Least squares approximations in system identification and model reduction
I. Vector Fitting for Matrix-Valued Rational Approximations

Zlatko Drmač
Based on joint work with Chris Beattie and Serkan Gugercin

Faculty of Science, Department of Mathematics
University of Zagreb
Supported by HRZZ 9345

drmac@math.hr

November 1, 2017
Groningen Autumn School on Model Order Reduction

Overview

1. Introduction
   - Problem statement
   - Sanathanan–Koerner iterations
   - VF of Gustavsen and Semlyen
2. Numerical analysis and modifications of VF
   - Local pivoting in the global QR factorization
   - The algorithm: global description
   - Stopping criterion
   - Cancellations, column scalings, numerical rank and LS solver
3. Controlling the McMillan degree
   - Rank one truncation + ALS correction
   - Order reduction in $\mathcal{H}_2$, $\mathcal{H}_\infty$ setting
4. Tangential rational LS approximation of matrix functions

Introduction

Suppose the dynamics of an n–dimensional LTI stable MIMO system

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad \rightleftarrows \mathbf{H}(s) = C(sI - \mathbf{A})^{-1}B \in \mathbb{C}^{p\times m}$$

with transfer function $\mathbf{H}(s)$ may be inaccessible to direct modeling, yet input-output relationships may be observed as a function of frequency, yielding an enormous amount of data.

- Challenge: Deduce an empirical dynamical system model represented as a matrix-valued rational function, that fits the measured frequency response data $\mathbf{H}(\xi_i) \in \mathbb{C}^{p\times m}$. Use the derived model as a surrogate in order to predict system behavior or to determine suitable control.
- The McMillan degree r of the approximant $\mathbf{H}_r$ is given or is to be determined. Even if $\mathbf{H}$ is known, we may want to replace it with $\mathbf{H}_r$.
- For given frequency responses $\mathbf{H}(\xi_i)$ and weighting factors ρ_i, the error is measured by $\sum_{i=1}^{\ell} \rho_i \|\mathbf{H}_r(\xi_i) - \mathbf{H}(\xi_i)\|_F^2$. $(\xi_i = \boldsymbol{i}\omega_i,\ \omega_i \in \mathbb{R})$

Basic Iterations (Sanathanan–Koerner, Kalman)

Problem: Minimize over $\mathcal{R}_r$ (nonlinear, non-convex optimization)

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

$\mathcal{R}_r$ consists of $p \times m$ matrix valued functions with entries that are strictly proper rational functions of McMillan degree r.

Sanathanan–Koerner (SK) Iterations

Compute a sequence of $\mathbf{H}_r^{(k)}(s) = \mathbf{N}^{(k)}(s)/d^{(k)}(s)$, 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. Iterate for $k = 0, 1, 2, \ldots$

$$\epsilon^{(k)} = \sum_{i=1}^{\ell} \frac{\rho_i}{|d^{(k)}(\xi_i)|^2} \left\| \mathbf{N}^{(k+1)}(\xi_i) - d^{(k+1)}(\xi_i)\mathbf{H}(\xi_i) \right\|_F^2 \longrightarrow \min .$$

Polynomial representation (Example: SISO, $m = p = 1$)

$$\mathbf{H}_r(s) = \frac{n(s)}{d(s)} \text{ with } n(s) = \sum_{j=0}^{r-1} \alpha_j s^j \text{ and } d(s) = 1 + \sum_{j=1}^{r} \beta_j s^j,$$

Starting with $d^{(0)}(s) \equiv 1$, solve successively for $k = 0, 1, 2, \ldots$

$$\sum_{i=1}^{\ell} \rho_i \left| \frac{n^{(k+1)}(\xi_i) - d^{(k+1)}(\xi_i)\mathbf{H}(\xi_i)}{d^{(k)}(\xi_i)} \right|^2 \equiv \|\Delta^{(k)}(\mathcal{B}y^{(k+1)} - h)\|_2 \to \min$$

$$y^{(k+1)} = \begin{pmatrix} \alpha_0^{(k+1)} & \alpha_1^{(k+1)} & \ldots & \alpha_{r-1}^{(k+1)} & \beta_1^{(k+1)} & \beta_2^{(k+1)} & \ldots & \beta_r^{(k+1)} \end{pmatrix}^T, \Delta^{(k)} = \operatorname{diag}\left(\frac{\rho_i}{|d^{(k)}(\xi_i)|} \right)$$

