

## VECTOR FITTING FOR MATRIX-VALUED RATIONAL APPROXIMATION\*

Z. DRMAČ<sup>†</sup>, S. GUGERCIN<sup>‡</sup>, AND C. BEATTIE<sup>‡</sup>

**Abstract.** Vector Fitting (VF) is a popular method of constructing rational approximants that provides a least squares fit to frequency response measurements. In an earlier work, we provided an analysis of VF for scalar-valued rational functions and established a connection with optimal  $\mathcal{H}_2$  approximation. We build on this work and extend the previous framework to include the construction of effective rational approximations to *matrix-valued* functions, a problem which presents significant challenges that do not appear in the scalar case. Transfer functions associated with multi-input/multioutput (MIMO) dynamical systems typify the class of functions that we consider here. Others have also considered extensions of VF to matrix-valued functions and related numerical implementations are readily available. However, to the best of our knowledge, a detailed analysis of numerical issues that arise does not yet exist. We offer such an analysis including critical implementation details here. One important issue that arises for VF on matrix-valued functions that has remained largely unaddressed is the control of the McMillan degree of the resulting rational approximant; the McMillan degree can grow very high in the case of large input/output dimensions. We introduce two new mechanisms for controlling the McMillan degree of the final approximant, one based on alternating least-squares minimization and one based on ancillary system-theoretic reduction methods. Motivated in part by our earlier work on the scalar VF problem as well as by recent innovations for computing optimal  $\mathcal{H}_2$  approximation, we establish a connection with optimal  $\mathcal{H}_2$  approximation, and are able to improve significantly the fidelity of VF through numerical quadrature, with virtually no increase in cost or complexity. We provide several numerical examples to support the theoretical discussion and proposed algorithms.

**Key words.** least squares, frequency response, model order reduction, MIMO vector fitting, transfer function

**AMS subject classifications.** 34C20, 41A05, 49K15, 49M05, 93A15, 93C05, 93C15

**DOI.** 10.1137/15M1010774

**1. Introduction.** Rational functions provide significant advantages over other classes of approximating functions, such as polynomials or trigonometric functions, that are important in the approximation of functions that occur in engineering and scientific applications. *Matrix-valued* rational functions offer substantial additional flexibility and broaden the domain of applicability by providing the potential for the interpolation and approximation of parameterized families of multidimensional observations. For example, in a variety of engineering applications, the dynamics arising from multi-input/multioutput (MIMO) dynamical systems may be inaccessible to direct modeling, yet input-output relationships often may be observed as a function of frequency, yielding an enormous amount of data. In such cases, one may wish to deduce an *empirical dynamical system model*, nominally represented as a matrix-valued

---

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 2, 2015; accepted for publication (in revised form) July 2, 2015; published electronically September 15, 2015.

<http://www.siam.org/journals/sisc/37-5/M101077.html>

<sup>†</sup>Faculty of Science, Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia (drmac@math.hr). The work of the first author was supported by the grant HRZZ-9345 from the Croatian Science Foundation.

<sup>‡</sup>Department of Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0123 (gugercin@math.vt.edu, beattie@vt.edu). The work of these authors was supported in part by NSF through grant DMS-1217156.

rational function, that fits the measured frequency response data. This derived model may then be used as a surrogate in order to predict system behavior or to determine suitable control strategies. See [2, 8, 55] for examples of rational approximation in action.

The *McMillan degree* of a matrix-valued rational function,  $\mathbf{H}(s)$ , is the sum of pole multiplicities over the (extended) complex plane, or, equivalently, the dimension of the state space in a minimal realization of  $\mathbf{H}(s)$ . It is convenient to think of  $\mathbf{H}(s)$  as a transfer function matrix associated with a stable MIMO linear time-invariant system having  $m$  inputs and  $p$  outputs (although this interpretation is not necessary for what follows); McMillan degree is a useful proxy for the level of complexity associated with  $\mathbf{H}(s)$ . Let  $r > 0$  be an integer denoting the desired McMillan degree for our approximant:  $\mathbf{H}_r(s) = \mathbf{N}(s)/d(s)$ , where  $\mathbf{N}(s)$  is a  $p \times m$  matrix having elements that are polynomials in  $s$  of order  $r - 1$  or less, and  $d(s)$  is a (scalar) polynomial function having exact order  $r$ . Denote by  $\mathcal{R}_r$  the set of matrix-valued functions of this form. Note that  $\mathcal{R}_r$  consists of matrix-valued functions having entries that are strictly proper rational functions of order  $r$ ; the McMillan degree of  $\mathbf{H}_r \in \mathcal{R}_r$  could range as high as  $r \cdot \min(m, p)$ , which could be significantly larger than our target,  $r$ . We assume that observations (evaluations) of  $\mathbf{H}(s)$  are available at predetermined points in the complex plane,  $s = \xi_1, \dots, \xi_\ell$ . Only observed values at these points:  $\mathbf{H}(\xi_j)$ , for  $j = 1, \dots, \ell$ , will be necessary in order to derive our approximations. We proceed by seeking a solution to

$$(1.1) \quad \min_{\mathbf{H}_r \in \mathcal{R}_r} \sum_{i=1}^{\ell} \rho_i \|\mathbf{H}_r(\xi_i) - \mathbf{H}(\xi_i)\|_F^2.$$

Sanathanan and Koerner [54] proposed an approach to solving (1.1) that produces a sequence of rational matrix approximants  $\mathbf{H}_r^{(k)}(s)$  having the form  $\mathbf{H}_r^{(k)}(s) = \mathbf{N}^{(k)}(s)/d^{(k)}(s)$ , where  $\mathbf{N}^{(k)}(s)$  is a  $p \times m$  matrix of polynomials of degree  $r - 1$  or less and  $d^{(k)}(s)$  is a (scalar-valued) polynomial of degree  $r$ . For each  $k$ , the coefficients of  $\mathbf{N}^{(k)}$  and  $d^{(k)}$  are adjusted by solving a weighted linear least squares problem with a weight determined by  $d^{(k-1)}$ . An important reformulation of the Sanathanan–Koerner (SK) iteration was introduced by Gustavsen and Semlyen [38], which became known under the name *Vector Fitting* (VF). The term “vector fitting” is appropriate in light of the interpretation of the Frobenius norm,  $\|\cdot\|_F$ , that appears in (1.1) as a standard Euclidean vector norm,  $\|\cdot\|_2$ , of an “unraveled” matrix listed columnwise as a vector in  $\mathbb{C}^{mp}$ . The Gustavsen–Semlyen VF method produces a similar sequence of rational approximants  $\mathbf{H}_r^{(k)}(s) = \mathbf{N}^{(k)}(s)/d^{(k)}(s)$  but now with  $\mathbf{N}^{(k)}$  and  $d^{(k)}$  defined as rational functions represented in barycentric form. Making central use of a clever change of representation at each step, the Gustavsen–Semlyen VF method achieves greater numerical stability and efficiency than the original SK iteration.

We describe both the SK and VF iterations in section 2 and make some observations that contribute to our analysis of it in section 3. In particular, we note that the change of representation implicit in the VF iteration can be related to a change of representation from barycentric to pole–residue form. This change of representation is made explicit in section 2.2, where we provide formulas that appear to be new. These formulas can be useful at any step of either the SK or VF iteration in order to estimate the contribution that each pole makes to the current approximation. This, in turn, is useful in determining whether,  $r$ , the initial estimate of McMillan degree, is unnecessarily large relative to the information contained in the observed

data. We note that many authors have applied, modified, and analyzed VF; see, e.g., [36, 40, 21, 20, 23, 22, 24]. A MATLAB implementation `vecfit3` is provided at [56] and is widely used. When applied to MIMO problems, high fidelity rational approximations are sought and generally achieved at the expense of a relatively high McMillan degree for the final approximant. By way of contrast, the approaches we develop here are capable of providing systematic estimates to the McMillan degree of the original function,  $\mathbf{H}(s)$ , and can produce high fidelity rational approximations of any desired McMillan degree, to the extent possible.

In section 3, we analyze VF within a numerical linear algebra framework and discuss several important issues that are essential for a numerically sound implementation of the method. Although VF is based on successive solution of least squares (LS) problems, which is a well understood procedure, subtleties enter in the VF context. We review commonly used numerical LS solution procedures in section 3.1 and point out some details that become significant in the VF setting. In section 3.2, we argue and illustrate by way of example that, as iterations proceed, the VF coefficient matrices in the pole identification phase tend to become noisy with a significant drop in column norms that can also coincide with a reduction in numerical rank. This prompts us to advise caution when rescaling columns in order to improve the condition number, since rescaling columns that have been computed through massive cancellations will preclude inferring an accurate numerical rank.

Ill-conditioning is intrinsic to rational approximation and manifested through the potentially high condition numbers of Cauchy matrices that naturally arise. We introduce in section 3.4.2 another approach toward curbing ill-conditioning. Using recent results on high accuracy matrix computations, we show that successful computation is possibly independent of the condition number of the underlying Cauchy matrix, and we open possibilities for application of Tichonov regularization and the Morozov discrepancy principle, which provide additional useful tools when working with noisy data. In section 3.5, we offer some suggestions for efficient software implementation and provide some algorithmic details that reduce computational cost. For the sake of brevity, we omit discussion of the algorithmic details that allow computation to proceed using only real arithmetic.

Typically, the weights in (1.1) are  $\rho_i = 1$  and the sample points,  $\xi_i$ , are either uniformly or logarithmically spaced according to sampling expedience. We do consider this case in detail, but also propose, in section 4, an alternate strategy for choosing  $\rho_i$  and  $\xi_i$  that is guided by numerical quadrature. This connects the “vector fitting” process with optimal  $\mathcal{H}_2$  approximation and follows up on our recent work, [27]. Indeed, with proper choice of the nodes  $\xi_i$  and the weights  $\rho_i$ , we can interpret the weighted LS approximation of (1.1) as rational approximation in a discretized  $\mathcal{H}_2$  norm in the Hardy space of matrix functions that are analytic in the open right half-plane. This leads to a significant improvement in the performance of the VF approximation and also sets the stage for a new postprocessing stage proposed in section 5, where we address the question of computing an approximation having predetermined McMillan degree.

Rational data fitting has a long history (going back at least to Kalman [44]) and continues to be studied in a variety of settings: For example, Gonnet, Pachón, and Trefethen [32] recently provided a robust approach to rational approximation through linearized LS problems on the unit disk. Hokanson in [42] presented a detailed study

of exponential fitting, which is closely related to rational approximation. Berljafa and Güttel [10] recently proposed rational Krylov approximations for solving more general rational least-squares problems. The Loewner framework developed by Mayo and Antoulas [48] and Lefteriu and Antoulas [46] is an effective and numerically efficient method to construct rational interpolants directly from measurements. This approach also has been extended successfully to parametric [6, 43] and weakly nonlinear [3, 4] problems. In this paper, we focus solely on matrix-valued rational least-squares approximation by VF.

**2. Background and problem setting.** The VF iteration is built upon the Sanathanan–Koerner (SK) procedure [54], which we describe briefly as follows: A sequence of approximations having the form  $\mathbf{H}_r^{(k)} = \mathbf{N}^{(k)}(s)/d^{(k)}(s) \in \mathcal{R}_r$  is developed by taking  $d^{(0)}(s) \equiv 1$ , and then solving successively for  $\mathbf{N}^{(k+1)}$  and  $d^{(k+1)}$  via the weighted linear least squares problem:

$$(2.1) \quad \epsilon^{(k)} = \sum_{i=1}^{\ell} \frac{\rho_i}{|d^{(k)}(\xi_i)|^2} \left\| \mathbf{N}^{(k+1)}(\xi_i) - d^{(k+1)}(\xi_i) \mathbf{H}(\xi_i) \right\|_F^2 \longrightarrow \min \quad \text{for } k = 0, 1, 2, \dots$$

Since  $\mathbf{N}^{(k)}$  and  $d^{(k)}$  have a presumed affine dependence on parameters, (2.1) does indeed define a linear least squares problem with respect to those parameters. Notice also that once  $\mathbf{N}^{(k+1)}(s)$  and  $d^{(k+1)}(s)$  have been computed, only  $d^{(k+1)}(s)$  is used in the next iteration as a new weighting function. Although convergence of this process remains an open question, if the iterations do converge at least in the sense that, at some  $k_*$ ,  $d^{(k_*+1)}(s)$  “is close to”  $d^{(k_*)}(s)$  at the sample points  $\xi_i$ , the process is halted and we take  $\mathbf{H}_r(s) = \mathbf{N}^{(k_*+1)}(s)/d^{(k_*+1)}(s)$ . Note also that before this last step (i.e., for  $k < k_*$ ), any effort to compute the value of  $\mathbf{N}^{(k+1)}$  is wasted.

The error in (2.1) may be rewritten as  $\epsilon^{(k_*)} = \sum_{i=1}^{\ell} \rho_i |d^{(k_*+1)}(\xi_i)/d^{(k_*)}(\xi_i)|^2 \|\mathbf{H}_r(\xi_i) - \mathbf{H}(\xi_i)\|_F^2$ , which corresponds to (1.1) up to an error that depends on the deviation of  $|d^{(k_*+1)}/d^{(k_*)}|$  from 1. This deviation becomes small as convergence occurs and can be associated with stopping criteria that we introduce in section 3.4.3. If the iteration is halted prematurely, then the last step leaves  $d^{(k)}$  unchanged and redefines  $\mathbf{N}^{(k+1)}$  as the solution to the LS problem  $\sum_{i=1}^{\ell} \rho_i \|\mathbf{N}^{(k+1)}(\xi_i)/d^{(k)}(\xi_i) - \mathbf{H}(\xi_i)\|_F^2 \rightarrow \min$ . The implementation of this procedure depends on specific choices for the parameterization of  $\mathbf{N}^{(k)}(s)$  and  $d^{(k)}(s)$  used in representing  $\mathbf{H}_r^{(k)}(s)$ ; barycentric representations will offer clear advantage.

**2.1. VF = SK + barycentric representation.** Suppose that at the  $k$ th iteration step we choose a set of  $r$  mutually distinct (but otherwise arbitrary) nodes,  $\{\lambda_j^{(k)}\}_{j=1}^r$ . Consider a barycentric representation for  $\mathbf{H}_r^{(k)}(s)$ :

$$(2.2) \quad \mathbf{H}_r^{(k)}(s) = \frac{\mathbf{N}^{(k)}(s)}{d^{(k)}(s)} \equiv \frac{\sum_{j=1}^r \Phi_j^{(k)}/(s - \lambda_j^{(k)})}{1 + \sum_{j=1}^r \varphi_j^{(k)}/(s - \lambda_j^{(k)})}, \quad \Phi_j^{(k)} \in \mathbb{C}^{p \times m}, \quad \varphi_j^{(k)}, \lambda_j^{(k)} \in \mathbb{C}.$$

Observe that if  $\varphi_j^{(k)} \neq 0$ ,  $\mathbf{H}_r^{(k)}(\lambda_j^{(k)}) = \Phi_j^{(k)}/\varphi_j^{(k)}$ . On the other hand, if  $\varphi_j^{(k)} = 0$ , then  $\mathbf{H}_r^{(k)}(s)$  has a simple pole at  $\lambda_j^{(k)}$  with an associated residue given explicitly in (2.9) of Proposition 2.1.

