Department of Mathematics,
North Carolina State University
Raleigh, North Carolina 27695-8205, USA
kaltofen@math.ncsu.edu
www4.ncsu.edu/~kaltofen

ABSTRACT

We consider the problem of solving a full rank consistent linear system $A(u)\mathbf{x} = b(u)$ where the $m \times n$ matrix A and the m-dimensional vector b has entries that are polynomials in u over a field. We give an algorithm that computes the unique solution $\mathbf{x} = \mathbf{f}(u)/g(u)$, which is a vector of rational functions, by evaluating the parameter u at distinct points. Those points ξ_λ where the matrix A evaluates to a matrix $A(\xi_\lambda)$, with entries over the scalar field, of lower rank, or in the numeric setting to an ill-conditioned matrix, are not identified but accounted for by error-correcting code techniques. We also correct true errors where the evaluation at some $u = \xi_\lambda$ results in an erroneous, possibly full rank consistent and well-conditioned scalar linear system. Our algorithm generalizes Welch/Berlekamp decoding of Reed/Solomon error correcting codes and their numeric floating point counterparts.

We have implemented our algorithms with floating point arithmetic. For the determination of the exact numerator and denominator degrees and number of errors we use singular values based numeric rank computations. The arising linear systems for the error-corrected parametric solution are demonstrated to be well-conditioned even when the input scalars have noise. In several initial experiments we have shown that our approach is numerically stable even for larger systems $m = n = 100$, provided the degrees in the solution are small (≤ 2). For smaller systems $m = n = 10$ with higher degrees (≤ 20) the algorithm works similarly to rational function recovery. Our implementation can correct 13 true errors in both settings.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms; E.4 [**Coding and Information Theory**]: Error control codes; G.1.1 [**Numerical Analysis**]: Interpolation—*Smoothing*; G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra

General Terms

Algorithms

Keywords

Error correcting codes, Numerical outlier, Linear algebra with parameters

1. INTRODUCTION

Linear systems $A(u)\mathbf{x} = b(u)$ with coefficients that contain a parameter u can be solved exactly by evaluating the parameter at distinct values ξ_ℓ, solving the evaluated system, and by interpolating from the evaluated solutions the parametric solution [8]. If the coefficients are polynomials over a field, one may also Hensel-lift the parameter and construct truncated power series approximations for the solution [9]. We suppose here that the system has full rank and that a unique parametric solution exists. If the scalar coefficients in the polynomial coefficients $a_{i,j}(u)$, $b_i(u)$ are floating point numbers, evaluations at $u = \xi_\ell$ may drop the numeric rank below full rank, but that condition is dependent on a threshold of the condition number of the evaluated matrix $A(\xi_\ell)$. By interpreting the solution of the evaluated numerically low rank system $A(\xi_\ell)\mathbf{x}^{[\ell]} = b(\xi_\ell)$ as an error in the data of the reconstruction problem, we can deploy techniques from algebraic error correcting codes [3,12] and their numeric counterparts [5,6,11] for recovering the solution from those error-free $\mathbf{x}^{[\ell]}$ where the numeric rank of $A(\xi_\ell)$ remained full. The Welch/Berlekamp decoder for Reed/Solomon codes can compute a reduced rational function solution of the algebraic interpolation-with-errors problem without locating the error. Our algorithm for parametric linear system solving does the same: it computes a low degree solution without locating the almost singular evaluation matrices $A(\xi_\ell)$ (see Theorem 2.1 below). Thus no condition number threshold comes into play.

The introduction of an error locator polynomial allows for more: we can suppose that at some $u = \xi_\ell$ we have obtained erroneous (possibly well-conditioned full rank) linear systems $\hat{A}^{[\ell]}\mathbf{x}^{[\ell]} = \hat{b}^{[\ell]}$. Again by using algebraic error correction decoding, we can compute the actual solution $\mathbf{x} = \mathbf{f}(u)/g(u)$ to $A(u)\mathbf{x} = b(u)$, where $\mathbf{f}$ is a vector of polynomials $\mathbf{f}^{[j]}$ and g the least common denominator.

[*] This research was supported in part by the National Science Foundation under Grant CCF-1115772 (Kaltofen).

SNC'14, July 28-31, 2014, Shanghai, China.
Copyright 2014 ACM 978-1-4503-2963-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2631948.2631956

We require $L = d_{\mathbf{f}} + d_g + E_1 + 2E_2 + 1$ evaluations with $d_{\mathbf{f}} \geq \max_j\{\deg(\mathbf{f}^{[j]})\}$, $d_g \geq \deg(g)$, and where $\leq E_1$ evaluations can have lower exact or numeric rank in $\hat{A}^{[\ell]} = A(\xi_\ell)$ and $\hat{b}^{[\ell]} = b(\xi_\ell)$, while $\leq E_2$ evaluations have $\hat{A}^{[\ell]} \neq A(\xi_\ell)$ or $\hat{b}^{[\ell]} \neq b(\xi)$ (see Theorem 2.2). In the numeric setting, "=" means here approximately equal and "≠" means significantly different. For a 1×1 identity matrix $A(u) = I_1$ our algorithm specializes to Welch/Berlekamp decoding.