$$\mathcal{B} = \begin{pmatrix} 1 & \xi_1 & {\xi_1}^2 & \ldots & {\xi_1}^{r-1} & -\mathbf{H}(\xi_1)\xi_1 & -\mathbf{H}(\xi_1){\xi_1}^2 & \ldots & -\mathbf{H}(\xi_1){\xi_1}^r \\ 1 & \xi_2 & \xi_2^2 & \ldots & \xi_2^{r-1} & -\mathbf{H}(\xi_2)\xi_2 & -\mathbf{H}(\xi_2)\xi_2^2 & \ldots & -\mathbf{H}(\xi_2)\xi_2^r \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_{\ell-1} & \xi_{\ell-1}^2 & \ldots & \xi_{\ell-1}^{r-1} & -\mathbf{H}(\xi_{\ell-1})\xi_{\ell-1} & -\mathbf{H}(\xi_{\ell-1})\xi_{\ell-1}^2 & \ldots & -\mathbf{H}(\xi_{\ell-1})\xi_{\ell-1}^r \\ 1 & \xi_\ell & \xi_\ell^2 & \ldots & \xi_\ell^{r-1} & -\mathbf{H}(\xi_\ell)\xi_\ell & -\mathbf{H}(\xi_\ell)\xi_\ell^2 & \ldots & -\mathbf{H}(\xi_\ell)\xi_\ell^r \end{pmatrix}, h = \begin{pmatrix} \mathbf{H}(\xi_1) \\ \mathbf{H}(\xi_2) \\ \vdots \\ \mathbf{H}(\xi_{\ell-1}) \\ \mathbf{H}(\xi_\ell) \end{pmatrix} \quad (1.1)$$

Problems with SK iterations in polynomial representation

Numerical problems

1. Polynomial representation induces ill-conditioned Vandermonde matrices, e.g. $\kappa_2(100 \times 100 \text{ real Vandermonde}) > 3 \cdot 10^{28}$.
2. Scaling by $1/|d^{(k)}(\xi_i)|^2$ may worsen the condition number.
3. High powers ξ_i^j in Vandermonde matrices may cause overflows; re-scaling may be required.
4. Stability of $\mathbf{H}_r^{(k)}$ must me monitored and, if necessary, enforced.
5. McMillan degree of $\mathbf{H}_r^{(k)}$ can be as high as $r \cdot \min(p, m)$, and order r is desired.
6. Pole residue representation $\mathbf{H}_r(s) = \sum_{j=1}^{r} \frac{\mathbf{\Phi}_j}{s - \lambda_j}$ may be preferable for further usage.

Convergence of this kind of iterations in general remains open challenge.

Change of basis: Barycentric representation of $H_r(s)$

Pick an arbitrary set of mutually distinct scalars $\lambda_0, \lambda_1, \ldots, \lambda_r$ ("interpolation points") and define the nodal polynomial $\omega_r(s) = \prod_{k=1}^{r}(s - \lambda_k)$ (notice λ_0 is excluded). Then,

$$n(s) = \omega_r(s) \sum_{j=1}^{r} \frac{w_j\, n(\lambda_j)}{s - \lambda_j} \quad \text{and} \quad d(s) = \omega_r(s) \left(\alpha + \sum_{j=1}^{r} \frac{w_j\, d(\lambda_j)}{s - \lambda_j} \right),$$

where $w_j = 1/\prod_{k\neq j}(\lambda_j - \lambda_k)$ enforces interpolation of $n(s)$, and hence $H_r(s)$, at $s = \lambda_j$ for $j = 1, \ldots, r$ and choosing $\alpha = \frac{d(\lambda_0)}{\omega_r(\lambda_0)} - \sum_{j=1}^{r} \frac{d(\lambda_j) w_j}{\lambda_0 - \lambda_j}$ then enforces interpolation of H_r also at $s = \lambda_0$. Since $\deg(d) = r$, $\alpha \neq 0$.

Coefficients in barycentric representation: $\phi_j = \frac{w_j}{\alpha} n(\lambda_j)$, $\varphi_j = \frac{w_j}{\alpha} d(\lambda_j)$

$$H_r(s) = \frac{\sum_{j=1}^{r} \frac{\phi_j}{s-\lambda_j}}{1 + \sum_{j=1}^{r} \frac{\varphi_j}{s-\lambda_j}} = \frac{\tilde{n}(s)}{\tilde{d}(s)} \qquad \text{with} \quad \begin{cases} \tilde{n}(s) = \sum_{j=1}^{r} \frac{\phi_j}{s-\lambda_j}, \text{ and} \\ \tilde{d}(s) = 1 + \sum_{j=1}^{r} \frac{\varphi_j}{s-\lambda_j}\,. \end{cases}$$

SK iterations in barycentric form

We may now use ϕ_j, φ_j as optimization parameters in each step of the SK iteration

$$\sum_{i=1}^{\ell} \rho_i \left| \frac{n^{(k+1)}(\xi_i) - d^{(k+1)}(\xi_i)\mathbf{H}(\xi_i)}{d^{(k)}(\xi_i)} \right|^2$$

Indeed, for a given set of interpolation points, $\lambda_1, \ldots, \lambda_r$, observation points $\xi_1, \ldots, \xi_\ell$, and system observations $\mathbf{H}(\xi_1), \ldots, \mathbf{H}(\xi_\ell)$, the parameters $\phi_j^{(k)}, \varphi_j^{(k)}$ describe $\mathbf{H}_r^{(k)}(s) = \frac{\tilde{n}^{(k)}(s)}{\tilde{d}^{(k)}(s)}$ in the kth step, replacing $n^{(k)}$ and $d^{(k)}$ with

$$\tilde{n}^{(k)}(s) = \sum_{j=1}^{r} \frac{\phi_j^{(k)}}{s - \lambda_j} \quad \text{and} \quad \tilde{d}^{(k)}(s) = 1 + \sum_{j=1}^{r} \frac{\varphi_j^{(k)}}{s - \lambda_j}, \tag{1.2}$$

respectively.