A new approximant is sought having the form

$$\mathbf{H}_r^{(k+1)}(s) = \frac{\sum_{j=1}^r \widehat{\Phi}_j^{(k+1)} / (s - \lambda_j^{(k)})}{1 + \sum_{j=1}^r \widehat{\varphi}_j^{(k+1)} / (s - \lambda_j^{(k)})}, \quad \widehat{\Phi}_j^{(k+1)} \in \mathbb{C}^{p \times m}, \quad \widehat{\varphi}_j^{(k+1)} \in \mathbb{C}.$$

The weighted least squares error as described in (2.1) is written explicitly as

$$(2.3) \quad \epsilon^{(k)} = \sum_{i=1}^{\ell} \frac{\rho_i}{|d^{(k)}(\xi_i)|^2} \left\| \sum_{j=1}^r \frac{\widehat{\Phi}_j^{(k+1)}}{\xi_i - \lambda_j^{(k)}} - \mathbf{H}(\xi_i) \left( 1 + \sum_{j=1}^r \frac{\widehat{\varphi}_j^{(k+1)}}{\xi_i - \lambda_j^{(k)}} \right) \right\|_F^2.$$

One of the distinguishing features of the Gustavsen–Semlyen VF method [38] emerges at this point: the explicit weighting factors  $1/|d^{(k)}(\xi_i)|^2$  are eliminated through “pole relocation”, i.e., the nodes in the barycentric representation are changed in such a way so as to absorb the weighting factors. Note that if  $\{\lambda_j^{(k+1)}\}$  are the zeros of  $d^{(k)}(s)$ , then we can write  $d^{(k)}(s) = \prod_{q=1}^r (s - \lambda_q^{(k+1)}) / \prod_{q=1}^r (s - \lambda_q^{(k)})$ , and then

$$(2.4) \quad \begin{aligned} & \sum_{i=1}^{\ell} \rho_i \left\| \frac{\prod_{q=1}^r (\xi_i - \lambda_q^{(k)})}{\prod_{q=1}^r (\xi_i - \lambda_q^{(k+1)})} \right\|^2 \left\| \sum_{j=1}^r \frac{\prod_{q \neq j} (\xi_i - \lambda_q^{(k)}) \widehat{\Phi}_j^{(k+1)}}{\prod_{q=1}^r (\xi_i - \lambda_q^{(k)})} \right. \\ & \quad \left. - \mathbf{H}(\xi_i) \frac{\prod_{q=1}^r (\xi_i - \lambda_q^{(k)}) + \sum_{j=1}^r \widehat{\varphi}_j^{(k+1)} \prod_{q \neq j} (\xi_i - \lambda_q^{(k)})}{\prod_{q=1}^r (\xi_i - \lambda_q^{(k)})} \right\|_F^2 \\ & = \sum_{i=1}^{\ell} \rho_i \left\| \sum_{j=1}^r \frac{\prod_{q \neq j} (\xi_i - \lambda_q^{(k)}) \widehat{\Phi}_j^{(k+1)}}{\prod_{q=1}^r (\xi_i - \lambda_q^{(k+1)})} \right. \\ & \quad \left. - \mathbf{H}(\xi_i) \frac{\prod_{q=1}^r (\xi_i - \lambda_q^{(k)}) + \sum_{j=1}^r \widehat{\varphi}_j^{(k+1)} \prod_{q \neq j} (\xi_i - \lambda_q^{(k)})}{\prod_{q=1}^r (\xi_i - \lambda_q^{(k+1)})} \right\|_F^2 \equiv \epsilon^{(k)}. \end{aligned}$$

The next iterate,  $\mathbf{H}_r^{(k+1)}$ , will be represented in barycentric form using nodes  $\lambda_q^{(k+1)}$ ,  $q = 1, \dots, r$ . We assume simple zeros for simplicity. We introduce new variables,  $\varphi_j^{(k+1)}$ , so that

$$\frac{\prod_{q=1}^r (s - \lambda_q^{(k)}) + \sum_{j=1}^r \widehat{\varphi}_j^{(k+1)} \prod_{q \neq j} (s - \lambda_q^{(k)})}{\prod_{q=1}^r (s - \lambda_q^{(k+1)})} = 1 + \sum_{j=1}^r \frac{\varphi_j^{(k+1)}}{s - \lambda_j^{(k+1)}}.$$

In the same way, we introduce new unknowns,  $\Phi_j^{(k+1)}$ , so that

$$(2.5) \quad \sum_{j=1}^r \frac{\prod_{q \neq j} (s - \lambda_q^{(k)}) \widehat{\Phi}_j^{(k+1)}}{\prod_{q=1}^r (s - \lambda_q^{(k+1)})} = \sum_{j=1}^r \frac{\Phi_j^{(k+1)}}{s - \lambda_j^{(k+1)}}.$$

After this change of variables, the LS objective function from (2.3) appears as

$$(2.6) \quad \epsilon^{(k)} = \sum_{i=1}^{\ell} \rho_i \left\| \sum_{j=1}^r \frac{\Phi_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}} - \mathbf{H}(\xi_i) \left( 1 + \sum_{j=1}^r \frac{\varphi_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}} \right) \right\|_F^2.$$

Now, the values for  $\{\Phi_j^{(k+1)}\}_{j=1}^r$  and  $\{\varphi_j^{(k+1)}\}_{j=1}^r$  that minimize  $\epsilon^{(k)}$  in (2.6) will determine the next iterate  $\mathbf{H}_r^{(k+1)}(s)$ . The iteration continues by defining  $d^{(k+1)}(s) = 1 + \sum_{j=1}^r \varphi_j^{(k+1)} / (s - \lambda_j^{(k+1)})$  and new poles  $\lambda_j^{(k+2)}$  will be the computed zeros of  $d^{(k+1)}(s)$ .

Numerical convergence is declared and the iteration terminates at an index  $k_*$  when  $\max_j |\varphi_j^{(k_*)}|$  is “small enough.” The zeros of  $d^{(k_*)}(s)$  are extracted as the eigenvalues of the matrix  $\text{diag}(\lambda_j^{(k_*)})_{j=1}^r + (1, \dots, 1)^T (\varphi_1^{(k_*)}, \dots, \varphi_r^{(k_*)})$  and assigned to  $\{\lambda_j^{(k_*+1)}\}_{j=1}^r$ . The denominator  $d^{(k_*)}(s)$  is set to the constant 1 indicating numerical pole-zero cancellation, and the LS problem (2.6) is solved for  $\Phi_j^{(k_*)}$  (assigning  $k+1 = k_*$ ) with all  $\varphi_j^{(k_*)}$  terms replaced with zeros. The resulting VF approximant is

$$(2.7) \quad \mathbf{H}_r(s) = \sum_{j=1}^r \frac{\Phi_j^{(k_*)}}{s - \lambda_j^{(k_*+1)}}.$$

It is usually reported that numerical convergence takes place within a few iterations, provided that initial poles are well chosen. However, in difficult cases, numerical convergence may not be achieved, and the iterate described in (2.2) may have a denominator that is far from constant. Even if the stopping criterion is not satisfied, the above procedure that yields (2.7) will be valid at any  $k$ , and may be interpreted as taking the last computed barycentric nodes, viewing them as poles, and then computing a best least-squares rational approximation in pole-residue form. We discuss this closing step of the iteration further at the end of section 2.3. The stopping criterion will be discussed in section 3.4.3.

**2.2. Pole-residue form of barycentric approximant.** The barycentric representation offers many advantages for both the VF iteration and the SK iteration (i.e., with the explicit weighting by  $1/|d^{(k)}(\xi_i)|^2$  as in (2.3)). Nonetheless, the pole-residue representation is more convenient for analysis and further usage. For example, a pole-residue representation of a (rational) transfer function leads immediately to a state space realization of the underlying dynamical system. Furthermore, we may wish to inspect, in the course of iterations, an approximant (2.2) with the goal of estimating the importance of the contribution of certain of its poles and the possible

effect of discarding them. Hence, it would be useful in general to have an efficient procedure to transform a barycentric representation of a function into a pole-residue representation. Toward that end, consider a  $p \times m$  matrix rational function

$$(2.8) \quad \mathbf{G}(s) = \frac{\sum_{j=1}^r \frac{\Phi_j}{s-\lambda_j}}{1 + \sum_{j=1}^r \frac{\varphi_j}{s-\lambda_j}}, \quad \Phi_j \in \mathbb{C}^{p \times m}, \quad \varphi_j, \lambda_j \in \mathbb{C}, \quad \text{and } |\varphi_j| + \|\Phi_j\|_F > 0.$$

Assume that the barycentric nodes,  $\lambda_1, \dots, \lambda_r$  are distinct and closed under conjugation (nonreal values appear in complex conjugate pairs), and that the corresponding  $\varphi_j$ s and  $\Phi_j$ s have a compatible conjugation symmetry ( $\varphi_j, \Phi_j$  real if  $\lambda_j$  real,  $\varphi_i = \overline{\varphi_j}$ ,  $\Phi_i = \overline{\Phi_j}$  if  $\lambda_i = \overline{\lambda_j}$ ). Define complementary index sets:

$$\mathcal{J}_0 = \{j : \varphi_j = 0\} \quad \text{and} \quad \mathcal{J}_1 = \{j : \varphi_j \neq 0\}.$$

Then  $\{\lambda_j : j \in \mathcal{J}_k\}$ ,  $\{\varphi_j : j \in \mathcal{J}_k\}$  are closed under complex conjugation for  $k = 0, 1$ . Note that  $\{\lambda_j : j \in \mathcal{J}_0\}$  are among the poles of  $\mathbf{G}(s)$ . The remaining poles can be indexed as  $\{\widehat{\lambda}_j : j \in \mathcal{J}_1\}$ .

PROPOSITION 2.1. *Assume, in addition to the above, that all poles of  $\mathbf{G}(s)$  in (2.8) are simple.<sup>1</sup> Then,*

$$\mathbf{G}(s) = \sum_{j \in \mathcal{J}_1} \frac{\widehat{R}_j}{s - \widehat{\lambda}_j} + \sum_{j \in \mathcal{J}_0} \frac{R_j}{s - \lambda_j},$$

where the residues can be efficiently calculated as

$$(2.9) \quad R_j = \frac{\Phi_j}{1 + \sum_{i \in \mathcal{J}_1} \varphi_i / (\lambda_j - \lambda_i)}, \quad j \in \mathcal{J}_0; \quad \widehat{R}_j = \frac{\prod_{i \in \mathcal{J}_1} (\widehat{\lambda}_j - \lambda_i)}{\prod_{i \neq j} (\widehat{\lambda}_j - \widehat{\lambda}_i)} \sum_{i=1}^r \frac{\Phi_i}{\widehat{\lambda}_j - \lambda_i}, \quad j \in \mathcal{J}_1.$$

*Proof.* The proof immediately follows from a calculation of residues, e.g., for  $j \in \mathcal{J}_0$ ,  $R_j = \lim_{s \rightarrow \lambda_j} (s - \lambda_j) \mathbf{G}(s)$ .  $\square$

The first formula in (2.9) (for  $j \in \mathcal{J}_0$ ) is a special case of the second one, which is actually given in (2.5). This reflects the fact that the change of variables (2.5) in the transition from SK to VF iterations implicitly seeks a pole-residue representation.

**2.3. Computing  $\Phi_j^{(k+1)}$  and  $\varphi_j^{(k+1)}$ .** Measurements are naturally kept in a tensor  $\mathbb{S} \in \mathbb{C}^{p \times m \times \ell}$  with  $\mathbb{S}(:, :, i) = \mathbf{S}^{(i)} = \mathbf{H}(\xi_i) + \mathcal{E}^{(i)} \in \mathbb{C}^{p \times m}$ .  $\mathbf{S}^{(i)}$  denotes a sampling of the transfer function at the node  $\xi_i$ , allowing also for measurement errors to be represented through  $\mathcal{E}^{(i)}$ . Notice that the mode-3 fiber  $\mathbb{S}(u, v, :) = (S_{uv}^{(1)}, S_{uv}^{(2)}, \dots, S_{uv}^{(\ell-1)}, S_{uv}^{(\ell)})^T \in \mathbb{C}^\ell$ , represents connections between input  $v$  and output  $u$  over all frequency samples.

The expression for  $\epsilon^{(k)}$  can be matricized in several natural ways. The one followed in the VF literature is derived from a pointwise matching of input-output relations

<sup>1</sup>The general case of multiple poles is just more technical and it follows by standard residue calculus.

over all measurements. This is natural as it decouples the problem into  $p \cdot m$  rational function approximations having a common set of poles. In terms of the matrix entries, the least squares error (2.6) can be written as

$$\begin{aligned} \epsilon^{(k)} &= \sum_{i=1}^{\ell} \rho_i \sum_{u=1}^p \sum_{v=1}^m \left| \sum_{j=1}^r \left( \frac{(\Phi_j^{(k+1)})_{uv}}{\xi_i - \lambda_j^{(k+1)}} - S_{uv}^{(i)} \frac{\varphi_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}} \right) - S_{uv}^{(i)} \right|^2 \\ (2.10) \quad &= \sum_{v=1}^m \sum_{u=1}^p \left\| \mathcal{D}_\rho \left( \mathcal{C}^{(k+1)}, -D^{(uv)} \mathcal{C}^{(k+1)} \right) \begin{pmatrix} \Phi^{(k+1)}(u, v, \cdot) \\ \varphi^{(k+1)} \end{pmatrix} - \mathcal{D}_\rho \mathbb{S}(u, v, \cdot) \right\|_2^2, \end{aligned}$$

where  $\mathcal{D}_\rho = \text{diag}(\sqrt{\rho_i})$ ,  $D^{(uv)} = \text{diag}(S_{uv}^{(i)})_{i=1}^\ell$ ,  $\varphi^{(k+1)} = (\varphi_1^{(k+1)}, \dots, \varphi_r^{(k+1)})^T$ , and

$$(2.11) \quad \mathcal{C}^{(k+1)} = \begin{pmatrix} \frac{1}{\xi_1 - \lambda_1^{(k+1)}} & \frac{1}{\xi_1 - \lambda_2^{(k+1)}} & \cdots & \frac{1}{\xi_1 - \lambda_r^{(k+1)}} \\ \frac{1}{\xi_2 - \lambda_1^{(k+1)}} & \frac{1}{\xi_2 - \lambda_2^{(k+1)}} & \cdots & \frac{1}{\xi_2 - \lambda_r^{(k+1)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\xi_{\ell-1} - \lambda_1^{(k+1)}} & \frac{1}{\xi_{\ell-1} - \lambda_2^{(k+1)}} & \cdots & \frac{1}{\xi_{\ell-1} - \lambda_r^{(k+1)}} \\ \frac{1}{\xi_\ell - \lambda_1^{(k+1)}} & \frac{1}{\xi_\ell - \lambda_2^{(k+1)}} & \cdots & \frac{1}{\xi_\ell - \lambda_r^{(k+1)}} \end{pmatrix}, \quad \Phi^{(k+1)}(u, v, \cdot) = \begin{pmatrix} (\Phi_1^{(k+1)})_{uv} \\ (\Phi_2^{(k+1)})_{uv} \\ \vdots \\ (\Phi_{r-1}^{(k+1)})_{uv} \\ (\Phi_r^{(k+1)})_{uv} \end{pmatrix}.$$

To minimize (2.10), it is convenient to introduce the QR factorizations for  $1 \leq u \leq p$ ,  $1 \leq v \leq m$ :

$$(2.12) \quad \mathcal{D}_\rho \left( \mathcal{C}^{(k+1)}, -D^{(uv)} \mathcal{C}^{(k+1)} \right) = \begin{pmatrix} \vdots & \vdots & \vdots & * & * & * \\ \vdots & \vdots & \vdots & * & * & * \\ \vdots & \vdots & \vdots & * & * & * \\ \vdots & \vdots & \vdots & * & * & * \\ \vdots & \vdots & \vdots & * & * & * \\ \vdots & \vdots & \vdots & * & * & * \\ \vdots & \vdots & \vdots & * & * & * \end{pmatrix} = Q_{uv}^{(k+1)} \begin{pmatrix} (R^{(k+1)})_{11} & (R_{uv}^{(k+1)})_{12} \\ 0 & (R_{uv}^{(k+1)})_{22} \\ 0 & 0 \end{pmatrix}$$

$$(2.13) \quad = \left( (Q^{(k+1)})_1 \quad (Q_{uv}^{(k+1)})_2 \quad (Q_{uv}^{(k+1)})_3 \right) \begin{pmatrix} * & * & * & \times & \times & \times \\ 0 & * & * & \times & \times & \times \\ 0 & 0 & * & \times & \times & \times \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where the unitary matrix  $Q_{uv}^{(k+1)}$  has been partitioned as

$$Q_{uv}^{(k+1)} = \left( (Q^{(k+1)})_1 \quad (Q_{uv}^{(k+1)})_2 \quad (Q_{uv}^{(k+1)})_3 \right),$$

with block columns of sizes  $\ell \times r$ ,  $\ell \times r$ ,  $\ell \times (\ell - 2r)$ , respectively. The leading  $r$  columns of (2.12) are independent of  $(u, v)$  hence the initial part of the factorization,  $(Q^{(k+1)})_1 (R^{(k+1)})_{11} = \mathcal{D}_\rho \mathcal{C}^{(k+1)}$ , need only be done once.

The LS residual norm may be decomposed as  $\epsilon^{(k)} = \epsilon_1^{(k)} + \epsilon_2^{(k)} + \epsilon_3^{(k)}$ , where

$$(2.14) \quad \begin{aligned} \epsilon_1^{(k)} &= \sum_{v=1}^m \sum_{u=1}^p \left\| (R^{(k+1)})_{11} \Phi^{(k+1)}(u, v, \cdot) + (R_{uv}^{(k+1)})_{12} \varphi^{(k+1)} - (Q^{(k+1)})_1^* \mathcal{D}_\rho \mathbb{S}(u, v, \cdot) \right\|_2^2, \\ \epsilon_2^{(k)} &= \sum_{v=1}^m \sum_{u=1}^p \left\| (R_{uv}^{(k+1)})_{22} \varphi^{(k+1)} - (Q_{uv}^{(k+1)})_2^* \mathcal{D}_\rho \mathbb{S}(u, v, \cdot) \right\|_2^2, \quad \text{and} \\ \epsilon_3^{(k)} &= \sum_{v=1}^m \sum_{u=1}^p \left\| (Q_{uv}^{(k+1)})_3^* \mathcal{D}_\rho \mathbb{S}(u, v, \cdot) \right\|_2^2, \end{aligned}$$



where we have used  $M^*$  to denote the conjugate transpose of a matrix  $M$ . Here,  $\epsilon_3^{(k)}$  is a part of the residual that is beyond the reach of the unknowns  $\Phi_j^{(k+1)}, \varphi_j^{(k+1)}$  – it corresponds to the component in the data that is orthogonal to the subspace of rational functions available with the current barycentric nodes. Only a set of new (better) barycentric nodes will incline this subspace toward the data in such a way as to reduce this component of the error. Extracting optimal information from a given subspace associated with the current barycentric nodes is achieved by minimizing  $\epsilon_1^{(k)} + \epsilon_2^{(k)}$ .