We have implemented our algorithm and report initial experimental results. We have focused on scalar arithmetic with double precision floating point numbers, and scalar coefficients of the polynomial entries of A and b with some noise and severe outliers. In the numeric setting, a heavily noisy scalar is indistinguishable from an outlier with small relative error, and the notion of correcting an error vs. denoising imprecise data is subject to thresholds: one may state that corrections of relative error $\leq \epsilon$ is denoising and that of relative error $\geq \Theta$ is error correction. The nature of the error-location and correction in our algorithms is that if $\Theta \gg \epsilon$, outliers are reliably identified. But when the ration Θ/ϵ is small, say $\leq 10^4$, error detection is less reliable. See [1, Section 3] for a complete numerical analysis of a Blahut's Reed/Solomon decoder for ≤ 1 outlier. With more errors, the analysis is missing even for the Blahut decoder, and our algorithm here is a generalization of that decoder.

Here we have only tested square systems $m = n$. The parameter u in the rational vector solution $\mathbf{f}(u)/g(u)$ introduces an added complication when dealing with noise in the scalars, that of approximate GCDs: high degree polynomials with random coefficients have complex roots clustered around the unit circle in the Gaussian plane [7, Section 2] and our algorithm identifies an evaluation point that is an approximate root of the solution polynomials as an error location. Note that for $m = n = 1$ the algorithm performs numeric rational function recovery with outlier correction, and with oversampling such mistaken error locations can be tolerated [5]. We have set up our experiments where the degrees in $\mathbf{f}(u)/g(u)$ are small (≤ 20), in which case no such spurious approximate factors appear. We note that our algorithms construct well-conditioned overdetermined linear systems with floating point numbers as entries. We use the singular value decomposition for solving the overdetermined systems, which essentially constitutes a total least squares solution of our noisy systems. We could also deploy, in the case of high parameter degrees and/or overdetermined systems $m \gg n$, structured total least norm (STLN) algorithms (cf. [7]).

Here we have presented the case where there is a single parameter u and where the linear system has a unique parametric solution. Our approach can be generalized to several parameters $u_1, \ldots, u_n$ and to rank deficient systems with multiple solutions or to systems with no solutions. For inconsistent systems we compute a Farkas-like certificate vector for inconsistency as our solution (see [2]). Errors in the evaluations could, of course, turn an inconsistent system into a consistent one, but not when one over-evaluates at $2E$ points and allows for $\leq E$ errors. The arising linear systems in our algorithms are structured, especially for low dimensions and high parameter degrees, and solvers for structured systems can by used for their solution (cf. [11]).

2. EXACT VECTOR-OF-FUNCTIONS SOLVING

We solve a system of linear equations

$$A(u)\,\mathbf{x} = b(u), \quad A(u) \in \mathsf{K}[u]^{m\times n}, b(u) \in \mathsf{K}[u]^m,$$

where K is a field. We shall assume that the system has a unique solution

$$\mathbf{x} = \begin{bmatrix} \vdots \\ \mathbf{f}^{[j]}(u)/g(u) \\ \vdots \end{bmatrix} \in \mathsf{K}(u)^n, \quad g \neq 0, \tag{1}$$

where g is the least common (monic) denominator, whose leading coefficient in u is 1. If $\mathbf{x}$ is a vector of zeros, we set $g = 1$. Our assumption is equivalent to

$$m \geq n \text{ and } \operatorname{rank}(A(u)) = \operatorname{rank}([A(u), b(u)]) = n. \tag{2}$$

We compute $\mathbf{x}$ as follows: we select $L = d_{\mathbf{f}} + d_g + E + 1$ (distinct) elements $\xi_\ell \in \mathsf{K}$, where $0 \leq \ell \leq L-1$ and $\xi_{\ell_1} \neq \xi_{\ell_2}$ for $\ell_1 \neq \ell_2$, where $d_{\mathbf{f}} \geq \max_{1\leq j\leq n}\{\deg(\mathbf{f}^{[j]})\}$ and $d_g \geq \deg(g)$; finally, if $\lambda_1, \ldots, \lambda_k$ is the list of indices for the evaluations with $\operatorname{rank}(A(\xi_{\lambda_\kappa})) < \operatorname{rank}(A(u))$ and with $\operatorname{rank}(A(\xi_\ell)) = \operatorname{rank}(A(u)) = n$ for all $\ell \notin \{\lambda_1, \ldots, \lambda_k\}$ then we presume that $k \leq E$. Here $d_{\mathbf{f}}, d_g$ and E are given or estimated on input. We compute (1) by solving the homogeneous linear system in the unknown coefficients of $\mathbf{\Phi}^{[j]}(u)$, $\Psi(u)$,

$$A(\xi_\ell)\begin{bmatrix} \vdots \\ \mathbf{\Phi}^{[j]}(\xi_\ell) \\ \vdots \end{bmatrix} = \Psi(\xi_\ell) b(\xi_\ell), \tag{3}$$

where $\deg(\mathbf{\Phi}^{[j]}) \leq d_{\mathbf{f}}, \deg(\Psi) \leq d_g, 0 \leq \ell \leq L-1$.

The linear system (3) has $n(d_{\mathbf{f}}+1) + d_g + 1$ unknown coefficients for the $\mathbf{\Phi}^{[j]}$ and Ψ and mL equations. There is at least one solution with $\Psi \neq 0$, namely $\mathbf{\Phi}^{[j]} = \mathbf{f}^{[j]}$ and $\Psi = g$.