SK iterations in barycentric form

Now, $\phi_j^{(k)}$, $\varphi_j^{(k)}$ are determined by solution of the successive LS problems

$$\|\Delta^{(k)}(\mathcal{A}x^{(k+1)} - h)\|_2 \to \min, \quad k = 0, 1, 2, \ldots, \tag{1.3}$$

where $x^{(k+1)} = \begin{pmatrix} \phi_1^{(k+1)} & \phi_2^{(k+1)} & \cdots & \phi_r^{(k+1)} & \varphi_1^{(k+1)} & \varphi_2^{(k+1)} & \cdots & \varphi_r^{(k+1)} \end{pmatrix}^T$ and

$$\mathcal{A} = \begin{pmatrix} \frac{1}{\xi_1-\lambda_1} & \frac{1}{\xi_1-\lambda_2} & \cdots & \frac{1}{\xi_1-\lambda_r} & \frac{-H(\xi_1)}{\xi_1-\lambda_1} & \frac{-H(\xi_1)}{\xi_1-\lambda_2} & \cdots & \frac{-H(\xi_1)}{\xi_1-\lambda_r} \\ \frac{1}{\xi_2-\lambda_1} & \frac{1}{\xi_2-\lambda_2} & \cdots & \frac{1}{\xi_2-\lambda_r} & \frac{-H(\xi_2)}{\xi_2-\lambda_1} & \frac{-H(\xi_2)}{\xi_2-\lambda_2} & \cdots & \frac{-H(\xi_2)}{\xi_2-\lambda_r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\xi_{\ell-1}-\lambda_1} & \frac{1}{\xi_{\ell-1}-\lambda_2} & \cdots & \frac{1}{\xi_{\ell-1}-\lambda_r} & \frac{-H(\xi_{\ell-1})}{\xi_{\ell-1}-\lambda_1} & \frac{-H(\xi_{\ell-1})}{\xi_{\ell-1}-\lambda_2} & \cdots & \frac{-H(\xi_{\ell-1})}{\xi_{\ell-1}-\lambda_r} \\ \frac{1}{\xi_\ell-\lambda_1} & \frac{1}{\xi_\ell-\lambda_2} & \cdots & \frac{1}{\xi_\ell-\lambda_r} & \frac{-H(\xi_\ell)}{\xi_\ell-\lambda_1} & \frac{-H(\xi_\ell)}{\xi_\ell-\lambda_2} & \cdots & \frac{-H(\xi_\ell)}{\xi_\ell-\lambda_r} \end{pmatrix}. \tag{1.4}$$

h is as in (1.1), $\Delta^{(k)} = \operatorname{diag}(\rho_i/|\tilde{d}^{(k)}(\xi_i)|)_{i=1}^{\ell}$.
From now on, use barycentric form,

$$\boxed{\tilde{\mathbf{d}}^{(\mathbf{k})}(\mathbf{s}) \equiv \mathbf{d}^{(\mathbf{k})}(\mathbf{s}), \quad \tilde{\mathbf{n}}^{(\mathbf{k})}(\mathbf{s}) \equiv \mathbf{n}^{(\mathbf{k})}(\mathbf{s}).}$$

And the nodes λ_j? Use $\lambda_j^{(k)}$

Suppose now that the interpolation points depend on k and denote them by $\lambda_j^{(k)}$; we define $\mathcal{A}^{(k)} \equiv \mathcal{A}(\boldsymbol{\lambda}^{(k)})$ to be $\mathcal{A}$ as defined in (1.4), but with λ_j replaced by $\lambda_j^{(k)}$. Let $\lambda_j^{(k+1)}$ to be the zeros of $d^{(k)}(s)$:

$$d^{(k)}(s) = 1 + \sum_{j=1}^{r} \frac{\varphi_j^{(k)}}{s - \lambda_j^{(k)}} = \frac{\prod_{j=1}^{r}(s - \lambda_j^{(k+1)})}{\prod_{j=1}^{r}(s - \lambda_j^{(k)})}. \tag{1.5}$$

Then, the goal of (1.3) becomes the minimization of

$$\sum_{i=1}^{\ell} \frac{\rho_i}{|d^{(k)}(\xi_i)|^2} \left| \sum_{j=1}^{r} \frac{\phi_j^{(k+1)}}{\xi_i - \lambda_j^{(k)}} - H(\xi_i) \left(1 + \sum_{j=1}^{r} \frac{\varphi_j^{(k+1)}}{\xi_i - \lambda_j^{(k)}} \right) \right|^2 \tag{1.6}$$

$$= \sum_{i=1}^{\ell} \rho_i \left| \frac{\prod_{j=1}^{r}(\xi_i - \lambda_j^{(k)})}{\prod_{j=1}^{r}(\xi_i - \lambda_j^{(k+1)})} \right|^2 \left| \frac{\tilde{p}^{(k+1)}(\xi_i)}{\prod_{j=1}^{r}(\xi_i - \lambda_j^{(k)})} - H(\xi_i) \frac{\tilde{q}^{(k+1)}(\xi_i)}{\prod_{j=1}^{r}(\xi_i - \lambda_j^{(k)})} \right|^2$$

where $\tilde{p}^{(k+1)}$ and $\tilde{q}^{(k+1)}$ are, polynomials of degree $r-1$ and r.