To that end, first note that we can assume that  $(R^{(k+1)})_{11}$  is nonsingular since it participates in a QR factorization of  $\mathcal{D}_\rho \mathcal{C}^{(k+1)}$ , which in turn must have full column rank since  $\{\lambda_j^{(k)}\}_{j=1}^r$  are presumed distinct and we (tacitly) assume that  $\xi_i \neq \lambda_j^{(k)}$  throughout the iteration. Thus, we can make  $\epsilon_1^{(k)}$  exactly zero for any choice of  $\varphi^{(k+1)}$  by taking, concurrently for all input-output pairs  $(u, v)$ , with  $1 \leq u \leq p$  and  $1 \leq v \leq m$ ,

$$(2.15) \quad \Phi^{(k+1)}(u, v, :) = (R^{(k+1)})_{11}^{-1} \left( (Q^{(k+1)})_1^* \mathcal{D}_\rho \mathcal{S}(u, v, :) - (R_{uv}^{(k+1)})_{12} \varphi^{(k+1)} \right).$$

When the poles of the approximant are assigned to the barycentric nodes,  $\{\lambda_j^{(k+1)}\}_{j=1}^r$ , then the residues in (2.7) may be found by solving (2.15) with  $\varphi^{(k+1)} = 0$ . For other cases, the optimal  $\varphi^{(k+1)}$  will be found by minimizing  $\epsilon_k^{(2)}$ , and this process is uncoupled from any information about  $\Phi^{(k+1)}$ . However, determining  $\varphi^{(k+1)}$  does involve a coupled minimization across all input-to-output  $(u, v)$  pairs. The computation (2.15) is skipped if we choose to proceed with the SK iterations.

Once the maximal number of iterations is reached without reducing  $\varphi^{(k)}$  enough to be neglected, then  $\varphi^{(k+1)}$  in (2.15) also cannot be neglected without losing information. That is why the approximant defined in (2.7) uses the poles  $\lambda_j^{(k_*+1)}$ . It follows from section 2.2 that this is equivalent to solving (2.15) with  $\varphi^{(k_*)}$  included in the right-hand sides and transforming the computed rational approximant from barycentric into pole-residue form. This is another elegant feature built into the Gustavsen–Semlyen VF framework. (If we use the original SK iterations with diagonal scalings and fixed poles, then the barycentric form can be transformed to pole-residue representation using Proposition 2.1.)

**2.4. Global structure.** This procedure can be represented as  $\|\mathbf{A}^{(k+1)}x - \mathbf{b}\|_2 \rightarrow \min$ , in the usual form, as follows. First, we specify that the indices  $(u, v)$  in the  $p \times m$  array will be vectorized in a column-by-column fashion,  $(u, v) \rightsquigarrow \iota_{uv} = p(v-1) + u$ . Define  $\mathbf{A}^{(k+1)}$  to be a block matrix, with  $pm \times (pm+1)$  block structure, each block of dimensions  $\ell \times r$ . Only  $2pm$  out of  $pm(pm+1)$  blocks are a priori nonzero. Initially, set  $\mathbf{A}^{(k+1)}$  to zero and update it as follows: in the block row  $\iota_{uv}$  set the diagonal block to  $\mathcal{D}_\rho \mathcal{C}^{(k+1)}$  and the last block in the row to  $-\mathcal{D}_\rho D^{(uv)} \mathcal{C}^{(k+1)}$ . The right-hand side is the vector  $\mathbf{b}$  with the  $pm \times 1$  block structure, where the  $\iota_{uv}$ th block is set to  $\mathcal{D}_\rho \mathcal{S}(u, v, :)$ . The concurrent QR factorizations (2.12) can be then represented by premultiplying  $\mathbf{A}^{(k+1)}$  with unitary block-diagonal matrix  $(Q^{(k+1)})^*$ , with the  $\iota_{uv}$ th diagonal block set to  $(Q_{uv}^{(k+1)})^*$ . It is easily seen that the rows of  $(Q^{(k+1)})^* \mathbf{A}^{(k+1)}$  can be permuted to obtain the structure illustrated in (2.16):<sup>2</sup>

<sup>2</sup>Elements not displayed are zeros.



**Algorithm 1** Vector Fitting–Basic Iterations.

- 
- 1: Given: The sampling data  $\mathbf{H}(\xi_i)$  for  $i = 1, \dots, \ell$ ; maximal number of iterations  $k_{\max}$ .
  - 2: Set  $k \leftarrow 0$  and make an initial pole selection  $\boldsymbol{\lambda}^{(k+1)} \in \mathbb{C}^r$ .
  - 3: **while** { stopping criterion not satisfied and  $k \leq k_{\max}$  } **do**
  - 4:   Form  $\mathbf{A}^{(k+1)}$  and  $\mathbf{b}$ .
  - 5:   Compute  $\mathbf{B}^{(k+1)} = \Pi(\mathcal{Q}^{(k+1)})^* \mathbf{A}^{(k+1)}$  and  $\mathbf{s}^{(k+1)} = \Pi(\mathcal{Q}^{(k+1)})^* \mathbf{b}$  and partition as in (2.17).
  - 6:   Solve  $\|\mathbf{B}_{[22]}^{(k+1)} \boldsymbol{\varphi}^{(k+1)} - \mathbf{s}_2^{(k+1)}\|_2 \rightarrow \min$  for  $\boldsymbol{\varphi}^{(k+1)}$ .
  - 7:   Set  $k \leftarrow k + 1$  and compute  $\boldsymbol{\lambda}^{(k+1)} = \text{zeros}(1 + \sum_{j=1}^r \varphi_j^{(k)} / (s - \lambda_j^{(k)}))$ .
  - 8: **end while**
  - 9:  $\boldsymbol{\Phi} = (\mathbf{B}_{[11]}^{(k)})^{-1} \mathbf{s}_1^{(k)}$ .
- 

*Remark 2.1.* Observe that the iteration on  $\boldsymbol{\varphi}^{(k)}$  proceeds independently of  $\boldsymbol{\Phi}$  and, upon convergence ( $\boldsymbol{\varphi}^{(k)} \rightarrow 0$ ),  $\boldsymbol{\Phi}$  is obtained only in the final step, line 9 of Algorithm 1, which accomplishes the simultaneous determination of residues by minimizing  $\|\mathcal{D}_\rho(\mathcal{E}^{(k+1)}\boldsymbol{\Phi}^{(k+1)}(u, v, \cdot) - \mathbb{S}(u, v, \cdot))\|_2$ , for  $u = 1, \dots, p$ ,  $v = 1, \dots, m$ . This observation was first exploited in [23]. This may be implemented as a solution of an LS problem with multiple right-hand sides, and additional measures can be taken to compute more accurate residues, see section 3.4.2. Stopping criteria (line 3 of Algorithm 1) will be discussed in section 3.4.3.

**3. Numerical issues that arise in standard VF.** The key variables in VF are computed as solutions of LS problems, where the coefficient matrices are built from Cauchy and diagonally scaled Cauchy matrices, thus potentially highly ill-conditioned. Further, as we hope to capture the data by reducing the residual, we also expect cancellation to take place. These issues pose tough challenges to numerical analyst during the finite precision implementation of the algorithm. In this section, we discuss several important details that are at the core of a robust implementation of VF.

**3.1. Least squares solution and rank revealing QR factorization.** To fully understand the global behavior of VF iterations in finite precision arithmetic, it is crucial to investigate all the details of an LS solver used in a robust software implementation. For example, consider line 6 in Algorithm 1, i.e., consider the LS problem  $\|\mathbf{B}_{[22]}\boldsymbol{\varphi} - \mathbf{s}\|_2 \rightarrow \min$  where we now drop all superfluous indices to ease notation. In a MATLAB implementation, the solution is obtained using the backslash operator, i.e.,  $\boldsymbol{\varphi} = \mathbf{B}_{[22]} \backslash \mathbf{s}$ , or using the pseudoinverse, i.e.,  $\boldsymbol{\varphi} = \text{pinv}(\mathbf{B}_{[22]})\mathbf{s}$ , computed using the SVD and an appropriate threshold for determining numerical rank. The state of the art LAPACK library [1] provides driver routines `xgelsy`, based on a complete orthogonal decomposition, and `xgelsd`, `xgelsd`, based on the SVD decomposition.

We briefly describe the decomposition approach. In the first step, the column pivoted QR factorization is computed and written in partitioned form:

$$(3.1) \quad \mathbf{B}_{[22]}P = WT = \begin{pmatrix} W_1 & W_2 \end{pmatrix} \begin{pmatrix} T_{[11]} & T_{[12]} \\ 0 & T_{[22]} \end{pmatrix}, \quad W^*W = \mathbb{I}, \quad \|T_{[22]}\|_F \leq \epsilon \|T_{[11]}\|_F,$$

where  $\epsilon$  is a threshold value, e.g.,  $\epsilon = n\varepsilon$ , with  $\varepsilon$  denoting the machine round-off unit. As a consequence of the Businger–Golub pivoting [12],

$$(3.2) \quad |T_{ii}| \geq \sqrt{\sum_{j=i}^k |T_{jk}|^2}, \quad 1 \leq i \leq k \leq r.$$

In the case of differently weighted rows of  $\mathbf{B}_{[22]}$ , numerical stability can be enhanced by using Powell–Reid complete pivoting [52] or by presorting the rows in order of decreasing  $\infty$ -norm [16]. The actual size of  $T_{[11]}$  may be determined by an incremental condition number estimator, or by inspecting for gaps in the sequence  $|T_{11}| \geq |T_{22}| \geq \dots \geq |T_{nm}|$ . If no such partition is possible, then  $T = T_{[11]}$  and the block  $T_{[22]}$  is void. In ill-conditioned cases, as we could have in Algorithm 1, such a partition is likely to be visible; see, e.g., Figure 1. Then,  $T_{[22]}$  is deemed negligible noise and set to zero. This is justified by backward error analysis: there exists a small perturbation of the initial matrix  $\mathbf{B}_{[22]}$  such that this part of the triangular factor is exactly zero. Then, an additional orthogonal reduction transformation  $Z$  (similar to QR factorization, see, e.g., LAPACK routine `xtzrzf`) is deployed from the right leading to a URV decomposition:

$$(3.3) \quad \mathbf{B}_{[22]}P \approx W \begin{pmatrix} \widehat{T}_{[11]} & 0 \\ 0 & 0 \end{pmatrix} Z \quad \text{and the solution } \varphi = PZ^* \begin{pmatrix} \widehat{T}_{[11]}^{-1} W_1^* \mathbf{s} \\ 0 \end{pmatrix}.$$

The vector  $\varphi$  in (3.3) is the minimal Euclidean norm solution of a nearby (backward perturbed) problem. However, in the rank deficient case, other particular choices from the solution manifold might be of interest. For instance, once we set  $T_{[22]}$  in (3.1) to zero, we can use

$$(3.4) \quad \mathbf{B}_{[22]}P \approx W \begin{pmatrix} T_{[11]} & T_{[12]} \\ 0 & 0 \end{pmatrix} \quad \text{and the solution } \varphi^{(0)} = P \begin{pmatrix} T_{[11]}^{-1} W_1^* \mathbf{s} \\ 0 \end{pmatrix},$$

with the same residual norm as  $\varphi$  in (3.3). Note that while the MATLAB command  `$\mathbf{B}_{[22]} \backslash \mathbf{s}$`  computes  $\varphi^{(0)}$ , the SVD based  `$\text{pinv}(\mathbf{B}_{[22]})\mathbf{s}$`  and the LAPACK routines return the minimal norm solution  $\varphi$ . These details may also significantly impact the computation in line 9 of Algorithm 1 (cf. Remark 2.1). Numerical rank deficiency will trigger truncation and in the case of the solution method in (3.4), some of the residues in (2.7) will be computed as  $p \times m$  zero matrices, thus effectively removing the corresponding poles from  $\mathbf{H}_r(s)$ . We discuss this residue computation step in more detail in section 3.4.2.

Determining the numerical rank is a delicate procedure and it should be tailored to a particular application, based on all available information and interpretation of the solution. For instance, what is a sensible choice for the threshold  $\epsilon$  in (3.1) and how do we decide whether to prefer the solution of minimal Euclidean length or the solution with most zero entries? What can we infer from the numerical rank of  $\mathbf{B}_{[22]}$ ? These issues are further discussed in section 3.2.

**3.2. Convergence introduces noise.** In this section, we analyze and illustrate that as VF proceeds, the coefficient matrices in the pole identification phase tend to become noisy with a significant drop in column norms that may coincide also with a reduction in numerical rank. This prompts us to advise caution when rescaling columns in order to improve the condition number, since rescaling columns that have

been computed through massive cancellations will preclude inferring an accurate numerical rank. Since VF simultaneously fits all input-output pairs using a common set of poles, it suffices to focus our analysis on only one fixed input-output pair  $(u, v)$ . For simplicity of notation, we drop the iteration index  $k$ , and take unit weights, i.e.,  $\mathcal{D}_\rho = \mathbb{I}_\ell$ . If  $r$  is large enough and the poles have settled, then, with some small *error*,

$$\mathbb{S}(u, v, 1 : \ell) \approx \mathcal{C}x + \text{error} = (Q)_1(R)_{11}x + \text{error},$$

where  $x = \Phi(u, v, 1 : r)$ ; see Remark 2.1. Here we used the QR factorization (2.12). Now, the right-hand side in the error contribution  $\epsilon_2$  in (2.14) that corresponds to the pair  $(u, v)$  is

$$(3.5) \quad (Q_{uv})_2^* \mathbb{S}(u, v, 1 : \ell) = (Q_{uv})_2^* (Q)_1 (R)_{11} x + (Q_{uv})_2^* \text{error} = (Q_{uv})_2^* \text{error}.$$

The vectors of the structure (3.5) are building the vector  $\mathbf{s}_2$  in (2.17). Furthermore, using (2.12),

$$(R_{uv})_{22} = (Q_{uv})_2^* \underbrace{\text{diag}(\mathbb{S}(u, v, 1 : \ell))}_{\ell \times \ell} \mathcal{C} = (Q_{uv})_2^* \{[(\mathcal{C}x + \text{error}) (1 \ \dots \ 1)] \circ \mathcal{C}\},$$

where  $\circ$  denotes the Hadamard product. Hence, a  $j$ th column of  $(R_{uv})_{22}$  reads

$$(R_{uv})_{22}(:, j) = (Q_{uv})_2^* \left\{ [(Q)_1 (R)_{11} x + \text{error}] \circ \begin{pmatrix} 1/(\xi_1 - \lambda_j) \\ \vdots \\ 1/(\xi_\ell - \lambda_j) \end{pmatrix} \right\},$$

which means that  $(R_{uv})_{22}(:, j)$  could also be small, depending on the position of  $\lambda_j$  relative to the  $\xi_i$ s. The LS coefficient matrix  $\mathbf{B}_{[22]}$  in line 6 of Algorithm 1 is assembled from the matrices  $(R_{uv})_{22}$ , and we can expect that it will have many small entries that are (when computed in floating point arithmetic) mostly contaminated by the roundoff noise. We illustrate this on an example.

*Example 3.1.* We use the one-dimensional heat diffusion equation model [14], obtained by spatial discretization of  $\frac{\partial}{\partial t} T(x, t) = \alpha \frac{\partial^2}{\partial x^2} T(x, t) + u(x, t)$ ,  $0 < x < 1$ ,  $t > 0$  with the zero boundary and initial conditions. The discretized system is of order  $n = 197$ . We generate  $\ell = 1000$  samples and set  $r = 80$ . The structures of  $\mathbf{B}_{[22]}$  and its column pivoted triangular factor are given in Figure 1. The column norms of  $\mathbf{B}_{[22]}$  in the first step are so particularly ordered due to the ordering of the initial poles  $\lambda_j = \alpha_j \pm i\beta_j$ , where the  $\beta_j$ s are logarithmically spaced between the minimal and the maximal sampling frequency and  $\alpha_j = -\beta_j$ . Note the sharp drop in the column norms in the second iteration, after the relocated poles induced better approximation.