Theorem 2.1 *We suppose that for $\geq d_{\mathbf{f}} + d_g + 1$ of the ξ_ℓ we have $\operatorname{rank}(A(\xi_\ell)) = \operatorname{rank}(A(u)) = n$. Let $\Psi_{\min}$ be the denominator component of a solution of (3) with $\Psi_{\min} \neq 0$ and scaled to have leading coefficient 1 in u, and of minimal degree of all such solutions, and let $\mathbf{\Phi}^{[j]}_{\min}$ be the corresponding numerator components of that solution. Then for all j we have $\mathbf{\Phi}^{[j]}_{\min} = \mathbf{f}^{[j]}$ and $\Psi_{\min} = g$.*

PROOF. From (1) we have $A(\xi_\ell)\mathbf{f}(\xi_\ell) = g(\xi_\ell)b(\xi_\ell)$, where $\mathbf{f}(u)$ is the vector of the $\mathbf{f}^{[j]}(u)$. Our solutions satisfy the relation $A(\xi_\ell)\mathbf{\Phi}(\xi_\ell) = \Psi(\xi_\ell)b(\xi_\ell)$, where $\mathbf{\Phi}(u)$ is the vector of the $\mathbf{\Phi}^{[j]}(u)$. Therefore $A(\xi_\ell)(\Psi(\xi_\ell)\mathbf{f}(\xi_\ell) - g(\xi_\ell)\mathbf{\Phi}(\xi_\ell)) = 0$. For $\ell \notin \{\lambda_1, \ldots, \lambda_k\}$ the matrix $A(\xi_\ell)$ has linearly independent columns, so $\Psi(\xi_\ell)\mathbf{f}(\xi_\ell) - g(\xi_\ell)\mathbf{\Phi}(\xi_\ell) = 0$ for such ℓ. The j-th entry, $(\Psi\mathbf{f}^{[j]} - g\mathbf{\Phi}^{[j]})(\xi_\ell) = 0$, where $(\Psi\mathbf{f}^{[j]} - g\mathbf{\Phi}^{[j]})(u)$ is a polynomial of degree $\leq d_{\mathbf{f}} + d_g$ that thus vanishes at $\geq d_{\mathbf{f}} + d_g + 1$ distinct evaluation points. Therefore for all solutions $(\mathbf{\Phi}^{[j]}, \Psi)$ of (3) we have $\Psi\mathbf{f}^{[j]} - g\mathbf{\Phi}^{[j]} = 0$. From $\Psi_{\min}\mathbf{f}^{[j]} = g\mathbf{\Phi}^{[j]}_{\min}$ we obtain $\mathbf{f}^{[j]}/g = \mathbf{\Phi}^{[j]}_{\min}/\Psi_{\min}$. Since g is the least common denominator, which is unique, $g = \Psi_{\min}$. □

Remark 1. The Theorem 2.1 shows that by minimizing the denominator degree in the solutions $(\boldsymbol{\Phi}^{[j]}, \Psi)$ of (3) one avoids to have to identify the "unlucky" evaluations ξ_{λ_κ}. In the exact setting, one may Cauchy interpolate the vector of rational functions $\mathbf{f}/g$ from the "lucky" evaluations by the algorithm in [10], and a rank drop can be diagnosed during solving the linear system $A(\xi_\ell)\mathbf{x} = b(\xi_\ell)$. In the numeric setting, however, a numeric rank drop is subject to a measure of ill-conditionedness. By using an additional E evaluations for k ill-conditioned $A(\xi_\lambda)$ we globally correct for those values. Furthermore, the linear system (3) is based on the initial coefficient data in A and b, and not on the derived data of the solutions at ξ_ℓ. Therefore an overdetermined parametric system $A(u)\mathbf{x} = b(u)$ that is only approximately consistent may be better solved. Other techniques such Hensel lifting may not perform as well (accumulation of numerical errors for not working with initial data, difficulty to detect ill-conditionedness in successive evaluations). □

The linear system (3) uses $A(u)$ and $b(u)$ evaluated at ξ_ℓ. Therefore, one can interpret A and b as given by a procedure ("black box") that produces the (dense) matrix $A(\xi_\ell)$ and the (dense) vector $b(\xi_\ell)$ on input ξ_ℓ. Note that what we [4] call a black box matrix is different: in the black box matrix setting one has a procedure for computing the product of a (sparse or structured) matrix times a vector. For Theorem 2.1 we must have the correct entries of $A(\xi_\ell)$ and $b(\xi_\ell)$ in the linear system (3). Nevertheless, it is possible to compute the solution vector $\mathbf{f}/g$ of (1) when some entries of $A(\xi_\ell)$ or $b(\xi_\ell)$ are erroneous. Let $\hat{A}^{[\ell]} \in \mathsf{K}^{m\times n}$ and $\hat{b}^{[\ell]} \in \mathsf{K}^m$ be the matrix and vector returned on evaluation at ξ_ℓ. One evaluates at $\hat{L} = d_\mathbf{f} + d_g + \hat{E}_1 + 2\hat{E}_2 + 1$ distinct $\xi_\ell \in \mathsf{K}$, where $0 \le \ell \le \hat{L} - 1$, and presumes that at exactly $k_1 \le \hat{E}_1$ evaluations ξ_{λ_κ}, where $1 \le \kappa \le k_1$ and $0 \le \lambda_\kappa \le \hat{L} - 1$, one has $\operatorname{rank}(\hat{A}^{[\lambda_\kappa]}) < n$ and $\hat{A}^{[\lambda_\kappa]}\mathbf{f}(\xi_{\lambda_\kappa}) = g(\xi_{\lambda_\kappa})\hat{b}^{[\lambda_\kappa]}$, and at exactly $k_2 \le \hat{E}_2$ evaluations ξ_{λ_μ}, where $k_1+1 \le \mu \le k_1+k_2$, $0 \le \lambda_\mu \le \hat{L} - 1$, one has

$$\hat{A}^{[\lambda_\mu]}\mathbf{f}(\xi_{\lambda_\mu}) \neq g(\xi_{\lambda_\mu})\hat{b}^{[\lambda_\mu]}. \tag{4}$$