$\lambda_j^{(k)} \rightsquigarrow \lambda_j^{(k+1)}$

Continuing with similar algebraic manipulations, one obtains

$$
\begin{aligned}
\dots &= \sum_{i=1}^{\ell} \rho_i \left| \frac{\tilde{p}^{(k+1)}(\xi_i)}{\prod_{j=1}^{r}(\xi_i - \lambda_j^{(k+1)})} - \mathbf{H}(\xi_i) \frac{\tilde{q}^{(k+1)}(\xi_i)}{\prod_{j=1}^{r}(\xi_i - \lambda_j^{(k+1)})} \right|^2 \\
&= \sum_{i=1}^{\ell} \rho_i \left| \sum_{j=1}^{r} \frac{\tilde{\phi}_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}} - \mathbf{H}(\xi_i) \left(1 + \sum_{j=1}^{r} \frac{\tilde{\varphi}_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}} \right) \right|^2 \qquad (1.7) \\
&= \| D_\rho (\mathcal{A}^{(k+1)} \tilde{x}^{(k+1)} - h) \|_2^2, \quad D_\rho = \operatorname{diag}(\sqrt{\rho_i})_{i=1}^{\ell}
\end{aligned}
$$

where $\tilde{x}^{(k+1)} = \begin{pmatrix} \tilde{\phi}_1^{(k+1)} & \tilde{\phi}_2^{(k+1)} & \dots & \tilde{\phi}_r^{(k+1)} & \tilde{\varphi}_1^{(k+1)} & \tilde{\varphi}_2^{(k+1)} & \dots & \tilde{\varphi}_r^{(k+1)} \end{pmatrix}^T$ with $\tilde{\phi}_j^{(k+1)}$ and $\tilde{\varphi}_j^{(k+1)}$ as defined in (1.7) denoting the coefficients of barycentric form of $\mathbf{H}_r(s)$. Thus, one step of VF corresponds to solving the least squares problem with iteration dependent coefficients matrix. Ease the notation an remove tildas, $\tilde{\phi}_j \equiv \phi_j$, $\tilde{\varphi}_j \equiv \varphi_j$.

VF - details - computing the new nodes $\lambda_j^{(k+1)}$

- The $\lambda_j^{(k+1)}$, the zeros of $1+\sum_{j=1}^{r}\varphi_j^{(k)}/(s-\lambda_j^{(k)})$ are computed (recall: $\det(A+uv^T)=(1+v^TA^{-1}u)det(A)$) as the eigenvalues of

$$Z^{(k)}=\begin{pmatrix}\lambda_1^{(k)} & & \\ & \ddots & \\ & & \lambda_r^{(k)}\end{pmatrix}-\begin{pmatrix}\varphi_1^{(k)}\\ \vdots \\ \varphi_r^{(k)}\end{pmatrix}\begin{pmatrix}1 & \dots & 1\end{pmatrix}$$

$$\det(sI-Z^{(k)})=(1+\sum_{i=1}^{r}\frac{\varphi_i^{(k)}}{s-\lambda_i^{(k)}})\prod_{i=1}^{r}(s-\lambda_i^{(k)})$$ S-M for $s\notin\{\lambda_i^{(k)},i\}$

- For real systems, the complex $\lambda_j^{(k)}$'s come in complex conjugate pairs. Same holds for the corresponding $\varphi_j^{(k)}$'s and $\phi_j^{(k)}$'s.
- Ustable $\lambda_j^{(k)}$'s can be immediately flipped across the imaginary axis, thus keeping the $\mathbf{H}_r^{(k)}$'s stable throughout the iterations.
- If the underlying system is real, need LS solution closed under complex conjugation and with the same complex inertia as of the $\lambda_1^{(k)},\dots,\lambda_r^{(k)}$!

Technical detail: closedness under complex conjugation

Let us drop the iteration index, and let the initial set of poles be closed under conjugation, $\boldsymbol{\lambda} = (\lambda_j) \in \mathbb{C}^r$. This can be formally written as $\boldsymbol{\lambda} = \Pi\overline{\boldsymbol{\lambda}}$, where Π is a permutation, and $\overline{\boldsymbol{\lambda}} = (\overline{\lambda_j})$.
Assume that $\boldsymbol{\lambda}$ is ordered so that real values are listed first, and then the complex conjugate pairs. In that case $\boldsymbol{\lambda} = \Pi\overline{\boldsymbol{\lambda}}$ holds with $\Pi = \Pi^T$.
To explicitly express that $\mathcal{A}$ depends on $\boldsymbol{\xi}$ and $\boldsymbol{\lambda}$ we write $\mathcal{A} = \mathcal{A}_{(\boldsymbol{\xi},\boldsymbol{\lambda})}$.
The LS solution vector x is partitioned as $x = \begin{pmatrix} \phi \\ \varphi \end{pmatrix}$, $\phi, \varphi \in \mathbb{C}^r$.