**3.3. The quandary of column scaling.** In the VF literature, it is often recommended to scale the columns of the LS coefficient matrix in line 6 of Algorithm 1, to make them all of the same Euclidean length, before deploying the *backslash* solver, and then to rescale the solution; see, e.g., [35]. One desirable effect of this column equilibration step is to reduce the effective condition number of LS coefficient matrix (see, e.g., [45, 57]). While this can be beneficial, nonetheless this tactic also may have a variety of deleterious effects and, in our opinion, must be considered with caution. Of foremost concern, following the discussion from section 3.2, is that scaling noisy matrix columns effectively increases the influence of noise, and allows these noisy columns to participate in the column pivoting process of the QR factorization, with a possibility that some of them become drafted and taken upfront as important. This,

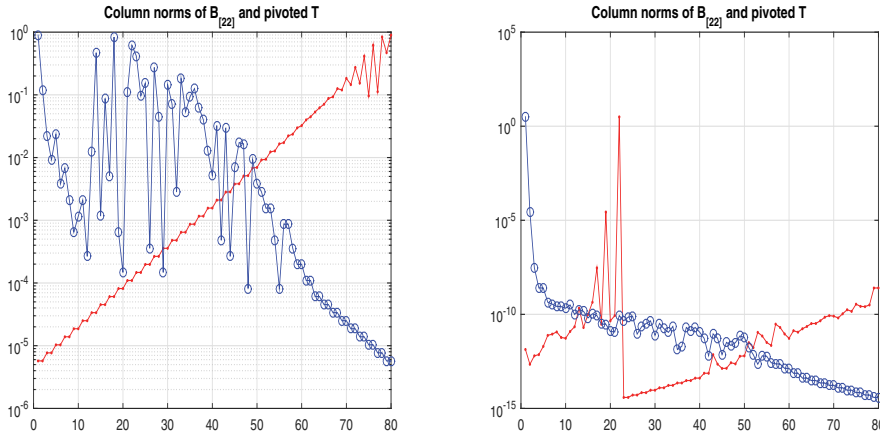


FIG. 1. The structure of the matrix  $\mathbf{B}_{[22]}$  and its column pivoted triangular factor in line 6 of Algorithm 1, during the first two iterations in Algorithm 1. In the first plot, showing the data in the first iteration, the column norms of  $\mathbf{B}_{[22]}$  are marked with (red)  $-$  and the column norms its triangular factor in the Businger–Golub pivoted QR factorization are marked with (blue)  $\circ-$ . The second plot shows the same information, but in the second iteration.

in turn, interferes with the rank revealing process and possibly precludes truncation based on a partition as in (3.1). For illuminating discussions related to this issue we refer to [31, 30, 58]. In the following two examples we illustrate the potentially baleful effects of column scaling in the particular context of VF iteration.

*Example 3.2.* We continue using the heat model from Example 3.1. In Figure 2 we show, for the first two iterations, the moduli of the diagonal entries ( $|T_{ii}|$ ) of the triangular factors of  $\mathbf{B}_{[22]}$  (appearing in line 6 of Algorithm 1) without and with column equilibration. Note that, due to the diagonal dominance (3.2), the distribution of  $|T_{ii}|$  is decisive for numerical rank revealing.

We now illustrate how the numerical rank deficiency may be manifested during the VF iterations. We solve the LS problems for the  $\varphi^{(k)}$ s using only a simple modification of MATLAB’s *backslash* operator: first reorder the equations so that the rows of the coefficient matrix have decreasing  $\ell_\infty$  norms, and then apply *backslash*.<sup>3</sup> This stabilizes the LS solution process in much the same way as does Powell–Reid complete pivoting (see [16, 41]). We use 1000 frequencies and  $r = 80$ . The samples are matched perfectly with both our implementation of VF and *vectfit3* [56] (up to relative errors of the order of  $10^{-13}$ ). The first plot in Figure 3 shows the structure of the denominators  $\varphi^{(k)}$  throughout ten iterations. Each  $\varphi^{(k)}$  is represented by the sorted vector of  $\log_{10}(|\varphi^{(k)}|/\|\varphi^{(k)}\|_1)$ , and the zero entries of  $\varphi^{(k)}$  are not shown. For  $k = 1, \dots, 10$ , the values of  $\text{sum}(|\varphi^{(k)}|/\|\varphi^{(k)}\|_\infty > \varepsilon)$  are, respectively, 53, 43, 38, 30, 44, 44, 44, 40, 42, 44. (If we restart the approximation with  $r = 53$ , the number of nonzero coefficients throughout the iterations are 40, 36, 38, 41, 41, 42, 41, 41, 39, 41.) To interpret these numbers, we compute the Hankel singular values  $\sigma_1 \geq \dots \geq \sigma_{197}$  and superimpose them on the graph as  $\log_{10}(\sigma_i/\sigma_1)$ —those values marked by  $\diamond$ . Since the  $\sigma_i$ s are forming a “devil’s staircase” and there is no clear cutoff index. For instance,  $\sigma_{30}/\sigma_1 \approx 1.28e - 11$ ,  $\sigma_{36}/\sigma_1 \approx 2.17e - 13$ ,  $\sigma_{44}/\sigma_1 \approx 1.12e - 16$ ,  $\sigma_{53}/\sigma_1 \approx 9.00e - 17$ .

<sup>3</sup>Recall the discussion in section 3.1. In a LAPACK implementation, instead of *backslash*, one uses *xgelsy*, which calls *xgeqp3* to compute the QR decomposition (3.1), (3.2) and then determines the partition (3.1) based on estimated condition numbers of the leading submatrices of  $T$ .

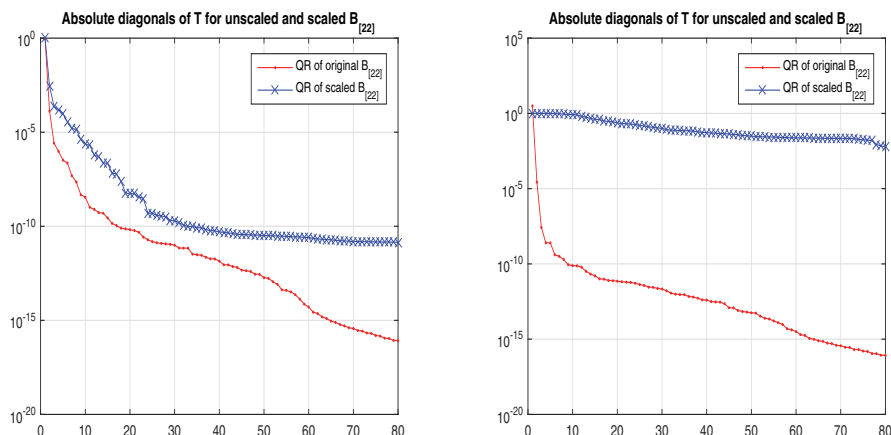


FIG. 2. (Example 3.1 cont.) The structure of the pivoted triangular factors (cf. (3.2)) of  $\mathbf{B}_{[22]}$  in the first two iterations in Algorithm 1. The plot shows the values  $|T_{ii}|$  of the unscaled  $\mathbf{B}_{[22]}$  (marked with (red)  $\cdot$ ) and of the column equilibrated  $\mathbf{B}_{[22]}$  (marked with (blue)  $\times$ ).

(If we use the backslash without the initial row pivoting, the numbers of nonzero coefficients throughout the iterations are 53, 44, 37, 42, 50, 50, 50, 49, 49, 50.)

Recall from Proposition 2.1 that the barycentric nodes corresponding to  $\varphi_j^{(k)} = 0$  are the poles of the current approximant  $\mathbf{H}_r^{(k)}$  and that the corresponding residues are accessible by an explicit formula (2.9), thus allowing an estimate of the contribution of the pole and perhaps discarding it and reducing  $r$ . The matching of the number of the untruncated entries in the LS solutions  $\varphi^{(k)}$  with the number of significant Hankel singular values is striking and may offer machinery to readjust the order of the approximant,  $r$ , during the iterations. This matching cannot be guaranteed in general, in particular for systems whose Hankel singular values show no significant decay—in that case the problem of finding good approximation of lower order is intrinsically difficult. The right-hand side plot in Figure 3 also shows  $\log_{10}(|\varphi^{(k)}|/|\varphi^{(k)}|_1)$ , but with the column scaling of the least squares coefficient matrix and rescaling the solution, as in `vectfit3`. Note that, once the scaling is applied, connection to the Hankel singular values decay is lost; supporting our discussion on the undesirable effects of column scaling in VF.

Full understanding of the potential of VF for determining the order of the underlying system and its connection with the Loewner framework [48] remains an important open problem. To illustrate this connection, consider the SISO case and choose  $\lambda_j = \xi_j$ ,  $\Phi_j = \varphi_j \mathbf{H}(\xi_j)$   $j = 1, \dots, r$ , which ensures  $\mathbf{H}_r(\xi_j) = \mathbf{H}(\xi_j)$ . If we require that LS fit to the remaining data (indexed by  $i = r + 1, \dots, \ell$ ), then the LS coefficient matrix has a Loewner structure.

**3.4. mimoVF: Putting the pieces together.** In this section, based on our preceding analysis of VF for matrix-valued rational approximation problem, we start testing our new implementation of VF. We will call the new implementation and the corresponding MATLAB toolbox `mimoVF`. This section will provide examples for verification and validation of `mimoVF`. We will compare our implementation with

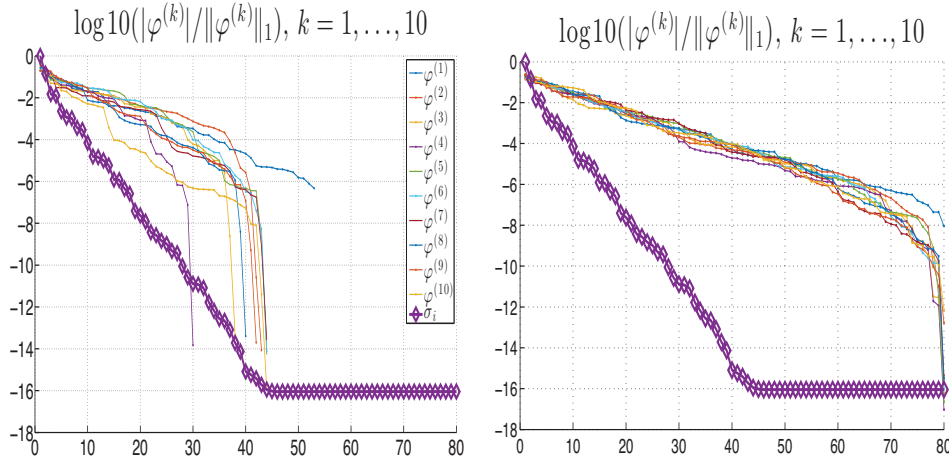


FIG. 3. (Example 3.2.) History of the first ten iterations,  $k = 1, \dots, 10$ . Plots show  $\log_{10}(|\varphi_j^{(k)}|/\|\varphi^{(k)}\|_1)$  versus  $j$ ; only nonzero coefficients are shown. Normalized Hankel singular values are represented as diamonds.

the original `vectfit3` [56]<sup>4</sup> and show that our proposed modifications based on the theoretical analysis can substantially improve the results. While on average `vectfit3` performs well, in the ill-conditioned cases, it has difficulties with numerical issues addressed in this paper.

For the resulting rational approximation  $\mathbf{H}_r$ , define the tensor  $\mathbb{S}_r(:, :, i) = \mathbf{H}_r(\xi_i)$ ,  $i = 1, \dots, \ell$ , and the relative LS error as  $\gamma = \|\mathbb{S} - \mathbb{S}_r\|_F / \|\mathbb{S}\|_F$ . Recall that  $\mathbb{S}(:, :, i) = \mathbf{H}(\xi_i) \in \mathbb{C}^{p \times m}$ ,  $i = 1, \dots, \ell$ , contains the original samples that are either measurements, or computed from a state space realization of the underlying LTI dynamical system.

**3.4.1. A stress test.** We consider a model for the ISS 1R module [14] with  $m = p = 3$ . The underlying dynamical system has dimension  $n = 270$ . This example presents a difficult test case with rather vivid dynamics. The model is very hard to approximate and presents significant challenges to model reduction; see [33]. To that end, we choose  $r = 50$  and take  $\ell = 300$  samples. The initial barycentric nodes are chosen as the eigenvalues of a pseudorandom real stable  $r \times r$  matrix, a potentially poor initialization. Using these initial nodes, two iteration steps are taken both in `vectfit3` and `mimoVF`. The goal of this example is to illustrate that once the computations in each iteration step are made more robust (following our preceding analysis), high-fidelity rational approximants can still be achieved even with a small number iteration count or even in the cases of poorly initial choices of barycentric nodes. The results of `vectfit3` and `mimoVF` are shown on Figure 4 where we depict the amplitude frequency response plots for the data and for the rational approximants. Note that this model has  $m = 3$  inputs and  $p = 3$  outputs, there are nine input/output channels corresponding to the different lines in Figure 4. This figure clearly illustrates the `mimoVF` performs significantly better than `vectfit3` for this example. Recall that both functions are given the same set of initial poles. Despite this unfavorable choice of initial poles, a restricted number of iterations, and ill-conditioned LS matrices, `mimoVF` succeeds to compute a model with relative LS error below  $\gamma \approx 6.45 \cdot 10^{-3}$ .

<sup>4</sup>The options used deploy the “relaxed vector fitting” technique [36].



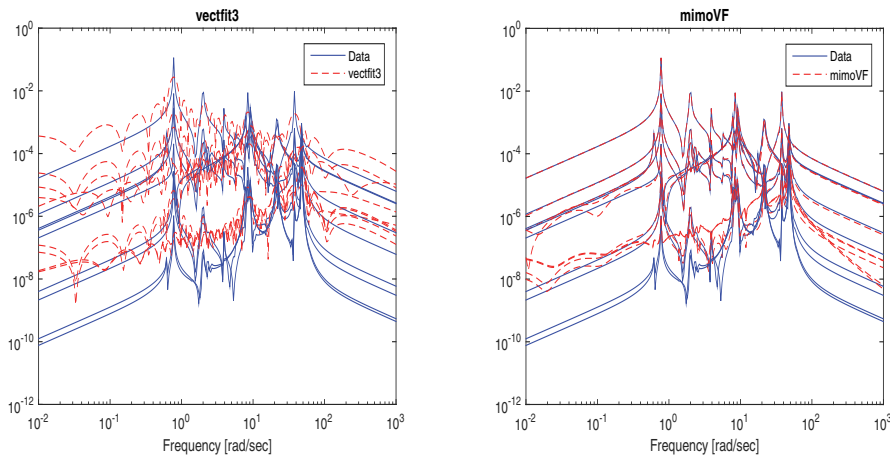


FIG. 4. (Example of section 3.4.1.) Comparison of *mimoVF* and the *vectfit3* on the ISS 1R module with initial poles set as the eigenvalues of a pseudorandom stable real matrix,  $\ell = 300$ , and  $r = 50$ . (The frequency response magnitudes of each of the possible nine input/output pairings is plotted in solid blue; the corresponding frequency response magnitudes from rational approximations provided by *vectfit3* and *mimoVF* appear as dashed red lines.) The purpose of the experiment is to check the robustness of the numerical implementation in the case of unpropitious distribution of the barycentric nodes. The first plot shows the output of *vectfit3* (with the relative error  $\gamma > 10$ ), and the second of *mimoVF* ( $\gamma < 10^{-2}$ ), both after two iterations.

On the other hand, the relative error due to *vectfit3* is  $\gamma \approx 10.41$ ; a significantly higher value than the error due to *mimoVF*. It is possible that allowing *vectfit3* to iterate further might realign the poles better, leading to a smaller LS error and an accurate approximate. But, of course, this comes with additional costs since every step of the iteration requires solving a potentially large-scale LS problem; especially when  $m$  and  $p$  are large. Therefore, any reduction in the iteration count is a gain in terms of computational efficiency.

**3.4.2. How to compute the residues in the final step.** One of the advantages of the barycentric implementation of the SK iterations over the original approach using polynomial representations is in the avoidance of high powers of  $\xi_i$  (which may cause overflow and underflow in finite precision arithmetic) as producing ill-conditioned Vandermonde matrices. The additional scalings by  $1/|d^{(k)}(\xi_i)|^2$ , which is another potential source of ill-conditioning, has been elegantly removed by the VF formulation and compensated by reallocating the barycentric nodes  $\lambda_j^{(k)}$ . However, once the VF iterations are completed, one needs to solve for the final residues  $\Phi$  in line 9 of Algorithm 1 for the converged poles. This step needs to be performed carefully as the coefficient matrix that determines the residues for a given set of poles is a Cauchy matrix, which, together with Vandermonde matrices, is among the most notoriously ill-conditioned matrices. To illustrate, the spectral condition number of an arbitrary  $100 \times 100$  real Vandermonde matrix is larger than  $3 \cdot 10^{28}$ , and the condition number of the  $100 \times 100$  Hilbert (Cauchy) matrix is more than  $10^{150}$ . The column norms of the latter are between 0.07 and 1.3, thus no column scaling can substantially reduce the condition number. Furthermore, the additional weightings  $\rho_i$  (whose values may spread many orders of magnitude) may further worsen the conditioning of the least squares coefficient matrix. All of this is a menace to the final computed residues, in

particular when the order  $r$  is sufficiently high and in cases of unfavorably distributed nodes. This issue has to be addressed if the method is to be applied to truly challenging problems with complex dynamics and of high orders, for instance, for  $m, p, r$  in hundreds. In this section we focus our attention to the very last step—given poles of a rational approximant, how to best numerically extract the residues.

*Example 3.3.* We continue using the ISS 1R example from section 3.4.1. However, in this case, we choose good initial poles of the form  $\lambda_j = \alpha_j \pm i\beta_j$ , where the (positive)  $\beta_j$ s are log-spaced over the frequency sample interval and  $\alpha_j = -\beta_j < 0$  as often recommended in the VF literature for good initial pole selection. We take  $\ell = 500$  and choose  $r = 100$ . Recall that the underlying system has dimension  $n = 270$  with  $p = m = 3$ ; therefore, with  $r = 100$  and  $\ell = 500$  samples, one expects to obtain a very good approximation even after one iteration. The amplitude frequency response plots are depicted in the first row of Figure 5 for both `vecfit3` and `mimoVF`; the relative LS errors due to `mimoVF` and `vecfit3` are, respectively,  $4.23 \cdot 10^{-2}$  and  $4.90 \cdot 10^{-3}$ . (Interestingly, the relative errors in the system  $\mathcal{H}_2$  norm (4.1) were  $2.80 \cdot 10^{-2}$  for `mimoVF` and  $2.19 \cdot 10^1$  for `vecfit3`.) In the next experiment, we use the initial poles with  $\alpha_j = -\beta_j/100 < 0$ , and linearly spaced  $\beta_j$ 's, as suggested in [38, relations (9), (10)], [37], and allow both procedures to run two iterations. The results are shown in the second row of Figure 5, once again illustrating that `mimoVF` outperforms `vecfit3`.