The condition (4) implies that $\hat{A}^{[\ell]} \neq A(\xi_{\lambda_\mu})$ or/and $\hat{b}^{[\ell]} \neq b(\xi_{\lambda_\mu})$, but the reverse may not be satisfied: the solution to an erroneous system can agree with the original $\mathbf{f}(\xi_{\lambda_\mu})/g(\xi_{\lambda_\mu})$. We then solve the linear system in the unknown coefficients of $\hat{\boldsymbol{\Phi}}^{[j]}(u)$ and $\hat{\Psi}(u)$,

$$\hat{A}^{[\ell]} \begin{bmatrix} \vdots \\ \hat{\boldsymbol{\Phi}}^{[j]}(\xi_\ell) \\ \vdots \end{bmatrix} = \hat{\Psi}(\xi_\ell)\hat{b}^{[\ell]}, \tag{5}$$

where $\deg(\hat{\boldsymbol{\Phi}}^{[j]}) \le d_\mathbf{f} + \hat{E}_2, \deg(\hat{\Psi}) \le d_g + \hat{E}_2, 0 \le \ell \le \hat{L} - 1$.

Theorem 2.2 *We suppose that for $\le \hat{E}_2$ of the ξ_ℓ we have $\hat{A}^{[\ell]}\mathbf{f}(\xi_\ell) \neq g(\xi_\ell)\hat{b}^{[\ell]}$ and for $\ge d_\mathbf{f} + d_g + \hat{E}_2 + 1$ of the ξ_ℓ we have $\operatorname{rank}(\hat{A}^{[\ell]}) = n$ and $\hat{A}^{[\ell]}\mathbf{f}(\xi_\ell) = g(\xi_\ell)\hat{b}^{[\ell]}$. Let $\hat{\Psi}_{\min}$ be the denominator component of a solution of (5) with $\hat{\Psi}_{\min} \neq 0$ and scaled to have leading coefficient 1 in u, and of minimal degree of all such solutions, and let $\hat{\boldsymbol{\Phi}}^{[j]}_{\min}$ be the corresponding numerator components of that solution. Furthermore, let $\Lambda(u) = \prod_{\mu \text{ subj. to } (4)}(u - \xi_{\lambda_\mu})$ be an error locator polynomial. Then for all j we have $\boldsymbol{\Phi}^{[j]}_{\min} = \Lambda\mathbf{f}^{[j]}$ and $\hat{\Psi}_{\min} = \Lambda g$.*

Proof. For $\ell \notin \{\lambda_1, \ldots, \lambda_{k_1+k_2}\}$ we must have by (4) that $\hat{A}^{[\ell]}\mathbf{f}(\xi_\ell) = g(\xi_\ell)\hat{b}^{[\ell]}$ and by (5) $\hat{A}^{[\ell]}\hat{\boldsymbol{\Phi}}(\xi_\ell) = \hat{\Psi}(\xi_\ell)\hat{b}^{[\ell]}$, where $\hat{\boldsymbol{\Phi}}(u)$ is the vector of the $\hat{\boldsymbol{\Phi}}^{[j]}(u)$. Thus we have $\hat{A}^{[\ell]}(\hat{\Psi}(\xi_\ell)\mathbf{f}(\xi_\ell) - g(\xi_\ell)\hat{\boldsymbol{\Phi}}(\xi_\ell)) = 0$. Since $\hat{A}^{[\ell]}$ for those ℓ has linearly independent columns, the vector of field elements $\hat{\Psi}(\xi_\ell)\mathbf{f}(\xi_\ell) - g(\xi_\ell)\hat{\boldsymbol{\Phi}}(\xi_\ell) = 0$. The j-th entry, $(\hat{\Psi}\mathbf{f}^{[j]} - g\hat{\boldsymbol{\Phi}}^{[j]})(\xi_\ell) = 0$, where $(\hat{\Psi}\mathbf{f}^{[j]} - g\hat{\boldsymbol{\Phi}}^{[j]})(u)$ is a polynomial of degree $\le d_\mathbf{f} + d_g + \hat{E}_2$ that thus vanishes at $\ge d_\mathbf{f} + d_g + \hat{E}_2 + 1$ distinct evaluation points. Therefore for all solutions $(\hat{\boldsymbol{\Phi}}^{[j]}, \hat{\Psi})$ of (5) we have $\hat{\Psi}\mathbf{f}^{[j]} - g\hat{\boldsymbol{\Phi}}^{[j]} = 0$. From $\hat{\Psi}_{\min}\mathbf{f}^{[j]} = g\hat{\boldsymbol{\Phi}}^{[j]}_{\min}$ we obtain $\mathbf{f}^{[j]}/g = \hat{\boldsymbol{\Phi}}^{[j]}_{\min}/\hat{\Psi}_{\min}$, and hence there is a polynomial $h(u)$ with $h\mathbf{f}^{[j]} = \hat{\boldsymbol{\Phi}}^{[j]}_{\min}$ for all j and $hg = \hat{\Psi}_{\min}$. For each μ that satisfies (4) we have a row i_μ, where $1 \le i_\mu \le m$, with $\hat{A}^{[\lambda_\mu]}_{i_\mu}\mathbf{f}(\xi_{\lambda_\mu}) \neq g(\xi_{\lambda_\mu})\hat{b}^{[\lambda_\mu]}_{i_\mu}$ and $\hat{A}^{[\lambda_\mu]}_{i_\mu}(hf)(\xi_{\lambda_\mu}) = \hat{A}^{[\lambda_\mu]}_{i_\mu}\hat{\boldsymbol{\Phi}}_{\min}(\xi_{\lambda_\mu}) = \hat{\Psi}_{\min}(\xi_{\lambda_\mu})\hat{b}^{[\lambda_\mu]}_{i_\mu} = (hg)(\xi_{\lambda_\mu})\hat{b}^{[\lambda_\mu]}_{i_\mu}$ which implies that $h(\xi_{\lambda_\mu}) = 0$. The proof concludes by observing that $(\Lambda\mathbf{f}^{[j]}, \Lambda g)$ solves (5). □