Requirement for real model

In order to preserve real approximant, we require that the LS problem $\|W(\mathcal{A}x - h)\|_2 \rightarrow \min$ is solved under the constraint $x = (\Pi \oplus \Pi)\overline{x}$, i.e. both ϕ and φ must be closed under conjugation.

This constrained problem is turned into an unconstrained LS minimization as follows.

... closedness under complex conjugation

First, augment the array of nodes $\boldsymbol{\xi} = (\xi_i) \in \mathbb{C}^\ell$ by taking their conjugate values $\overline{\xi_i}$, with the corresponding measurements $\overline{h_i}$ and define the matrix $\mathcal{A}_{(\overline{\boldsymbol{\xi}},\boldsymbol{\lambda})} \equiv \overline{\mathcal{A}_{(\boldsymbol{\xi},\boldsymbol{\lambda})}}(\Pi \oplus \Pi)$. Here, of course, introducing negative frequency by taking $\overline{\xi_i} = -\dot{\imath}2\pi f_i$ has no physical meaning, but it is a mathematical device used to implicitly impose certain structure. In fact, the ξ_i's are not necessarily from $\dot{\imath}\mathbb{R}$ – general real and complex values are allowed.
It is easily checked that $\|D_\rho(\mathcal{A}_{(\boldsymbol{\xi},\boldsymbol{\lambda})}x - h)\|_2 = \|D_\rho(\mathcal{A}_{(\overline{\boldsymbol{\xi}},\boldsymbol{\lambda})}x - \overline{h}))\|_2$ and that our constrained LS problem can be equivalently stated as

$$\left\| \begin{pmatrix} D_\rho \mathcal{A}_{(\boldsymbol{\xi},\boldsymbol{\lambda})} \\ D_\rho \mathcal{A}_{(\overline{\boldsymbol{\xi}},\boldsymbol{\lambda})} \end{pmatrix} \begin{pmatrix} \phi \\ \varphi \end{pmatrix} - \begin{pmatrix} D_\rho h \\ D_\rho \overline{h} \end{pmatrix} \right\|_2 \longrightarrow \min, \tag{1.8}$$

with the constraint that $\phi = \Pi\overline{\phi}$, $\varphi = \Pi\overline{\varphi}$. $D_\rho = D_\rho^T \succ 0$ is weighting.

The key observation

Every solution $x = \left(\begin{smallmatrix} \phi \\ \varphi \end{smallmatrix}\right)$ of the augmented unconstrained LS problem (1.8) is closed under conjugation, i.e. it satisfies $\phi = \Pi\overline{\phi}$, $\varphi = \Pi\overline{\varphi}$.

Vector Fitting (Gustavsen–Semlyen; Deschrijver, Dhaene)

MIMO case

Barycentric representation

$$\mathbf{H}_r^{(k)}(s) = \frac{\mathbf{N}^{(k)}(s)}{d^{(k)}(s)} \equiv \frac{\sum_{j=1}^{r} \mathbf{\Phi}_j^{(k)}/(s-\lambda_j^{(k)})}{1+\sum_{j=1}^{r} \varphi_j^{(k)}/(s-\lambda_j^{(k)})}, \quad \begin{array}{l} \mathbf{\Phi}_j^{(k)} \in \mathbb{C}^{p\times m} \\ \varphi_j^{(k)}, \lambda_j^{(k)} \in \mathbb{C} \end{array}$$

SK iterations in barycentric form

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

Recall : $$\epsilon^{(k)} = \sum_{i=1}^{\ell} \frac{\rho_i}{|d^{(k)}(\xi_i)|^2} \left\| \mathbf{N}^{(k+1)}(\xi_i) - d^{(k+1)}(\xi_i)\mathbf{H}(\xi_i) \right\|_F^2$$

Vector Fitting (Gustavsen–Semlyen)

Vector Fitting: great idea

Let $d^{(k)}(s) \equiv 1 + \sum_{j=1}^{r} \varphi_j^{(k)}/(s-\lambda_j^{(k)}) = \prod_{j=1}^{r}(s-\lambda_j^{(k+1)})/\prod_{=1}^{r}(s-\lambda_j^{(k)})$.

Then seek the next iterate in barycentric form using the nodes $\lambda_j^{(k+1)}$.

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

Compare with

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

VF details - computing the new residues $\varphi_j^{(k)}$ and $\mathbf{\Phi}_j^{(k)}$

Data structure

Samples $\mathbb{S}(:,:,i) = S^{(i)} = \mathbf{H}(\xi_i) + \mathcal{E}_i$, $i = 1, \ldots, \ell$, $\mathbb{S} \in \mathbb{C}^{p\times m\times \ell}$.
Residues $\mathcal{F}^{(k)}(:,:,j) = \mathbf{\Phi}_j^{(k)}$, $j = 1, \ldots, r$, $\mathcal{F} \in \mathbb{C}^{p\times m\times r}$.