In this case, `vecfit3` suffers from the numerical ill-conditioning of the final residue computation. Thus, this example shows that even a plenty of good initial barycentric nodes, one does not necessarily guarantee a good approximation due to the numerical issues arising in the residue computation step.

**A regularization approach for residue extraction.** Even though `mimoVF` performed relatively well in Example 3.3, some of the less-dominant input/output pairs were not captured as accurately as one would prefer; see Figure 5. In this section, we consider a regularization technique to further improve the final residue extraction step in `mimoVF`. Recall line 9 of Algorithm 1 to compute the final residues: solving  $\Phi = \mathbf{B}_{[11]}^{-1} \mathbf{s}_1$ . As noted in Remark 2.1, this corresponds to the simultaneous determination of the residue matrices by solving  $\|\mathcal{D}_\rho (\mathcal{C}^{(k+1)} \Phi^{(k+1)}(u, v, \cdot) - \mathbb{S}(u, v, \cdot))\|_2 \rightarrow \min$ ,  $u = 1, \dots, p$ ,  $v = 1, \dots, m$ . To simplify the notation, denote this LS problem by  $\|\mathcal{D}_\rho \mathcal{C}x - h\|_2 \rightarrow \min$ , where  $\mathcal{C} = \mathcal{C}_{\xi, \lambda}$  is a Cauchy matrix as in (2.11),  $h$  is the corresponding scaled right-hand side,  $\lambda$  is closed under conjugation, and the solution vector should also be closed under conjugation. Such a constrained problem can be replaced by an equivalent unconstrained LS problem

$$(3.6) \quad \left\| \begin{pmatrix} \mathcal{D}_\rho \mathcal{C}_{\xi, \lambda} \\ \mathcal{D}_\rho \mathcal{C}_{\bar{\xi}, \lambda} \end{pmatrix} x - \begin{pmatrix} h \\ \bar{h} \end{pmatrix} \right\|_2 \equiv \|\widehat{\mathcal{C}}x - \widehat{h}\|_2 \rightarrow \min$$

with the coefficient matrix again of the diagonally scaled Cauchy structure,  $\widehat{\mathcal{C}} = (\mathcal{D}_\rho \oplus \mathcal{D}_\rho) \mathcal{C}_{(\xi, \bar{\xi}), \lambda}$ .

The SVD of  $\widehat{\mathcal{C}}$  can be computed to high relative accuracy based on the pivoted LU decomposition  $\Pi_1 \widehat{\mathcal{C}} \Pi_2 = LDU$ , where each entry (including the tiniest ones) of the computed factors  $\widetilde{L}$ ,  $\widetilde{D}$ ,  $\widetilde{U}$  is computed to high relative accuracy, and  $\widetilde{L}$ ,  $\widetilde{U}$  are well conditioned. The ill-conditioning of  $\widehat{\mathcal{C}}$  is revealed in the diagonal matrix  $\widetilde{D}$ . This decomposition can be used immediately in an LS solver [13], or it can be used to compute an accurate SVD [19, 17, 28, 29] which is then used to compute an approximate LS solution.  $\widehat{\mathcal{C}}$  can be severely ill-conditioned so that small changes of  $\xi$ 's and  $\lambda$ 's can cause significant perturbation of the SVD. However, we can consider

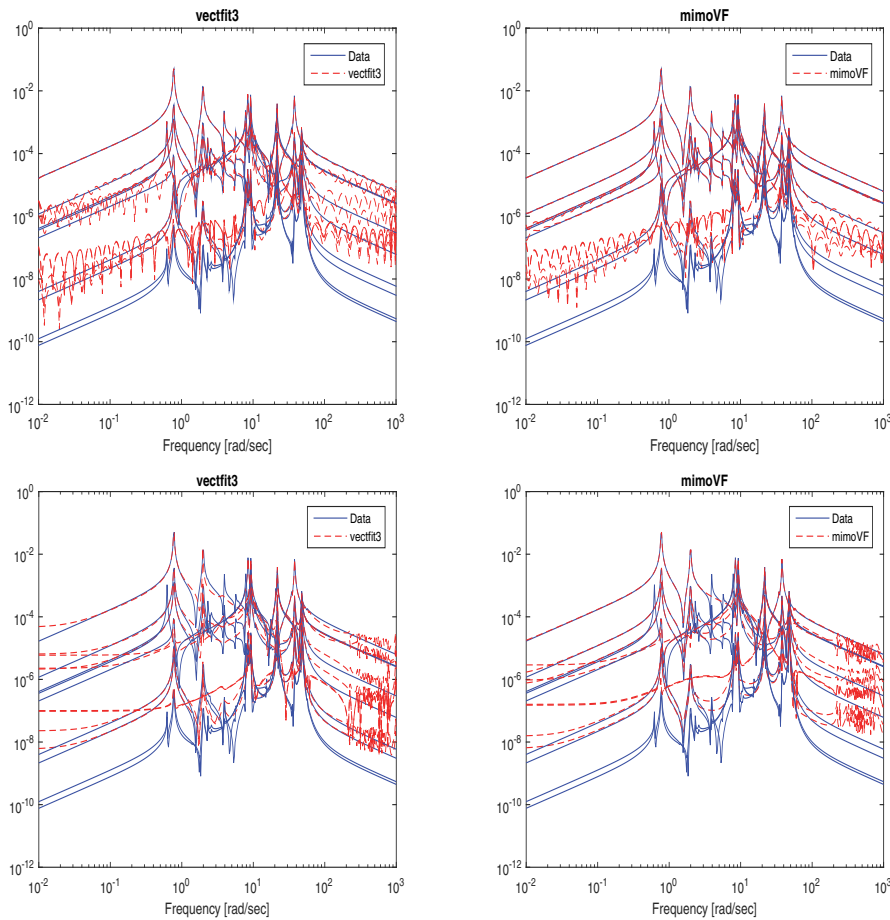


FIG. 5. (Example 3.3.) Comparison of *mimoVF* and the *vectfit3* on an ISS example. First row: The initial poles  $\lambda_j = \alpha_j \pm i\beta_j$  set with imaginary parts  $\beta_j > 0$  log-spaced over the frequency range,  $\alpha_j = -\beta_j$ . The first plot shows the output of *vectfit3* (with the relative error  $\gamma \approx 4.23 \cdot 10^{-2}$ ), and the second of *mimoVF* ( $\gamma \approx 4.90 \cdot 10^{-3}$ ) after one iteration. Second row: Following [38], the  $\beta_j$ 's are linearly spaced and  $\alpha_j = -\beta_j/100$ . The relative errors after two iterations are  $\gamma \approx 1.60 \cdot 10^{-1}$  (*vectfit3*) and  $\gamma \approx 7.00 \cdot 10^{-2}$  (*mimoVF*). Since the order of the underlying system is  $n = 270$  and  $p = m = 3$ ,  $r = 100$  should provide good approximation.

the values of  $\xi_i$ s and  $\lambda_j$ s, as stored in the machine memory, as exact and attempting to compute accurate SVD is justified.

Let  $\widehat{\mathcal{C}} = W\Sigma V^*$  be the SVD and let the unique<sup>5</sup> LS solution be  $x = V\Sigma^\dagger W^* = \sum_{i=1}^r v_i(w_i^* \widehat{h})/\sigma_i$ . Unfortunately, an accurate SVD is not enough to have the LS solution computed to high relative accuracy, and additional regularization techniques must be deployed. This is, in particular, important if the right-hand side is contaminated by noise. In the Tichonov regularization, we choose  $\mu \geq 0$  and use the solution of  $\|\widehat{\mathcal{C}}x - \widehat{h}\|_2^2 + \mu^2 \|x\|_2^2 \rightarrow \min$ , explicitly computable as

$$(3.7) \quad x_\mu = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \mu^2} (w_i^* \widehat{h}) v_i.$$

<sup>5</sup>Since all nodes are distinct and the poles are assumed simple, the matrix is of full column rank.

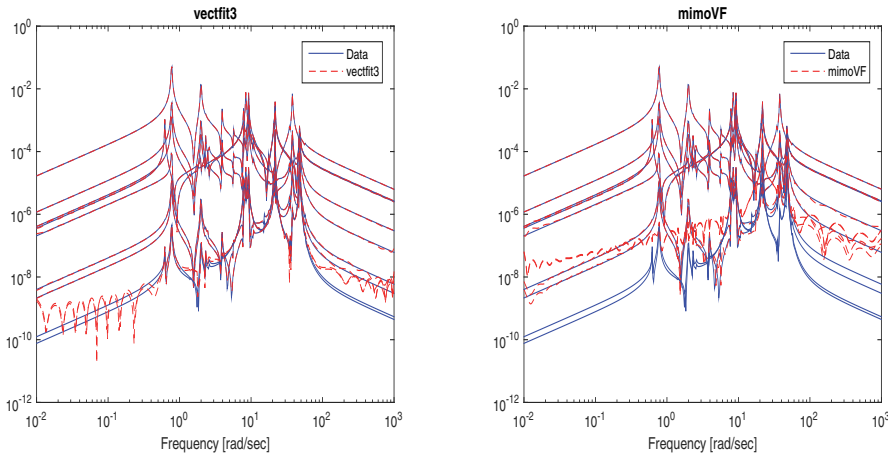


FIG. 6. Comparison of *mimoVF* and the *vectfit3* on an ISS example with initial poles set as *log-spaced* (imaginary parts *log-spaced* over the frequency range). In this example,  $\ell = 500$  and  $r = 100$ . The first plot shows the output of *vectfit3* after two iterations (with the relative error  $\gamma \approx 7.95 \cdot 10^{-4}$  and the  $\mathcal{H}_2$  error  $1.05 \cdot 10^{-1}$ ), and the second of *mimoVF* ( $\gamma \approx 4.90 \cdot 10^{-3}$  and the  $\mathcal{H}_2$  error  $2.81 \cdot 10^{-2}$ ) after one iteration.

The parameter  $\mu$  can be further adjusted using the Morozov discrepancy principle [50], i.e., to achieve  $\|\widehat{\mathcal{E}}x_\mu - \widehat{h}\|_2 \approx \nu$ , where  $\nu$  is the estimated level of noise  $\delta\widehat{h}$  in the right-hand side,  $\nu \approx \|\delta\widehat{h}\|_2$ .

*Example 3.4.* Here we continue Example 3.3, use the same data, and apply *vectfit3* and *mimoVF* where we use the preceding regularization approach in the final residue computation step. Using accurate SVD of  $\widehat{\mathcal{E}}$  we compute  $x_\mu$  as in (3.7) with an ad hoc choice of  $\mu = 10^{-3}$ . The result after *one* iteration of *mimoVF*, shown in the right plot in Figure 6, when compared to the right plot in the first row of Figure 5 indicates an improvement in approximation of nondominant input-output pairs, with nearly the same LS error  $\gamma \approx 4.90 \cdot 10^{-3}$ . Running *two* iterations of *vectfit3*, as shown in the left plot of Figure 6, reduces the error to  $\gamma \approx 7.95 \cdot 10^{-4}$ , and the  $\mathcal{H}_2$  error drops to  $1.05 \cdot 10^{-1}$ , which is still larger than the corresponding  $\mathcal{H}_2$  error from one iteration of *mimoVF* ( $2.81 \cdot 10^{-2}$ ).

It should also be noted that, for both *vectfit3* and *mimoVF*, the quality of VF approximations usually rapidly improves throughout the iteration, in particular due to a procedure introduced by Gustavsen [36].<sup>6</sup> However, our goal in this section is to show that a more robust implementation may reduce the total number of iterations needed to reach satisfactory approximation. The numerical issues discussed here are crucial when the dimensions  $n, r, m, p$  are so large that the ill-conditioning of the underlying matrices impedes the convergence of VF to a good approximation.

*Remark 3.1.* The LDU decomposition, which is the initial step of the accurate SVD, can also be used in the pole identification phase to compute accurate QR factorization of Cauchy matrices at all stages of the computation. We have not included those modifications for the sake of brevity of the presentation. Interested readers can find the details of such an approach in [25].

<sup>6</sup>Full mathematical explanation of this procedure is an important challenging problem.

**3.4.3. Stopping criterion.** It should be noted that the stopping criterion in the VF framework is rather vaguely specified. To the best of our knowledge, the VF literature does not provide a precise and numerically justified strategy of halting the iterations. For instance, `vectfit3` allows only running for a given fixed number of iterations.

Following our recent analysis [27], we propose to declare the  $|\varphi_j^{(k+1)}|$ s “small enough” if

$$(3.8) \quad \sum_{j=1}^r \frac{|\varphi_j^{(k+1)}|}{|\Re(\lambda_j^{(k+1)})|} \equiv \theta^{(k+1)} \leq \epsilon, \quad \text{where } \epsilon \text{ is a suitable threshold.}$$

This seems appropriate because  $\max_{s \in \mathbb{Z}\mathbb{R}} |d^{(k+1)}(s) - 1| \leq \theta^{(k+1)}$ , and in (2.6) we can write

$$\mathbf{S}^{(i)} \left( 1 + \sum_{j=1}^r \frac{\varphi_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}} \right) = \mathbf{S}^{(i)} + \Delta \mathbf{S}^{(i)}, \quad \|\Delta \mathbf{S}^{(i)}\|_F \leq \theta^{(k+1)} \|\mathbf{S}^{(i)}\|_F,$$

thus interpreting this as introducing backward perturbation into the data. In fact, this interpretation can be a guidance for choosing the threshold value  $\epsilon$  by following a discrepancy principle, i.e., so that this backward error matches the estimated size of the noise level on the input.

**3.5. Guidelines for efficient implementation.** We now further discuss implementation details that are relevant for an efficient software implementation of `mimoVF` or Algorithm 1 in general. Recall that VF for MIMO systems with  $m$  input and  $p$  inputs will require  $p \cdot m$  QR factorizations (2.12) of size  $\ell \times (2r)$  before advancing toward finding  $\varphi_j^{(k+1)}$ . It is clear that this is a demanding computational challenge even for moderate  $m$  and  $p$ ; e.g., in the case  $m = p = 100$ , it will require  $p \cdot m = 10000$  QR factorizations of size  $\ell \times (2r)$ . Since these factorizations are independent, they can be very efficiently parallelized and the whole computation can be optimized for a multicore computing machinery. This has been nicely described by China and Grivet-Talocia [15], who showed a nearly ideal speedup on a four quad-core architecture.

**3.5.1. Efficient computation of  $\mathbf{B}_{[22]}$ .** It was pointed out earlier that the QR factorizations in (2.12) are independent of  $(u, v)$  in the first  $r$  columns and  $(R^{(k+1)})_{11}$  from (2.15) can be computed by a single QR factorization, optionally with column pivoting, of  $\mathcal{D}_\rho \mathcal{L}^{(k+1)}$ , i.e.,  $\mathcal{D}_\rho \mathcal{L}^{(k+1)} \Pi = V^{(k+1)} \begin{pmatrix} T^{(k+1)} \\ 0 \end{pmatrix}$ . Further, the introduction of rank-revealing column pivoting  $\Pi$  in this factorization incurs a negligible overhead, while preserving the structure (2.16). We discussed in section 3.1 that this pivoting is very important for numerical robustness of the LS solution as well. In an LAPACK-style implementation, the matrix  $V^{(k+1)}$  can be computed and stored in the form of  $r$  Householder vectors (using `Xgeqp3`), and then, using `Xormqr`,  $(V^{(k+1)})^*$  can be concurrently applied to all  $-\mathcal{D}_\rho D^{(uv)} \mathcal{L}^{(k+1)}$ ,  $u = 1, \dots, p$ ,  $v = 1, \dots, m$ . Then, it only remains to compute the QR factorizations of the  $(r+1 : \ell, r+1 : 2r)$  submatrices of  $-(V^{(k+1)})^* \mathcal{D}_\rho D^{(uv)} \mathcal{L}^{(k+1)}$ . One should note that in a blocked QR factorization a similar computation is done anyway in the process of computing the QR factorizations (2.12). The computed triangular  $r \times r$  factors are the  $(2, 2)$  blocks in (2.12) that build the matrix  $\mathbf{B}_{[22]}^{(k+1)}$ . Hence, the saving of this modified approach is equivalent to the cost of  $(pm - 1)$  QR factorizations of size  $\ell \times r$ , or, approximately,  $const \cdot (pm - 1)\ell r^2$ .

The total work on the QR factorizations (2.12) without this modification is  $4 \cdot \text{const} \cdot pmlr^2$ .