Remark 2. If $m = n = 1$ and $A = I_1 = [1]$ (which implies $\hat{E}_1 = 0$) we recover the polynomial $b(u)$ from $d_\mathbf{f} + d_g + 2\hat{E}_2 + 1$ values, where $\le \hat{E}$ of the values can be erroneous. The algorithm then is Welch/Berlekamp decoding of an algebraic Reed/Solomon error correcting code [3, 6, 11, 12]. If $m = n$ and $A = I_n$ we recover the rational function vector $\mathbf{f}(u)/g(u) = b(u) \in \mathsf{K}(u)^n$ from $d_\mathbf{f} + d_g + \hat{E}_1 + 2\hat{E}_2 + 1$ values $b(\xi_\ell) \in \mathsf{K}$, where $\le \hat{E}_1$ evaluations are roots of g, indicated by $b(\xi_\ell) = \infty^n$ leading to the equation $\hat{\Psi}(\xi_\ell) = 0$ in (5), and $\le \hat{E}_2$ evaluations are erroneous in one or more components of $\hat{b}^{[\ell]} \neq b(\xi)$ (cf. [3]). □

3. ALGORITHM AND EXPERIMENTS

We now describe in Section 3.1 the numerical algorithm derived from Theorems 2.1 and 2.2. We make essential use of the singular value decomposition (SVD) to get the numerical rank of our linear systems and to solve them as the SVD also tolerates some noise (total least squares solutions). Then we show implementation results in Section 3.2. We craft some parametric systems with low or high degree solutions, set errors and singularities at the evaluation points, and show the behavior of our implementation.

3.1 Numerical Algorithm

The following Algorithm 3.1 computes a rational numerical solution $\mathbf{x} = \hat{\mathbf{f}}/\hat{g}$ (see (1)) to a parametric system $A(u)\mathbf{x} = b(u)$ with base field $\mathbb{C}$, where the numerical evaluations of $A(u)$ can drop its rank at most $\hat{E}_1$ times and can be erroneous at most $\hat{E}_2$ times. Bounds on the degree $d_\mathbf{f} \ge \max_j \deg_u(\mathbf{f}^{[j]})$ and $d_g \ge \deg_u(g)$ are also given as inputs.

Algorithm 3.1 (Numerical Vector-of-Functions Solver With Errors)

Input: Two "black boxes" $A \in \mathbb{C}[u]^{m\times n}$ and $b \in \mathbb{C}[u]^{n\times 1}$;
Bounds $\hat{E}_1, \hat{E}_2$, and $d_\mathbf{f}, d_g$.

Output: Polynomials $\mathbf{f} \in \mathbb{C}[u]^{n\times 1}$ and $g \in \mathbb{C}[u]$ s.t. $\mathbf{x} = \mathbf{f}/g$ is a solution to the system $A(u)\,\mathbf{x}(u) = b(u)$.

1. Initialization.

    a) Create a set Ξ of $L = d_\mathbf{f} + d_g + 1 + \hat{E}_1 + 2\hat{E}_2$ distinct random evaluation points.

b) Set up the vectors $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ corresponding to the unknowns in $\mathbf{f}$ and g, of resp. length $n(d_{\mathbf{f}} + \hat{E}_2 + 1)$ and $d_g + \hat{E}_2 + 1$ (corresponding to reps. degrees $d_{\mathbf{f}} + \hat{E}_2$ and $d_g + \hat{E}_2$ of $\mathbf{f}$ and g).

c) Set up a linear system W from evaluations of A and b at Ξ, such that $W\ [\vec{\mathbf{x}}\ \vec{\mathbf{y}}]^\top = 0$.

2. SVD step.

a) Compute a SVD of W and find its numeric rank ρ.

b) Construct a reduced linear system $\widetilde{W}$ by removing the $\rho - 1$ unknowns of highest degree to $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$.

c) From the column vector in $\widetilde{W}$ corresponding to the leading coefficient in $\hat{g}$ (of degree $d_g + \hat{E}_2 - \rho + 1$ now), reconstruct $\hat{\mathbf{\Phi}}$ and $\hat{\Psi}$ as possible candidates for $\hat{\mathbf{f}}$ and $\hat{g}$. Either one moves that column to the right and solves a non-singular system (since the co-rank of $\widetilde{W}$ is 1 and the polynomial $\hat{g}$ is monic), or one performs a second SVD and normalizes the leading coefficient of $\hat{\Psi}$ to 1.