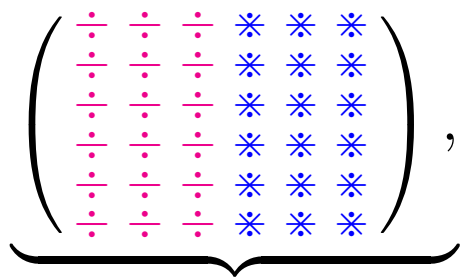
The residual for the input–output pair $(u, v) \in \{1, \ldots, p\} \times \{1, \ldots, m\}$ is

$$\left\| \mathcal{D}_\rho \left(\mathscr{C}^{(k+1)}, \quad -D^{(uv)}\mathscr{C}^{(k+1)} \right) \begin{pmatrix} \mathcal{F}^{(k+1)}(u,v,:) \\ \boldsymbol{\varphi}^{(k+1)} \end{pmatrix} - \mathcal{D}_\rho \mathbb{S}(u,v,:) \right\|_2^2 ,$$

$$\mathcal{D}_\rho = \operatorname{diag}(\sqrt{\rho_i}) \ , \ D^{(uv)} = \operatorname{diag}(S_{uv}^{(i)})_{i=1}^{\ell}, \ \boldsymbol{\varphi}^{(k+1)} = (\varphi_1^{(k+1)}, \ldots, \varphi_r^{(k+1)})^T .$$

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

VF details - assembling the global erroor $A^{(k+1)}x^{(k+1)} - b$

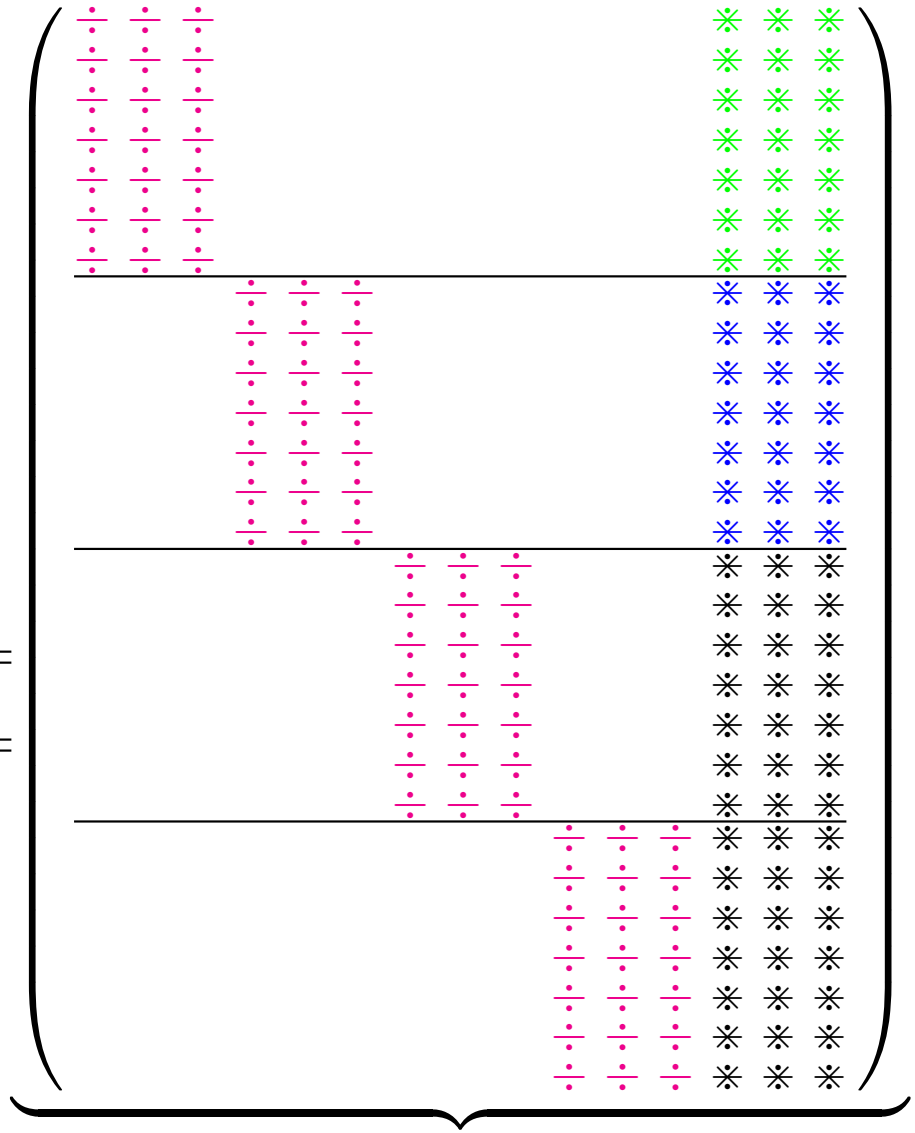


$$\underbrace{\qquad\qquad}_{\left(\mathscr{C}^{(k+1)},\quad -D^{(uv)}\mathscr{C}^{(k+1)}\right)}$$

paired Cauchy structure can be exploited for more accurate computation

block row = input-output pair (u, v)