Further, when we are solving only for the  $\varphi_j^{(k)}$  during the VF iterations, the elements of  $\mathcal{Q}^*A$ , denoted by  $\times$  in (2.13) and (2.16) are not used in the pole identification phase, but they are computed as  $(R_{uv}^{(k+1)})_{12}$  parts of the QR factorizations (2.12). On the other hand, once the poles are fixed, the LS problem is solved with the approximant of the form (2.2), but with the unit denominator,  $d^{(k)}(s) \equiv 1$ , or, equivalently, with  $\varphi_j^{(k+1)} = 0$ ,  $j = 1, \dots, r$ . This means that, for computing (2.7) by Algorithm 1, we do not compute the matrices  $(R_{uv}^{(k+1)})_{12}$ , which further reduces the complexity. Our implementation of this more efficient approach is based on adapting the LAPACK's functions Xormqr, Xlarft, and Xlarfb.

**3.5.2. Locally pivoted factorization.** In the presence of ill-conditioning and noise, pivoting is essential when using the QR factorization. Hence, we propose including pivoting in the procedure outlined in section 3.5.1. More precisely, the QR factorization of  $\mathcal{D}_\rho \mathcal{C}^{(k+1)}$  is computed with column pivoting, i.e.,  $\mathcal{D}_\rho \mathcal{C}^{(k+1)} \Pi^{(k+1)} = V^{(k+1)} \begin{pmatrix} T^{(k+1)} \\ 0 \end{pmatrix}$ . This enhances the accuracy of the computed residues in line 9 of Algorithm 1. In a software implementation, line 9 is reshaped into the LS problem with the coefficient matrix  $T^{(k+1)}$  and with  $m \cdot p$  right-hand sides, for all input-output pairs. Optionally, one can use the truncation discussed in section 3.1, or the accurate SVD as explained in section 3.4.2.

Further, we also advocate the use of pivoting when computing the QR factorizations of the  $(r+1 : \ell, r+1 : 2r)$  submatrices of  $-(V^{(k+1)})^* \mathcal{D}_\rho D^{(uv)} \mathcal{C}^{(k+1)}$ . This increases the accuracy of the computed matrix  $\mathbf{B}_{[22]}^{(k+1)}$  in line 6 of Algorithm 1. For details how pivoting influences the accuracy of the rows of the computed triangular QR factor, we refer to [26]. Furthermore, this may allow (in the cases of numerical rank deficiency, as revealed by the pivoted QR factorization and discussed in section 3.1) one set certain numbers of rows of  $(R_{uv}^{(k+1)})_{22}$  to zero and thus increase the number of zero rows in  $\mathbf{B}_{[22]}^{(k+1)}$  and reduce the complexity of line 6 in Algorithm 1, where, as described in section 3.1, the LS solver starts with the QR factorization with column pivoting.

The pivoted QR factorization of the tall and skinny matrix  $\mathbf{B}_{[22]}^{(k+1)}$  can be computed by, e.g., first computing the QR without pivoting using the techniques of [9], and then computing the pivoted QR factorization of the computed  $r \times r$  triangular factor, or, e.g., as in [18]. These approaches become particularly attractive if  $p \cdot m$  and  $r$  are large.

*Remark 3.2.* Interestingly, we do not need to compute the  $(R_{uv}^{(k+1)})_{22}$ s. Instead, we can build the matrix  $\mathbf{B}_{[22]}^{(k+1)}$  from the  $(r+1 : \ell, r+1 : 2r)$  submatrices of  $-(V^{(k+1)})^* \mathcal{D}_\rho D^{(uv)} \mathcal{C}^{(k+1)}$ . This will lead to increased number of rows in  $\mathbf{B}_{[22]}^{(k+1)}$ , but overall it reduces the complexity with potential gain increased if the QR factorization of  $\mathbf{B}_{[22]}^{(k+1)}$  is computed using the strategies of [9, 18].

#### 4. Numerical quadrature in mimoVF for discretized $\mathcal{H}_2$ approximation.

The framework for mimoVF is based on the algebraic least squares (LS) error minimization (1.1) where one usually chooses the weights  $\rho_j = 1$  and the nodes  $\xi_j$  are usually selected heuristically. In [27], for SISO systems, we have shown that with the underlying dynamical system in mind, reformulating the discrete LS problem as discretization of an underlying continuous  $\mathcal{H}_2$  error measure and then choosing the

nodes and weights by an appropriate numerical quadrature improves the performance of VF significantly. The same conclusion holds in the MIMO case as well since, once the common set of poles has been determined, `mimoVF` works separately on each input–to–output pair. We illustrate these considerations briefly in this section.

**4.1.  $\mathcal{H}_2$  approximation and numerical quadrature.** The algebraic least squares error is closely related to the  $\mathcal{H}_2$  system norm. More precisely, consider the space  $\mathcal{H}_{2,+}^{p \times m}$  of  $p \times m$  matrix functions  $\mathbf{M}(s)$ , analytic in the open right half-plane  $\mathbb{C}_+ = \{s \in \mathbb{C} : \Im(s) > 0\}$ , such that  $\sup_{x>0} \int_{-\infty}^{\infty} \|\mathbf{M}(x + iy)\|_F^2 dy < \infty$ . The space  $\mathcal{H}_{2,+}^{p \times m}$  is a Hilbert space with the associated inner product and norm defined by

$$(4.1) \quad \begin{aligned} \langle \mathbf{M}_1, \mathbf{M}_2 \rangle_{\mathcal{H}_2} &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Trace} \left( \overline{\mathbf{M}_1(i\omega)} \mathbf{M}_2(i\omega)^T \right) d\omega, \\ \|\mathbf{M}\|_{\mathcal{H}_2} &= \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{M}(i\omega)\|_F^2 d\omega \right)^{1/2}. \end{aligned}$$

The  $\mathcal{H}_2$  approximation problem is, then, to find a degree- $r$  rational approximant,  $\mathbf{H}_r(s)$ , that minimizes the  $\mathcal{H}_2$  error norm  $\|\mathbf{H} - \tilde{\mathbf{H}}_r\|_{\mathcal{H}_2}$  over all degree- $r$  rational function  $\tilde{\mathbf{H}}_r(s)$ . Such an optimal rational approximant must satisfy certain Hermite tangential interpolation conditions; for details we refer to [5, 34]. The *Iterative Rational Krylov Algorithm* (IRKA) of Gugercin, Antoulas, and Beattie [34] is a numerically effective iterative algorithm that constructs degree- $r$  rational approximants satisfying the  $\mathcal{H}_2$ -optimality conditions.

Our goal in this section is to repeat the success of [27] for SISO systems, i.e., improve the performance of `mimoVF` by formulating the discrete LS measure as discretization of the continuous  $\mathcal{H}_2$  error. Towards this goal, approximate the  $\mathcal{H}_2$  error with a quadrature rule to obtain

$$(4.2) \quad \begin{aligned} \int_{-\infty}^{+\infty} \|\mathbf{H}(i\omega) - \mathbf{H}_r(i\omega)\|_F^2 d\omega &\approx \sum_{j=1}^{\ell} \rho_j^2 \|\mathbf{H}(\xi_j) - \mathbf{H}_r(\xi_j)\|_F^2 \\ &+ \rho_+^2 M_+ [|\mathbf{H} - \mathbf{H}_r|^2] + \rho_-^2 M_- [|\mathbf{H} - \mathbf{H}_r|^2], \end{aligned}$$

where  $M_{\pm}[G]$  are linear functionals of  $G$  that capture information about asymptotic behavior of  $G$  at  $\pm\infty$ . Note that the usual VF formulation correspond to  $\rho_+ = \rho_- = 0$ , with all other  $\rho_j = 1$ , and choosing sampling nodes  $\xi_j$  to be equidistant and in complex conjugate pairs. Thus, the usual VF objective function is as a composite trapezoid quadrature rule for the integral in (4.2) approximating the  $\mathcal{H}_2$  error. As we discussed in [27], more effective quadrature options may be considered, e.g., Gauss–Legendre, Gauss–Kronrod, and Gauss–Hermite quadrature rules. We do not go into the details of what quadrature method to choose here; since our main point is just to illustrate that `mimoVF` can perform much better once formulated as a discretized  $\mathcal{H}_2$  measure.

**4.2. A numerical example.** Here, with a simple example, we illustrate the effect of choosing the sampling points and the weights using numerical quadrature. We use the Clenshaw–Curtis type quadrature rule developed by Boyd [11]. We use the ISS 1R module [33] with  $m = 3$  inputs and  $p = 3$  outputs. We use only 100 function evaluations ( $\ell = 200$ ) and apply `mimoVF` to this data set for different  $r$  values. Resulting relative  $\mathcal{H}_2$  errors are shown in Table 1 below. The quadrature-based selection yields the smallest error in each case; for  $r = 20$  and  $r = 30$ , it leads to one order of magnitude improvements.

TABLE 1  
Effect of quadrature nodes and weights on *mimoVF*.

| Method                           | $r = 10$                | $r = 20$                | $r = 30$                | $r = 40$                |
|----------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| <i>mimoVF</i> without quadrature | $3.4104 \times 10^{-1}$ | $3.1775 \times 10^{-1}$ | $1.2778 \times 10^{-1}$ | $5.0257 \times 10^{-2}$ |
| <i>mimoVF</i> with quadrature    | $2.5209 \times 10^{-1}$ | $4.6074 \times 10^{-2}$ | $3.3226 \times 10^{-2}$ | $2.1436 \times 10^{-2}$ |

**5. Controlling the McMillan degree.** Let  $\{\Phi_j\}_1^r \subset \mathbb{C}^{p \times m}$  and  $\{\lambda_j\}_1^r \subset \mathbb{C}$  denote the final set of matrix residues and poles, respectively, resulting from *mimoVF*. To ease notational clutter, we drop the iteration index  $k$ . The associated rational matrix approximant can be represented as

$$(5.1) \quad \mathbf{H}_r(s) = \sum_{j=1}^r \frac{\Phi_j}{s - \lambda_j} = (\mathbb{I}_p \dots \mathbb{I}_p) \begin{pmatrix} \frac{\mathbb{I}_p}{s - \lambda_1} & & & \\ & \ddots & & \\ & & \frac{\mathbb{I}_p}{s - \lambda_r} & \\ & & & \ddots \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_r \end{pmatrix} \\ = (\Phi_1 \dots \Phi_r) \begin{pmatrix} \frac{\mathbb{I}_m}{s - \lambda_1} & & & \\ & \ddots & & \\ & & \frac{\mathbb{I}_m}{s - \lambda_r} & \\ & & & \ddots \end{pmatrix} \begin{pmatrix} \mathbb{I}_m \\ \vdots \\ \mathbb{I}_m \end{pmatrix}.$$

If  $\mathbf{H}_r(s)$  has simple poles, then  $\mathbf{H}_r(s)$  has nominal McMillan degree  $\deg(\mathbf{H}_r(s)) = \sum_{j=1}^r \text{rank}(\Phi_j)$ . Evidently,  $r \leq \deg(\mathbf{H}_r(s)) \leq r \min(p, m)$ . Note that  $\deg(\mathbf{H}_r(s))$  will be strictly larger than  $r$  unless the residues,  $\Phi_j$ , all have rank 1, and indeed,  $\deg(\mathbf{H}_r(s))$  can be potentially much larger than  $r$  if either the input space or output space has significant dimension. McMillan degree is a proxy for the complexity involved in evaluating an approximant; therefore, it is generally desirable and sometimes necessary to reduce the McMillan degree of a rational approximant to the target value of  $r$ , while retaining its approximating quality as much as possible. One straightforward method to accomplish this is to use truncation as suggested in [39]: For each  $j = 1, \dots, r$ , find the best rank-one approximation of  $\Phi_j \approx c_j b_j^T$ , where  $c_j \in \mathbb{C}^p$  and  $b_j \in \mathbb{C}^m$ . However, it is often the case that the residues  $\Phi_j$  are not close to rank-one matrices and so, truncation to rank-one residues can substantially increase the LS error. Reference [39] suggested using the SVD to determine the numerical ranks of the  $\Phi_j$ s and truncating them to their respective best low-rank (not necessarily rank-one) approximations; [39] also proposed Gauss–Newton correction, but no details on how to proceed in this direction were provided.

We propose here two different approaches toward retaining rank-one residues, allowing us to achieve a true McMillan degree of  $r$  while keeping the approximation quality as high as possible. The first approach, presented in section 5.1, is based on a nonlinear least-squares minimization, the second one, presented in section 5.2, combines *mimoVF* with well-established optimal systems-theoretic model reduction methodologies.

**5.1. Rank-one residue correction via alternating least squares.** We seek an optimal rational approximant,  $\hat{\mathbf{H}}_r(s)$ , having the same poles,  $\{\lambda_j\}_1^r$ , as the *mimoVF* approximant, but taking the form

$$(5.2) \quad \hat{\mathbf{H}}_r(s) = \sum_{j=1}^r \frac{c_j b_j^T}{s - \lambda_j} \equiv C(s\mathbb{I} - \Lambda)^{-1} B^T, \quad \text{where } \Lambda = \text{diag}(\lambda_j)_{j=1}^r, \quad C = (c_1 \dots c_r) \\ B = (b_1 \dots b_r).$$



Optimality here will mean that  $C$  and  $B$  are chosen so that  $\widehat{\mathbf{H}}_r(s)$  satisfies

$$(5.3) \quad \min_{C, B} \sum_{i=1}^{\ell} \left\| \sum_{j=1}^r \frac{c_j b_j^T}{\xi_i - \lambda_j} - \mathbf{H}(\xi_i) \right\|_F^2 = \min_{C, B} \left\| \underbrace{(\mathcal{C} \otimes \mathbb{I}_p)}_{\mathcal{M}} \begin{pmatrix} c_1 b_1^T \\ \vdots \\ c_r b_r^T \end{pmatrix} - \begin{pmatrix} \mathbf{H}(\xi_1) \\ \vdots \\ \mathbf{H}(\xi_\ell) \end{pmatrix} \right\|_F^2.$$

A weighting factor  $\rho_i > 0$  can also be attached to each sample; yet for simplicity of presentation, we take all  $\rho_i = 1$ . In (5.3),  $\mathcal{C} \in \mathbb{C}^{\ell \times r}$  denotes the Cauchy matrix  $\mathcal{C}_{ij} = 1/(\xi_i - \lambda_j)$ . In many applications,  $\widehat{\mathbf{H}}_r(s)$  should be real-valued for real-valued  $s$ . In that case, a constraint is added that the poles  $\lambda_j$  and the residues  $c_j b_j^T$  must be closed under conjugation: all nonreal poles appear in complex conjugate pairs, say  $\lambda_j, \lambda_{j+1} = \bar{\lambda}_j$ ;  $c_j, b_j$  are real if  $\lambda_j$  is real, otherwise  $c_{j+1} = \bar{c}_j, b_{j+1} = \bar{b}_j$ .

The nonlinearity of the LS error (5.3) with respect to the variables,  $B$  and  $C$ , can be evaded by reformulating the problem in terms of alternating least squares (ALS): if  $B$  (alternatively,  $C$ ) is fixed, then (5.3) becomes a linear least squares problem in terms of  $C$  (alternatively,  $B$ ). So, we minimize alternately with respect to  $C$  (holding  $B$  fixed) and then with respect to  $B$  (holding  $C$  fixed), repeating the cycle until convergence. We provide some algorithmic details below and illustrate the effectiveness of ALS iteration with an example. An analogous approach has been used for “residue correction” in realization-independent (data-driven) approaches to optimal  $\mathcal{H}_2$  model reduction [7].

**Correction of  $C$ .** Assume  $B$  is fixed and seek an updated  $C$  (with conforming conjugation symmetry) that will minimize the LS error (5.3). To that end, we vectorize the error matrix columnwise and write the  $k$ th column ( $k = 1, \dots, m$ ) of the residual matrix in (5.3) as

$$(5.4) \quad \mathcal{M} \begin{pmatrix} (b_1)_k \mathbb{I} & & \\ & \ddots & \\ & & (b_r)_k \mathbb{I} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} \mathbf{H}(\xi_1) e_k \\ \vdots \\ \mathbf{H}(\xi_\ell) e_k \end{pmatrix} \\ = \mathcal{M}(\text{diag}(B(k, :)) \otimes \mathbb{I}_p) \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} \mathbf{H}(\xi_1) e_k \\ \vdots \\ \mathbf{H}(\xi_\ell) e_k \end{pmatrix}.$$

Stacking all columns together, the problem becomes: minimize the Euclidean norm of the residual

$$(5.5) \quad \begin{pmatrix} \mathcal{M}(\Delta_1 \otimes \mathbb{I}_p) \\ \vdots \\ \mathcal{M}(\Delta_m \otimes \mathbb{I}_p) \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} \mathbf{H}(\xi_1) e_1 \\ \vdots \\ \mathbf{H}(\xi_\ell) e_1 \\ \mathbf{H}(\xi_1) e_m \\ \vdots \\ \mathbf{H}(\xi_\ell) e_m \end{pmatrix}, \quad \text{where } \Delta_i = \text{diag}(B(i, :)), \quad i = 1, \dots, m,$$

with conjugation symmetry constraints:  $c_j$  is real if  $\lambda_j$  is real and  $c_k = \bar{c}_j$ , if  $\lambda_k = \bar{\lambda}_j$ . Note that the number of rows above is  $\ell \cdot m \cdot p$ , and the number of unknowns is  $p \cdot r$ . In practice,  $\ell$  is much larger than  $m, p, r$ , and it is always assumed that  $\ell \geq 2r$ .