3. Error removal.

a) From the evaluations of $\hat{\mathbf{\Phi}}$ and $\hat{\Psi}$ at Ξ, construct Λ, the error locating polynomial. For $\xi \in \Xi$, a factor $u - \xi$ is added to Λ when the evaluation $\|\hat{\mathbf{\Phi}}(\xi)\|_1 + |\hat{\Psi}(\xi)|$ is smaller than a predefined ϵ_{root} or there is a large jump ($> \iota_{\text{root}}$) in the sequence of evaluations.

b) Via a least squares fit, compute the approximate divisions $\hat{\mathbf{f}} = \hat{\mathbf{\Phi}}/\Lambda$ and $\hat{g} = \hat{\Psi}/\Lambda$.

c) Check the residue of the least squares of Step 3-b or return failure. □

We first observe that this algorithm works whenever the field for A and b is $\mathbb{R}$ or $\mathbb{C}$. On $\mathbb{C}$, we can for instance choose roots of unity for the set of evaluations Ξ. Therefore this is an algorithm that approximates the solution on the unit circle. Second, the tolerances for detecting small non-zero singular values, for a jump in the singular values (Step 2-a) or for detecting an error (Step 3-a) can be given as an additional input, but we prefer to detect the largest jump between consecutive small values (< 1) and use that as a threshold for considering those values below to be 0.

In Step 2-a, the numeric rank ρ is ≥ 1 because the solution is homogenous of degree ≥ 1 (there is at least one degree of liberty before normalisation of the leading term of g to 1.) However, we can have $\rho > 1$, in which case (Step 2-b) we minimize the degree of $\mathbf{f}$ and g by removing the $\rho - 1$ coefficients of highest degree. The reduced system W has then numeric rank 1. We note that the the ordering of the coefficients in $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ (Step 1-b and following) has no importance.

In Step 3-a, we evaluate the quantity $N(\xi) := \|\hat{\mathbf{\Phi}}(\xi)\|_1 + |\hat{\Psi}(\xi)| = \sum_{i=1}^{n} |\mathbf{\Phi}^{[i]}(\xi)| + |\Psi(\xi)|$. Indeed, we expect that a random element ξ that makes $|\Psi(\xi)| \approx 0$ and all $\mathbf{\Phi}^{[i]}(\xi) \approx 0$ is a common root to Ψ and to all $\mathbf{\Phi}^{[i]}$, $i \in \{1, \dots, n\}$ (this can be related to the numerical Zippel lemma in [7, Lemma 3.1]). It is possible that such a common root is an approximate root for Ψ and $\mathbf{\Phi}$, in which case the error locating polynomial Λ also contains a common factor to Ψ and $\mathbf{\Phi}$; this is related to numeric GCDs and the result of the algorithm is acceptable as long as the residue in Step 3-c is small.

Note that if two evaluation points in Ξ are chosen near one another and one point yields an erroneous system ($\hat{E}_2$), then both points are identified as error locations but it is impossible to identify the erroneous system. Such a situation is particular to the numeric setting. The near-multiplicity in Λ then causes an unwanted drop in the degree of $\mathbf{f}$ and g.

Finally, we remark that this algorithm does not correct the matrix entries.

3.2 Implementation and performance

Our algorithm has been implemented in Maple 17 and we report initial experiments in the following Tables 1 and 2. These tables show the behavior over $\mathbb{C}$ and $\mathbb{R}$ of our algorithm on polynomial matrices of relatively low degree (< 5), with solutions of low degree or maximal degree. For low degree, the input matrices $A(u)$ and right-side vectors $b(u)$ have been constructed in a special manner, also so that the set Ξ contains roots of $\det(A)$. In particular, $A(u) = g(u)A_{[\text{rand}]}(u)D(u)$ and $b(u) = A_{[\text{rand}]}(u)D(u)\mathbf{f}(u)$, where $D(u)$ is a diagonal matrix with diagonal entries $u - \xi$ where k_1 of the ξ belong to Ξ, which is a set of random roots of unity $\exp(2r\pi i/1001)$ (Table 1) or random fractions of bounded numerator (absolutely by 20 000) and denominator 10 000 (Table 2), and the coefficient of the entries of $\mathbf{f}$, $A_{[\text{rand}]}(u)$ and the polynomial g are random integers between -7 and 7; the system is then scaled so that the infinity norms of the polynomial entries of b are ≤ 7. For maximal degree, $A(u)$ and $b(u)$ are random integer linear matrix and vector polynomials that are later scaled so that the exact solution has bounded infinity norm. Singularities up to $\hat{E}_1$ are introduced using elements in Ξ and errors up to $\hat{E}_2$ are created by changing a term in A, namely by setting $(\hat{A}^{[\ell]})_{1,1} = 10\,000$ (see (5)). The timings were performed on a `x86_64` Gentoo Linux laptop running Maple 17 with a 2.3GHz Intel® Core™ i7 and 6Gb of RAM.

In Table 1 and Table 2 we report our experiments. There we denote:

- $m = n$, the dimensions of the square system $A(u)\mathbf{x} = b(u)$;
- *Rel. noise* the amount of relative noise added to an evaluation $A(\xi_\ell)$, the number 0 corresponding to the default precision (hardware floats);
- $\deg(A) = \max_{i,j}\{\deg_u(a_{i,j})\}$, $\deg(b) = \max_i\{\deg_u(b_i)\}$;
- height(A) we denote the maximum absolute value among all coefficients of all $a_{i,j}(u)$;
- *Rel. Error* the relative backward error, computed as $(\|\hat{\mathbf{f}} - \mathbf{f}\|_2^2 + \|\hat{g} - g\|_2^2)^{1/2}/(\|\mathbf{f}\|_2^2 + \|g\|_2^2)^{1/2}$;
- $\kappa_{\text{eval}}^{-1}$ is the smallest reciprocal of all the absolute condition numbers of all $A(\xi)$, $\xi \in \Xi$ (= smallest singular value among all evaluated systems);
- $\kappa_{\text{solve}}^{-1}$ is the reciprocal of the absolute condition number (= smallest non-zero singular value) of the non-singular system in Step 2-c.