$\kappa_2\left(\mathscr{C}^{(k+1)}\right) = \dfrac{\sigma_{\max}\left(\mathscr{C}^{(k+1)}\right)}{\sigma_{\min}\left(\mathscr{C}^{(k+1)}\right)}$ improves as $k \to \infty$



$$\underbrace{\qquad\qquad\qquad\qquad}_{\mathbf{A}^{(k+1)},\ell=7,r=3,p=m=2}$$

$p \cdot m$ QR factorization(s) of size $\ell \times 2 \cdot r$

$$\begin{pmatrix} (R^{(k+1)})_{11} & (R_{uv}^{(k+1)})_{12} \\ 0 & (R_{uv}^{(k+1)})_{22} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} * & * & * & \times & \times & \times \\ 0 & * & * & \times & \times & \times \\ 0 & 0 & * & \times & \times & \times \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & 0 & \star & \star \\ 0 & 0 & 0 & 0 & 0 & \star \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

single pivoted QR factorization $*$

QR factorizations with column pivoting $\star$

not computed $\times$

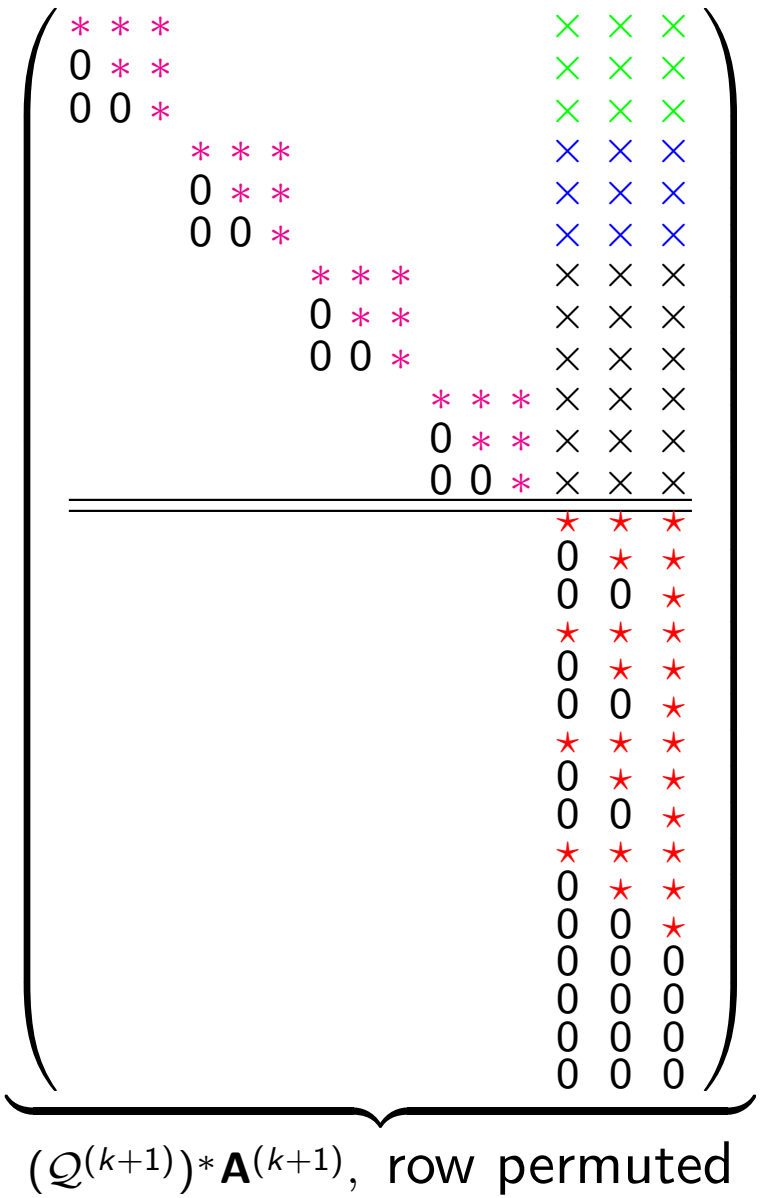
---

faster implementation, LAPACK, BLAS

numerically more robust; details follow ...