Let  $\mathcal{C} = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = \widehat{Q}R$  be the QR factorization, where  $Q = (\widehat{Q}\check{Q})$ ,  $\widehat{Q} = Q(:, 1:r)$ . Then  $\mathcal{M} = (Q \otimes \mathbb{I}_p) \left( \begin{pmatrix} R \\ 0 \end{pmatrix} \otimes \mathbb{I}_p \right)$  is the QR factorization of  $\mathcal{M}$ . Multiplying the blocks in the residual (5.5) by  $(Q^* \otimes \mathbb{I}_p)$  and using the block-partitioned structure of  $Q$ , we obtain an equivalent LS problem:

$$(5.6) \quad \left\| \begin{pmatrix} (R \otimes \mathbb{I}_p)(\Delta_1 \otimes \mathbb{I}_p) \\ 0 \\ \vdots \\ (R \otimes \mathbb{I}_p)(\Delta_m \otimes \mathbb{I}_p) \\ 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} (\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, 1, :)) \\ (\check{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, 1, :)) \\ \vdots \\ (\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, m, :)) \\ (\check{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, m, :)) \end{pmatrix} \right\|_F \rightarrow \min.$$

The blocks  $(\check{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, i, :))$ ,  $i = 1, \dots, m$ , in the right-hand side constitute a part of the residual that cannot be influenced with any choice of the  $c_j$ s and the corresponding  $(\ell-r) \cdot p \cdot m$  equations (with the corresponding zero rows in the coefficient matrix) are dropped, i.e., only the thin QR factorization  $\mathcal{C} = \widehat{Q}R$  is needed. This reduces the row dimension of the problem from  $\ell \cdot p \cdot m$  to  $r \cdot p \cdot m$ . Using the properties of the Kronecker product, we can further simplify it to

$$(5.7) \quad \left\| \left[ \begin{pmatrix} R\Delta_1 \\ \vdots \\ R\Delta_m \end{pmatrix} \otimes \mathbb{I}_p \right] \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} (\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, 1, :)) \\ \vdots \\ (\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, m, :)) \end{pmatrix} \right\|_F \rightarrow \min.$$

To solve (5.7) we compute the QR factorizations

$$(5.8) \quad R_{\square} = U \begin{pmatrix} T \\ 0 \end{pmatrix}, \quad R_{\square} \otimes \mathbb{I}_p = (U \otimes \mathbb{I}_p) \left( \begin{pmatrix} T \\ 0 \end{pmatrix} \otimes \mathbb{I}_p \right), \quad \text{where } R_{\square} = \begin{pmatrix} R\Delta_1 \\ \vdots \\ R\Delta_m \end{pmatrix} \in \mathbb{C}^{m \cdot r \times r},$$

and, using the partition  $U = (\widehat{U} \quad \check{U})$ , we reduce the problem to solving the triangular system

$$(5.9) \quad (T \otimes \mathbb{I}_p) \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} = (\widehat{U}^* \otimes \mathbb{I}_p) \begin{pmatrix} (\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, 1, :)) \\ \vdots \\ (\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, m, :)) \end{pmatrix}.$$

Note that only the thin QR factorization  $R_{\square} = \widehat{U}T$  is needed. Folding the unknowns back into the structure of  $C$  we obtain, using that  $(\widehat{Q}^* \otimes \mathbb{I}_p) \text{vec}(\mathbb{S}(:, i, :)) = \text{vec}(\mathbb{S}(:, i, :)\widehat{Q}^{*T})$ ,

$$(5.10) \quad \begin{aligned} \text{vec}(CT^T) &= (\widehat{U}^* \otimes \mathbb{I}_p) \begin{pmatrix} \text{vec}(\mathbb{S}(:, 1, :)\widehat{Q}^{*T}) \\ \vdots \\ \text{vec}(\mathbb{S}(:, m, :)\widehat{Q}^{*T}) \end{pmatrix} \\ &= (\widehat{U}^* \otimes \mathbb{I}_p) \text{vec} \left( \left( \mathbb{S}(:, 1, :)\widehat{Q}^{*T} \quad \dots \quad \mathbb{S}(:, m, :)\widehat{Q}^{*T} \right) \right) \\ &= \text{vec} \left( \left( \mathbb{S}(:, 1, :)\widehat{Q}^{*T} \quad \dots \quad \mathbb{S}(:, m, :)\widehat{Q}^{*T} \right) \widehat{U}^{*T} \right). \end{aligned}$$

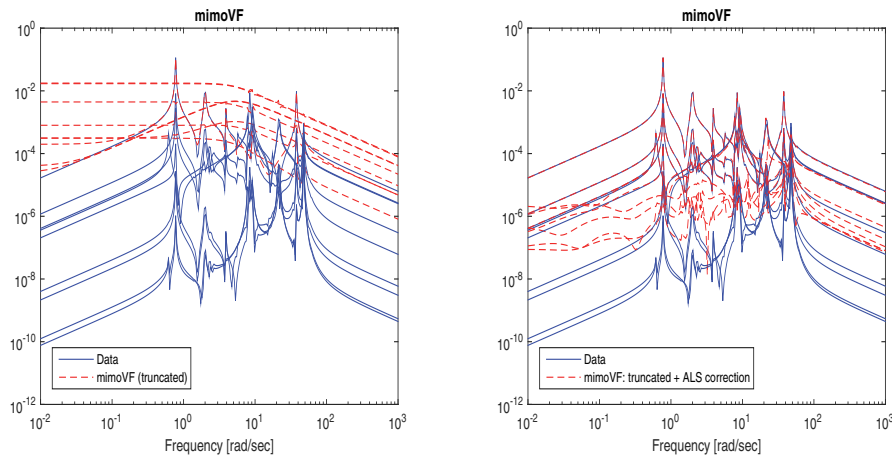


FIG. 7. (Example 5.1.) Illustration of the truncation of *mimoVF* and the ALS correction. (Truncation applied to the *mimoVF* output shown on the right graph on Figure 4. Only one ALS iteration is used.) The relative  $\mathcal{H}_2$  error of  $\hat{\mathbf{H}}_r$  is  $\chi \approx 1.50e - 01$ .

As an alternative to solving (5.9),  $C$  can be computed efficiently as the solution of a triangular matrix equation. The formula  $C = (\mathbb{S}(:, 1, :)\hat{Q}^{*T} \dots \mathbb{S}(:, m, :)\hat{Q}^{*T})\hat{U}^{*T}T^{-T}$  is rich in BLAS 3 operations and can be highly optimized. Finally, we note that the QR factorizations involved can be done with pivoting, but we omit details for the sake of simplicity.

**Correction of  $B$ .** If the matrix  $C$  is fixed and we want to update  $B$ , we use the preceding procedure, with a few simple modifications. First, transpose the residuals at each  $\xi_i$  to get  $\sum_{j=1}^r \frac{b_j c_j^T}{\xi_i - \lambda_j} - \mathbf{H}(\xi_i)^T$ . As a consequence, swap the roles of the  $c_j$ s and the  $b_j$ s, and use  $\mathbb{S}(i, :, :)$  instead of  $\mathbb{S}(:, i, :)$ . The rest follows *mutatis mutandis*.

**A numerical example.** We illustrate the usefulness of the ALS correction process in building a final approximant  $\hat{\mathbf{H}}_r$  that has exact McMillan degree  $r$ . We use the data of Example 3.4.1, and the output of *mimoVF* after the second iteration. The simple truncation of the residue matrices causes the LS error jump from  $\gamma \approx 6.45 \cdot 10^{-3}$  to  $\gamma \approx 2.72$ , and one step of ALS correction reduces it down to  $\gamma \approx 1.41 \cdot 10^{-2}$ . This improvement is evident in Figure 7.

**5.2. Rank-one residue correction via  $\mathcal{H}_2/\mathcal{H}_\infty$  model reduction approaches.** The ALS iteration described above is built purely upon algebraic least squares error minimization. However, if the underlying context relates the rational approximants to dynamical systems, then it may be advantageous to perform this reduction using well-developed systems-theoretic reduction tools. Recasting our rank-one residue correction problem into this setting, we consider constructing an  $r$ th order system  $\hat{\mathbf{H}}_r$  that closely approximates the VF computed model  $\mathbf{H}_r$  in some appropriate system norm.

The  $\mathcal{H}_2$  norm discussed in section 4 is the most natural choice and the first one we consider. This approach is compelling when weights and nodes in *mimoVF* are chosen using an appropriate quadrature as in section 4 and the algebraic LS measure is viewed as a discretized  $\mathcal{H}_2$  measure. In this case, the complete procedure; both *mimoVF* step and reduction to true McMillan degree- $r$  will be performed with the  $\mathcal{H}_2$  system norm in mind. To achieve this goal, i.e., to minimize  $\|\mathbf{H}_r - \hat{\mathbf{H}}_r\|_{\mathcal{H}_2}$  over

all stable  $r$ th order  $\widehat{\mathbf{H}}_r$ , we will apply the optimal  $\mathcal{H}_2$  approximation method IRKA of [34] as modified in [7] for a realization independent procedure.<sup>7</sup> Note that the output of `mimoVF`,  $\mathbf{H}_r(s)$ , has the McMillan degree up to  $r \min(p, m)$ ; thus  $\mathbf{H}_r(s)$  can have a modest state-space dimension. If, for example,  $r = 80$  and  $m = p = 50$ ,  $\mathbf{H}_r$  can have degree as high as 4000. Thus, it is important to perform this second reduction step effectively. A particularly attractive aspect of the IRKA framework of [7] is that it needs only function and derivative evaluations at dynamically generated points. This works perfectly in our setting since the explicit state-space form of the `mimoVF` output in (5.1) makes these computations trivial. This approach can be viewed as a data driven implementation of the IRKA—the measurements are fed into `mimoVF` to produce an intermediate model, a surrogate of the order  $\tilde{r} \geq r$ , based on measurements, and then this intermediate model is reduced by IRKA to its locally best  $r$ th order approximant.

The  $\mathcal{H}_\infty$  norm is another commonly used system norm. For a stable dynamical system with transfer function  $\mathbf{H}(s)$ , the  $\mathcal{H}_\infty$  norm is defined as  $\|\mathbf{H}\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \|\mathbf{H}(i\omega)\|_2$ . For details, we refer the reader to [59]. The commonly used approach to model reduction towards obtaining a small  $\mathcal{H}_\infty$  error measure is Balanced Truncation (BT) [49, 51]. Even though BT requires solving two Lyapunov equations, once again particular state space realization in (5.1) allows straightforward solution of the Lyapunov equations and makes the BT related computations cheap. Thus, we may also employ BT in reduction to true McMillan degree- $r$  without much additional computational cost.

**5.3. An aggregate procedure: `mimoFIT`.** Our overall approach to adapting VF to matrix-valued rational approximation consists first of the `mimoVF` process (described in detail in sections 2 and 3), followed by a postprocessing step that performs the reduction to true McMillan degree- $r$  with minimal loss of fidelity. This postprocessing stage can be performed either by the ALS correction of section 5.1 or by systems-theoretic approaches such as IRKA or BT as described in section 5.2. We will refer to this two-step process as `mimoFIT`.

**Numerical examples.** Here, we illustrate the performance of `mimoFIT` with four numerical examples. In each case, we investigate the effect of the methodology employed in the postprocessing stage on the overall approximation quality. We also compare the final models produced by `mimoFIT` with the optimal- $\mathcal{H}_2$  approximations obtained by IRKA.

**5.3.1. Heat model.** We consider the Heat Model from the NICONET Benchmark collection [14]; the model has  $m = 2$  inputs and  $p = 2$  outputs. We use only 20 function evaluations ( $\ell = 40$  samples due to complex conjugacy) and obtain rational approximations of order  $r = 6$  and  $r = 10$ . Table 2 lists the resulting relative  $\mathcal{H}_2$  errors due to different approaches. The first row is the error due to the output of `mimoVF`. Note that this approximation has order  $r \times m = 2r$  since it has full-rank residues. This is not our final approximation and is included here as a reference point. We obtain a true degree- $r$  approximant using four different approaches: (i) simple rank-one truncation of the residues by SVD, (ii) ALS correction of section 5.1, (iii) IRKA on the degree- $2r$  output of `mimoVF` to reduce it to degree- $r$ , (iv) BT on the degree- $2r$  output of `mimoVF` to reduce it to degree- $r$ . These four methods are labeled, respectively, as `mimoFIT-(Trnct)`, `mimoFIT-(ALS)`, `mimoFIT-(IRKA)`, and `mimoFIT-(BT)`. The first

<sup>7</sup>It is not required that the underlying transfer function is rational.

TABLE 2

The relative  $\mathcal{H}_2$  errors due to *mimoVF*, *mimoFIT*, and *IRKA*. Twenty function evaluations.

| Method                                  | $r = 6$                 | $r = 10$                |
|---|-------------------------|-------------------------|
| <b>mimoVF (degree: <math>2r</math>)</b> | $1.6530 \times 10^{-2}$ | $1.0759 \times 10^{-3}$ |
| mimoFIT-(Trnct)                         | $3.7022 \times 10^{-2}$ | $2.6137 \times 10^{-2}$ |
| mimoFIT-(ALS)                           | $1.8218 \times 10^{-2}$ | $2.8774 \times 10^{-3}$ |
| mimoFIT-(IRKA)                          | $1.7359 \times 10^{-2}$ | $1.1686 \times 10^{-3}$ |
| mimoFIT-(BT)                            | $3.0604 \times 10^{-2}$ | $1.1591 \times 10^{-3}$ |
| IRKA                                    | $8.5566 \times 10^{-3}$ | $1.0925 \times 10^{-3}$ |

TABLE 3

The relative  $\mathcal{H}_2$  errors due to *mimoVF*, *mimoFIT*, and *IRKA*. One hundred function evaluations.

| Method                                  | $r = 20$                | $r = 30$                |
|---|-------------------------|-------------------------|
| <b>mimoVF (degree: <math>3r</math>)</b> | $4.6074 \times 10^{-2}$ | $3.3226 \times 10^{-2}$ |
| mimoFIT-(Trnct)                         | $1.2904 \times 10^{-1}$ | $1.2508 \times 10^{-1}$ |
| mimoFIT-(ALS)                           | $1.2558 \times 10^{-1}$ | $1.2223 \times 10^{-1}$ |
| mimoFIT-(IRKA)                          | $7.7305 \times 10^{-2}$ | $4.4757 \times 10^{-2}$ |
| mimoFIT-(BT)                            | $7.7457 \times 10^{-2}$ | $3.3483 \times 10^{-2}$ |
| IRKA                                    | $6.7779 \times 10^{-2}$ | $1.1423 \times 10^{-2}$ |

observation is that the simple rank-one truncation of the residues by SVD leads to a high loss of accuracy compared to the ALS correction; this is most apparent in the  $r = 10$  case where *mimoFIT*-(Trnct) has one order of magnitude higher error than *mimoFIT*-(ALS). For both cases, *mimoFIT*-(IRKA) and *mimoFIT*-(BT) perform extremely well (especially *mimoFIT*-(IRKA)) and even with a true degree- $r$  approximant, they almost match the accuracy of the degree- $2r$  *mimoVF* approximant; i.e., reduction from  $2r$  to  $r$  causes a negligible loss of accuracy. The last row indicates the relative  $\mathcal{H}_2$  error associated with the optimal approximant from IRKA. As expected, IRKA yields smaller error; we do not anticipate beating the continuous optimal approximation via a discretized least-square measure. However, it is important to note that *mimoFIT*-(IRKA) and *mimoFIT*-(BT) with only 20 function evaluation yield results close to those obtained by IRKA; this is especially true for  $r = 10$ .

**5.3.2. ISS-1R module.** We repeat the above studies for the ISS 1R module [33] with  $m = 3$  inputs and  $p = 3$  outputs. We use 100 function evaluations ( $\ell = 200$  samples) and obtain rational approximations of order  $r = 20$  and  $r = 30$ . Table 3 depicts the resulting relative  $\mathcal{H}_2$  error values for the same methods used in the previous example in section 5.3.1. As in the previous example, both *mimoFIT*-(IRKA) and *mimoFIT*-(BT) yield very accurate results and show negligible loss of accuracy in reduction from the intermediate  $3r$  approximant to the final degree- $r$  approximant; the order of the *mimoVF* approximant is reduced three-fold yet not much accuracy is lost. *mimoFIT*-(IRKA) and *mimoFIT*-(BT) again yield approximation errors close to that of IRKA. The main difference from the previous case is that in this case even the ALS correction suffers from the loss of accuracy as the simple truncation approach.