For example, line 7 in Table 1 reads as follows. For a 50×50 matrix A of degree 4 with scalars absolutely bounded by 19, and a vector b of degree 4 and scalars bounded (by default, see construction above) by 7, we solve a system with bounds 3 and 3 on the degrees of $\mathbf{f}$ and g and bounds 15 and

Ex.	n	*Rel. noise*	$\deg A$, $\deg b$, height(A)	$d_{\mathbf{f}}(\deg \mathbf{f})$; $d_g(\deg g)$	$\hat{E}_1(k_1)$; $\hat{E}_2(k_2)$	*Time (s)*	*Rel. Error*	$\kappa_{\text{eval}}^{-1}$	$\kappa_{\text{solve}}^{-1}$
1	10	0	5, 5, 16	2 (1); 2 (1)	0 (0); 1 (1)	< 1	4.8e-7	3.8e-2	3.4e-1
2	10	0	5, 5, 16	2 (1); 2 (1)	1 (1); 1 (1)	< 1	6.8e-9	6.4e-10	4.5e-1
3	10	1e-6	5, 5, 16	2 (1); 2 (1)	1 (1); 1 (1)	< 1	5.9e-6	7.3e-10	4.5e-1
4	20	0	5, 5, 18	3 (3); 3 (3)	5 (3); 5 (3)	< 5	1.7e-6	1.4e-10	1.2e-1
5	20	1e-6	5, 5, 18	3 (3); 3 (3)	5 (3); 5 (3)	< 5	1.3e-3	1.6e-10	1.2e-1
6	50	0	4, 4, 19	3 (2); 3 (2)	6 (5); 6 (5)	< 35	1.2e-6	4.8e-11	2.6e-2
7	50	1e-6	4, 4, 19	3 (2); 3 (2)	6 (5); 6 (5)	< 80	1.1e-3	4.8e-11	2.6e-2
8	100	0	4, 4, 16	3 (2); 3 (2)	15 (13); 15 (13)	< 1000	6.9e-6	1.8e-11	7.2e-3
9	100	1e-6	4, 4, 16	3 (2); 3 (2)	15 (13); 15 (13)	< 1200	6.5e-3	1.8e-11	7.2e-3
10	10	0	1, 1, 158	10 (10); 10 (10)	3 (0); 15 (13)	< 7	6.6e-5	3.1e-1	7.1e-3
11	10	0	1, 1, 158	10 (10); 10 (10)	4 (0); 15 (13)	< 7	1.3e-5	3.1e-1	2.4e-2
12	10	1e-6	1, 1, 158	10 (10); 10 (10)	4 (0); 15 (13)	< 7	5.8e-2	3.1e-1	2.4e-2
13	10	0	1, 1, 158	10 (10); 10 (10)	4 (2); 15 (13)	< 15	8.8e-5	7.1e-10	2.3e-2
13	10	1e-6	1, 1, 158	10 (10); 10 (10)	4 (2); 15 (13)	< 15	9.0e-3	7.1e-10	2.3e-2
14	10	0	2, 2, 4001	20 (20); 20 (20)	4 (2); 15 (13)	< 35	2.3	7.1e-10	2.3e-2
15	10	1e-6	2, 2, 4001	20 (20); 20 (20)	4 (2); 15 (13)	< 35	FAIL	7.2e-8	8.6e-4

Table 1: Algorithm performance on $\mathbb{C}$

Ex.	n	*Rel. noise*	$\deg A$, $\deg b$, height(A)	$d_{\mathbf{f}}(\deg \mathbf{f})$; $d_g(\deg g)$	$\hat{E}_1(k_1)$; $\hat{E}_2(k_2)$	*Time (s)*	*Rel. Error*	$\kappa_{\text{eval}}^{-1}$	$\kappa_{\text{solve}}^{-1}$
1	10	0	5, 5, 20	2 (1); 2 (1)	0 (0); 1 (1)	< 1	9.3e-7	3.2e-3	7.9e-3
2	10	0	5, 5, 20	2 (1); 2 (1)	1 (1); 1 (1)	< 1	7.9e-7	6.5e-10	1.3e-2
3	10	1e-6	5, 5, 20	2 (1); 2 (1)	1 (1); 1 (1)	< 1	1.4e-5	9.6e-10	4.9e-1
4	20	0	5, 5, 16	3 (3); 3 (3)	5 (3); 5 (3)	< 2	1.6e-6	1.5e-12	6.4e-2
5	20	1e-6	5, 5, 16	3 (3); 3 (3)	5 (3); 5 (3)	< 5	3.7e-4	1.5e-12	6.4e-2
6	50	0	4, 4, 19	3 (2); 3 (2)	6 (5); 6 (5)	< 15	8.8e-9	5.8e-12	1.6e-2
7	50	1e-6	4, 4, 19	3 (2); 3 (2)	6 (5); 6 (5)	< 40	8.9e-6	5.8e-12	1.6e-2

Table 2: Algorithm performance on $\mathbb{R}$

15 on the number of singularities and errors in the evaluations, resp. Furthermore, the evaluations are done with a relative noise of magnitude 10^{-6}. The actual degrees of $\mathbf{f}$ and g are 2 and 2 and the actual number of singularities and errors are 13 and 13, resp. The time spent on solving this parametric system was just below 80 seconds. During the construction of the system in Step 1-c (a system of size 1250×510), the evaluated matrices at points in Ξ had a smallest non-zero singular value as small as 4.8×10^{-11}, while in solving the final reduced system in Step 2-c (a system of size 1250×408), the smallest singular value was 2.1×10^{-2}. Also, on line 14 and 15 of Table 1, we report a relative error > 1 or failure because the relative error is too large ($\approx 8.8 \times 10^{1}$), although the location of the errors (Λ) was correct.