**5.3.3. ISS-12A module.** We now investigate the larger ISS 12A module [33] with  $m = 3$  inputs and  $p = 3$  outputs. We focus on this problem since the underlying system of degree  $n = 1412$  is very hard to approximate with a lower order system; it presents significant challenges to model reduction, not necessarily from a computation perspective but from an approximation quality perspective. As illustrated in [33], the Hankel singular values decay rather slowly, so to obtain a reduced model with

TABLE 4

The relative  $\mathcal{H}_2$  errors due to *mimoVF*, *mimoFIT*, and *IRKA*. Two hundred and fifty function evaluations.

| Method                        | $r = 80$                |
|-------------------------------|-------------------------|
| <i>mimoVF</i> (degree: $3r$ ) | $1.4678 \times 10^{-1}$ |
| <i>mimoFIT</i> -(Trnct)       | $2.4549 \times 10^{-1}$ |
| <i>mimoFIT</i> -(ALS)         | $2.2075 \times 10^{-1}$ |
| <i>mimoFIT</i> -(IRKA)        | $1.5116 \times 10^{-1}$ |
| <i>mimoFIT</i> -(BT)          | $1.5130 \times 10^{-1}$ |
| IRKA                          | $1.1317 \times 10^{-1}$ |

a relative error tolerance of  $10^{-3}$ , one needs a reduced model of order at least 226 even using balanced truncation. We use 250 function evaluations, obtain rational approximations of order  $r = 80$  using *mimoFIT*, and compare the result with the continuous optimal approximation *IRKA*. Table 4 depicts the resulting relative  $\mathcal{H}_2$  error values. As before, both *mimoFIT*-(IRKA) and *mimoFIT*-(BT) show negligible loss of accuracy in reduction from the intermediate  $3r$  approximant to the final degree- $r$  approximant and have approximation errors close to that of *IRKA*. *mimoFIT*-(Trnct) and *mimoFIT*-(ALS) perform reasonably well as well in this case with *mimoFIT*-(Trnct) having the largest error among the four.

**5.3.4. A power system example with large input/output space.** This example results from small-signal stability studies for large power systems. The specific model we consider here is one of the models from the Brazilian Interconnect Power System (BIPS); refer to [47]<sup>8</sup> for details.

The underlying dynamical system has dimension  $n = 13250$  with  $m = 46$  inputs and  $p = 46$  outputs. We apply *mimoFIT*-(BT) to obtain our rational approximant. We use  $\ell = 200$  frequency samples. The *mimoVF* (first step of *mimoFIT*) is applied with  $r = 40$ . Due to  $m = p = 46$ , the output of *mimoVF* has an effective McMillan degree of  $r \times m = 1840$ . Note that this is only marginally a reduced model, having a McMillan degree roughly 14% of the originally system order. The decay of the leading Hankel singular values of the intermediate model is shown in the upper plot of Figure 8. Note the slow decay; even after the 300th one, the normalized Hankel singular values are still above the threshold of  $10^{-4}$ . This system is difficult to reduce. Indeed, in earlier works, even for simpler versions of the model having a smaller number of input and outputs (such as  $m = p = 28$ ), a reduced model of degree 291 was used; see [53]. We choose the final degree to be a point at which the normalized Hankel singular values have decayed below  $2 \times 10^{-4}$ , leading to a final McMillan degree of 253 (around 2% of the original). The sigma plots, i.e.,  $\|\mathbf{H}(i\omega)\|_2$  versus  $\omega \in \mathbb{R}$ , for the full model and the final *mimoFIT* approximant are shown in the lower plot of Figure 8. As the figure illustrates, the *mimoFIT* approximant does an excellent job in approximating the underlying dynamics.

**6. Conclusions.** The VF method, as originated by Gustavsen and Semlyen, has been and continues to be an important tool for rational approximation and many authors have applied, modified, and analyzed this approach. Although their method is based on a successive solution of linear LS problems, which are well understood problems in of themselves, we find that subtleties enter that can degrade both the

<sup>8</sup>This model can be downloaded from <https://sites.google.com/site/rommes/software>.

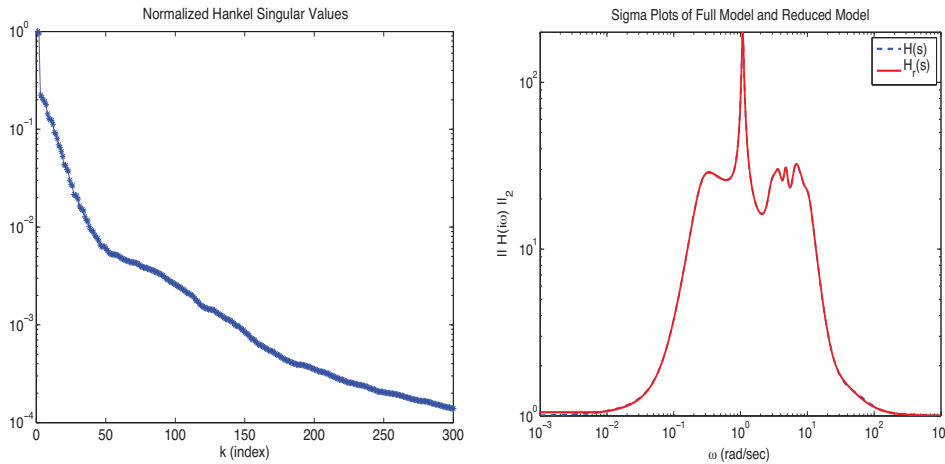


FIG. 8. (Example 5.3.4.) Large-scale power system example. Left plot: Decay of the Hankel singular values. Right plot: The sigma plots of the full and reduced model.

performance and accuracy of current VF approaches, especially for matrix-valued rational approximation of modest dimension which often produce extremely poorly conditioned problems. These issues include

- (1) balancing the potential conflict between rank revealing pivoting in QR factorizations used in LS solvers and column-scaling used to improve conditioning;
- (2) avoiding redundant computation within the multiple subproblems solved as part of the large LS problem that arises;
- (3) the need for rigorous termination criteria and the efficient recovery of the best possible rational approximant in case the iteration must be terminated prematurely;
- (4) the use of regularized least squares and the discrepancy principle, both implemented using high accuracy linear algebra methods; and
- (5) control of the McMillan degree of the resulting rational approximant.

In this paper, we have considered these issues carefully, together with other more minor ones. We have integrated these developments into a robust, efficient implementation of VF for matrix-valued rational approximation, called *mimoVF*, that appears to be both faster and more accurate than currently available implementations. Further, we have connected the underlying discrete LS approximation problem to a continuous optimal  $\mathcal{H}_2$  approximation problem through numerical quadrature, which motivated a reformulation of the original VF objective as a weighted LS problem. For essentially the same cost as *mimoVF*, we were then able to use this reformulation to significantly improve the quality of the approximation. Finally, we have offered here an aggregate procedure, called *mimoFIT*, that combines *mimoVF* with  $\mathcal{H}_2/\mathcal{H}_\infty$ -based approximation methods that yield high-fidelity rational approximants with low McMillan degree.

#### REFERENCES

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, L. S. BLACKFORD, J. DEMMEL, J. J. DONGARRA, J. DU CROZ, S. HAMMARLING, A. GREENBAUM, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.
- [2] A. C. ANTOUNAS, *Approximation of Large-Scale Dynamical Systems*, *Adv. Des. Control* 6, SIAM, Philadelphia, 2005.

- [3] A. C. ANTOULAS, *Model reduction of nonlinear systems in the Loewner framework*, in Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems, Groningen, The Netherlands, 2014.
- [4] A. C. ANTOULAS, *Data-driven model reduction for weakly nonlinear systems: A summary*, in Proceedings of the 8th Vienna International Conference on Mathematical Modeling, Vienna, Austria, 2015.
- [5] A. C. ANTOULAS, C. A. BEATTIE, AND S. GUGERCIN, *Interpolatory Model Reduction of Large-Scale Dynamical Systems*, in Efficient Modeling and Control of Large-Scale Systems, J. Mohammadpour and K. Grigoriadis, eds., Springer-Verlag, New York, 2010, pp. 3–58.
- [6] A. C. ANTOULAS, A. C. IONITA, AND S. LEFTERIU, *On two-variable rational interpolation*, Linear Algebra Appl., 436 (2012), pp. 2889–2915.
- [7] C. BEATTIE AND S. GUGERCIN, *Realization-independent  $H_2$  approximation*, in Proceedings of the 51st IEEE Conference on Decision & Control, IEEE, Piscataway, NJ, 2012, pp. 4953–4958.
- [8] P. BENNER, S. GUGERCIN, AND K. WILLCOX, *A Survey of Model Reduction Methods for Parametric Systems*, Technical report MPIMD/13-14, Max Planck Institute, Magdeburg, Austria, 2013.
- [9] A. R. BENSON, D. F. GLEICH, AND J. DEMMEL, *Direct QR factorizations for tall-and-skinny matrices in mapreduce architectures*, CoRR, abs/1301.1071, 2013.
- [10] M. BERLJAJA AND S. GÜTTEL, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 894–916.
- [11] J. P. BOYD, *Exponentially convergent Fourier-Chebyshev quadrature schemes on bounded and infinite intervals*, J. Sci. Comput., 2 (1987), pp. 99–109.
- [12] P. A. BUSINGER AND G. H. GOLUB, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.
- [13] N. CASTRO-GONZÁLEZ, J. CEBALLOS, F. M. DOPICO, AND J. M. MOLERA, *Accurate solution of structured least squares problems via rank-revealing decompositions*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1112–1128.
- [14] Y. CHAHLAOUI AND P. VAN DOOREN, *A Collection of Benchmark Examples for Model Reduction of Linear Time Invariant Dynamical Systems*, Technical report, SLICOT Working Note 2002-2, 2002.
- [15] A. CHINEA AND S. GRIVET-TALOCIA, *On the parallelization of vector fitting algorithms*, IEEE Trans. Compon., Packag. Manuf. Technol., 1 (2011), pp. 1761–1773.
- [16] A. J. COX AND N. J. HIGHAM, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., Pitman Res. Notes Math. Ser. 380, Longman, Harlow, UK, 1998, pp. 57–73.
- [17] J. DEMMEL, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 562–580.
- [18] J. DEMMEL, L. GRIGORI, M. GU, AND H. XIANG, *Communication Avoiding Rank Revealing QR Factorization with Column Pivoting*, Technical report UCB/EECS-2013-46, EECS Department, University of California, Berkeley, CA, 2013.
- [19] J. DEMMEL, M. GU, S. EISENSTAT, I. SLAPNIČAR, K. VESELIĆ, AND Z. DRMAČ, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.
- [20] D. DESCHRIJVER, B. GUSTAVSEN, AND T. DHAENE, *Advancements in iterative methods for rational approximation in the frequency domain*, IEEE Trans. Power Del., 22 (2007), pp. 1633–1642.
- [21] D. DESCHRIJVER, B. HAEGEMAN, AND T. DHAENE, *Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses*, IEEE Trans. Adv. Packag., 30 (2007), pp. 216–225.
- [22] D. DESCHRIJVER, L. KNOCKAERT, AND T. DHAENE, *Improving robustness of vector fitting to outliers in data*, Electronics Letters, 46 (2010), pp. 1–2.
- [23] D. DESCHRIJVER, M. MROZOWSKI, T. DHAENE, AND D. DE ZUTTER, *Macromodeling of multiport systems using a fast implementation of the vector fitting method*, IEEE Microw. Compon. Lett., 18 (2008), pp. 383–385.
- [24] D. DESCHRIJVER, M. MROZOWSKI, T. DHAENE, AND D. DE ZUTTER, *A barycentric vector fitting algorithm for efficient macromodeling of linear multiport systems*, IEEE Microw. Compon. Lett., 23 (2013), pp. 60–62.



- [25] Z. DRMAČ, *On principal angles between subspaces of Euclidean space*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 173–194.
- [26] Z. DRMAČ AND Z. BUJANOVIĆ, *On the failure of rank revealing QR factorization software—a case study*, ACM Trans. Math. Software, 35 (2009), article 12.
- [27] Z. DRMAČ, S. GUGERCIN, AND C. A. BEATTIE, *Quadrature-based vector fitting for discretized  $\mathcal{H}_2$  approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A625–A652.
- [28] Z. DRMAČ AND K. VESELIĆ, *New fast and accurate Jacobi SVD algorithm: I*, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1322–1342.
- [29] Z. DRMAČ AND K. VESELIĆ, *New fast and accurate Jacobi SVD algorithm: II*, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1343–1362.
- [30] G. GOLUB, V. KLEMA, AND S. C. PETERS, *Rules and software for detecting rank degeneracy*, J. Econometrics, 12 (1980), pp. 41–48.
- [31] G. H. GOLUB, V. C. KLEMA, AND G. W. STEWART, *Rank Degeneracy and Least Squares Problems*, Technical report, Stanford University, Stanford, CA, 1976.
- [32] P. GONNET, R. PACHÓN, AND L. N. TREFETHEN, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167.
- [33] S. GUGERCIN, A. C. ANTOULAS, AND N. BEDROSSIAN, *Approximation of the international space station 1r and 12a models*, in Proceedings of the 40th IEEE Conference on Decision and Control, 2001, Vol. 2, IEEE, Piscataway, NJ, 2001, pp. 1515–1516.
- [34] S. GUGERCIN, A. C. ANTOULAS, AND C. BEATTIE,  *$\mathcal{H}_2$  model reduction for large-scale linear dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638.
- [35] B. GUSTAVSEN, *Comments on “a comparative study of vector fitting and orthonormal vector fitting techniques for EMC applications,”* in Proceedings of the 18th International Zurich Symposium on Electromagnetic Compatibility, IEEE, Piscataway, NJ, 2006, pp. 131–134.
- [36] B. GUSTAVSEN, *Improving the pole relocating properties of vector fitting*, IEEE Trans. Power Delivery, 21 (2006), pp. 1587–1592.
- [37] B. GUSTAVSEN, *User’s Manual for VECTFIT*, SINTEF Energy Research, October 27, 2005.
- [38] B. GUSTAVSEN AND A. SEMLYEN, *Rational approximation of frequency domain responses by vector fitting*, IEEE Trans. Power Delivery, 14 (1999), pp. 1052–1061.
- [39] B. GUSTAVSEN AND A. SEMLYEN, *A robust approach for system identification in the frequency domain*, IEEE Trans. Power Delivery, 19 (2004), pp. 1167–1173.
- [40] W. HENDRICKX AND T. DHAENE, *A discussion of “Rational approximation of frequency domain responses by vector fitting,”* IEEE Trans. Power Systems, 21 (2006), pp. 441–443.
- [41] N. J. HIGHAM, *QR factorization with complete pivoting and accurate computation of the SVD*, Linear Algebra Appl., 309 (2000), pp. 153–174.
- [42] J. HOKANSON, *Numerically Stable and Statistically Efficient Algorithms for Large Scale Exponential Fitting*, Ph.D. thesis, Doctoral Thesis, Rice University, Houston, TX, <http://hdl.handle.net/1911/77161>, 2013.
- [43] A. C. IONITA AND A. C. ANTOULAS, *Data-driven parametrized model reduction in the Loewner framework*, SIAM J. Sci. Comput., 36 (2014), pp. A984–A1007.
- [44] R. E. KALMAN, *Design of a self-optimizing control system*, Trans. ASME, 80 (1958), pp. 468–478.
- [45] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, SIAM, Philadelphia, 1974.
- [46] S. LEFTERIU AND A. C. ANTOULAS, *A new approach to modeling multiport systems from frequency-domain data*, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, 29 (2010), pp. 14–27.
- [47] N. MARTINS, P. C. PELLANDA, AND J. ROMMES, *Computation of transfer function dominant zeros with applications to oscillation damping control of large power systems*, IEEE Trans. Power Syst., 22 (2007), pp. 1657–1664.
- [48] A. J. MAYO AND A. C. ANTOULAS, *A framework for the solution of the generalized realization problem*, Linear Algebra Appl., 425 (2007), pp. 634–662.
- [49] B. MOORE, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Transactions on Automatic Control, 26 (1981), pp. 17–32.
- [50] V. MOROZOV, *On the solution of functional equations by the method of regularization*, Soviet Mathematics Doklady, 7 (1966), pp. 414–417.
- [51] C. T. MULLIS AND R. A. ROBERTS, *Synthesis of minimum roundoff noise fixed point digital filters*, IEEE Trans. Circuits Syst., 23 (1976), pp. 551–562.
- [52] M. J. D. POWELL AND J. K. REID, *On applying householder transformations to linear least squares problems*, in Information Processing 68, Proceedings of the International Federation of Information Processing Congress, Edinburgh, 1968, North-Holland, Amsterdam, 1969, pp. 122–126.

- [53] J. ROMMES AND N. MARTINS, *Efficient computation of multivariable transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Systems, 21 (2006), pp. 1471–1483.
- [54] C. SANATHANAN AND J. KOERNER, *Transfer function synthesis as a ratio of two complex polynomials*, IEEE Trans. Automat. Control, 8 (1963), pp. 56–58.
- [55] W. H. A. SCHILDERS, H. A. VAN DER VORST, AND J. ROMMES, *Model Order Reduction: Theory, Research Aspects and Applications*, Vol. 13, Springer-Verlag, Berlin-Heidelberg, 2008.
- [56] SINTEF, *VFIT3*, <http://www.sintef.no/Projectweb/VECTFIT/Downloads/VFUT3/>.
- [57] A. V. D. SLUIS, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
- [58] G. W. STEWART, *Determining rank in the presence of error*, in Linear Algebra for Large Scale and Real-Time Applications (Proceedings of the NATO Advanced Study Institute, Leuven, Belgium, 1992), M. S. Moonen, G. H. Golub, and B. L. de Moor, eds., Kluwer Academic Publishers, Dordrecht, 1993, pp. 275–291.
- [59] K. ZHOU, J. DOYLE, AND K. GLOVER, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1996.