4. REFERENCES

[1] Boyer, B., Comer, M. T., and Kaltofen, E. L. Sparse polynomial interpolation by variable shift in the presence of noise and outliers in the evaluations. In *Proc. Tenth Asian Symposium on Computer Mathematics (ASCM 2012)* (Oct. 2013). Submitted to SLNCS; URL: http://www.math.ncsu.edu/~kaltofen/bibliography/13/BCK13.pdf.

[2] Giesbrecht, M., Lobo, A., and Saunders, B. D. Certifying inconsistency of sparse linear systems. In *ISSAC 98 Proc. 1998 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 1998), O. Gloor, Ed., ACM Press, pp. 113–119.

[3] Kaltofen, E., and Pernet, C. Cauchy interpolation with errors in the values. Submitted manuscript, 13 pages, Dec. 2013.

[4] Kaltofen, E., and Trager, B. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput. 9*, 3 (1990), 301–320. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/90/KaTr90.pdf.

[5] Kaltofen, E., and Yang, Z. Sparse multivariate function recovery from values with noise and outlier errors. In *ISSAC 2013 Proc. 38th Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2013), M. Kauers, Ed., Association for Computing Machinery, pp. 219–226. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/13/KaYa13.pdf.

[6] Kaltofen, E., and Yang, Z. Sparse multivariate function recovery with a high error rate in evaluations. In *ISSAC 2014 Proc. 39th Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2014), K. Nabeshima, Ed., Association for Computing Machinery. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/priv/KaYa14.pdf.

[7] Kaltofen, E., Yang, Z., and Zhi, L. On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms. In *SNC'07 Proc. 2007 Internat. Workshop on Symbolic-Numeric Comput.* (New York, N. Y., 2007), J. Verschelde and S. M. Watt, Eds., ACM Press, pp. 11–17. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/07/KYZ07.pdf.

[8] McClellan, M. T. The exact solution of systems of linear equations with polynomial coefficients. *J. ACM 20* (1973), 563–588.

[9] Moenck, R. T., and Carter, J. H. Approximate algorithms to derive exact solutions to systems of linear equations. In *Symbolic and Algebraic Computation* (Heidelberg, Germany, 1979), vol. 72 of *Lect. Notes Comput. Sci.*, Springer Verlag, pp. 65–73. Proc. EUROSAM '79.

[10] Olesh, Z., and Storjohann, A. The vector rational function reconstruction problems. In *Proc. Waterloo Workshop on Computer Algebra: devoted to the 60th birthday of Sergei Abramov (WWCA)* (2007), pp. 137–149.

[11] Olshevsky, V., and Shokrollahi, M. A. A displacement approach to decoding algebraic codes. In *Algorithms for Structured Matrices: Theory and Applications*. American Mathematical Society, Providence, Rhode Island, USA, 2003, pp. 265–292. Contemporary Math., vol. 323. URL: http://www.math.uconn.edu/~olshevsky/papers/shokrollahi_f.pdf.

[12] Welch, L. R., and Berlekamp, E. R. Error correction of algebraic block codes. US Patent 4,633,470, 1986. Filed 1983; see http://patft.uspto.gov/.

Notation (in alphabetic order):	
$A(u)$	the matrix of coefficients with parameter u
$\hat{A}^{[\ell]}$	either $A(\xi_\ell)$ or an erroneous matrix
$b(u)$	the right-side vector of constants with parameter u
$\hat{b}^{[\ell]}$	either $b(\xi_\ell)$ or an erroneous right-side vector
$d_{\mathbf{f}}^{[j]}, d_{\mathbf{f}}$	degree bounds $\geq \deg_u(\mathbf{f}^{[j]}), \geq \max_j \deg_u(\mathbf{f}^{[j]})$
d_g	a degree bound $\geq \deg_u(g)$

Notation continued (in alphabetic order):	
ϵ	a numerical tolerance for algorithmic decisions; not an outlier error, as is sometimes used
E, $\hat{E}_1$	$\geq k, k_1$, an upper bound on the number of evaluations where the rank drops
$\hat{E}_2$	$\geq k_2$, an upper bound on the number of evaluations with errors in the entries
$\mathbf{f}^{[j]}$	the j-th entry in the solution, a polynomial (in u), $1 \leq j \leq n$
$\mathbf{\Phi}^{[j]}$	the j-th entry in a computed solution, a polynomial (in u), $1 \leq j \leq n$
g	the (least common) denominator of the solution, a polynomial (in u)
k, k_1	is the actual number of evaluations that drop the rank
k_2	the actual number of evaluations with errors in the entries
K	an arbitrary field with exact arithmetic
L	the length of the list of a batch of evaluations
λ_κ	$1 \leq \kappa \leq k$, the positions of the evaluations that drop the rank: $\operatorname{rank} A(\xi_{\lambda_\kappa}) < \operatorname{rank} A(u)$; or where the evaluations constructs an erroneous system
m	the number of rows in $A(u)$
n	is the number of columns in $A(u)$ and rows in $b(u)$
Ψ	the common denominator of a computed solution
u	the parameter in the coefficients
$\mathbf{x}$	a solution vector $A(u)\mathbf{x}(u) = b(u)$
ξ_ℓ	disting values for the parameter u from a field $\in \mathsf{K}$