$$\underbrace{\begin{pmatrix} * & * & * & & & & & & & \times & \times & \times \\ 0 & * & * & & & & & & & \times & \times & \times \\ 0 & 0 & * & & & & & & & \times & \times & \times \\ 0 & 0 & 0 & & & & & & & \star & \star & \star \\ 0 & 0 & 0 & & & & & & & 0 & \star & \star \\ 0 & 0 & 0 & & & & & & & 0 & 0 & \star \\ 0 & 0 & 0 & & & & & & & 0 & 0 & 0 \\ & & & * & * & * & & & & \times & \times & \times \\ & & & 0 & * & * & & & & \times & \times & \times \\ & & & 0 & 0 & * & & & & \times & \times & \times \\ & & & 0 & 0 & 0 & & & & \star & \star & \star \\ & & & 0 & 0 & 0 & & & & 0 & \star & \star \\ & & & 0 & 0 & 0 & & & & 0 & 0 & \star \\ & & & 0 & 0 & 0 & & & & 0 & 0 & 0 \\ & & & & & & * & * & * & \times & \times & \times \\ & & & & & & 0 & * & * & \times & \times & \times \\ & & & & & & 0 & 0 & * & \times & \times & \times \\ & & & & & & 0 & 0 & 0 & \star & \star & \star \\ & & & & & & 0 & 0 & 0 & 0 & \star & \star \\ & & & & & & 0 & 0 & 0 & 0 & 0 & \star \\ & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{(\mathcal{Q}^{(k+1)})^* \mathbf{A}^{(k+1)}}$$

Reorder the block-rows for the LS solution for $\varphi^{(k+1)}$



$(\mathcal{Q}^{(k+1)})^* \mathbf{A}^{(k+1)}$, row permuted

$$= \begin{pmatrix} \mathbf{B}^{(k+1)}_{[11]} & \mathbf{B}^{(k+1)}_{[12]} \\ 0 & \mathbf{B}^{(k+1)}_{[22]} \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{B}^{(k+1)}_{[11]} & \mathbf{B}^{(k+1)}_{[12]} \\ 0 & \mathbf{B}^{(k+1)}_{[22]} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{\Phi}^{(k+1)} \\ \varphi^{(k+1)} \end{pmatrix} \approx \begin{pmatrix} \mathbf{s}_1^{(k+1)} \\ \mathbf{s}_2^{(k+1)} \\ \mathbf{s}_3^{(k+1)} \end{pmatrix}$$

VF algorithm: global structure

1. Given: The sampling data $\mathbf{H}(\xi_i) \in \mathbb{C}^{p\times m}$ for $i = 1, \ldots, \ell$; 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 $k \leq k_{\max}$ }
   - Form $\mathbf{A}^{(k+1)}$ and $\mathbf{b}$.
   - 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 shown on previous slide.
   - Solve $\|\mathbf{B}_{[22]}^{(k+1)} \boldsymbol{\varphi}^{(k+1)} - \mathbf{s}_2^{(k+1)}\|_2 \longrightarrow \min$ for $\boldsymbol{\varphi}^{(k+1)}$.
   - Set $k \leftarrow k+1$ and compute $\boldsymbol{\lambda}^{(k+1)} = zeros(1 + \sum_{j=1}^{r} \varphi_j^{(k)}/(s - \lambda_j^{(k)}))$.
4. END WHILE
5. $\boldsymbol{\Phi} = (\mathbf{B}_{[11]}^{(k)})^{-1} \mathbf{s}_1^{(k)}$. ($\Longleftrightarrow$ For $u = 1, \ldots, p$, $v = 1, \ldots, m$, solve $\|\mathcal{D}_\rho \left(\mathscr{C}^{(k+1)} \Phi^{(k+1)}(u, v, :) - \mathbb{S}(u, v, :)\right) \|_2 \longrightarrow \min$.)

For efficient implementation we use underlying Kronecker product structure. This allows efficient use of BLAS 3 operations.

Stopping citerion

New stopping criterion

We propose to declare the $|\varphi_j^{(k+1)}|$s "small enough" if

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

A justification

This assures $\max_{s \in i\mathbb{R}} |d^{(k+1)}(s) - 1| \leq \theta^{(k+1)}$, and we can write

$$\mathbf{S}^{(i)}(1 + \sum_{j=1}^{r} \frac{\varphi_j^{(k+1)}}{\xi_i - \lambda_j^{(k+1)}}) = \mathbf{S}^{(i)} + \Delta\mathbf{S}^{(i)}, \quad \begin{array}{l} \mathbf{S}^{(i)} = \mathbf{H}(\xi_i) + \mathcal{E}_i, \\ \|\Delta\mathbf{S}^{(i)}\|_F \leq \theta^{(k+1)} \|\mathbf{S}^{(i)}\|_F \end{array},$$

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

LS solution for $\varphi^{(k+1)}$: Recall the QR factorizations

$$\begin{pmatrix} (R^{(k+1)})_{11} & (R_{uv}^{(k+1)})_{12} \\ 0 & (R_{uv}^{(k+1)})_{22} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} * & * & * & \times & \times & \times \\ 0 & * & * & \times & \times & \times \\ 0 & 0 & * & \times & \times & \times \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & 0 & \star & \star \\ 0 & 0 & 0 & 0 & 0 & \star \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B}_{[22]}^{(k+1)} = \begin{pmatrix} (R_{11}^{(k+1)})_{22} \\ (R_{21}^{(k+1)})_{22} \\ \vdots \\ (R_{pm}^{(k+1)})_{22} \end{pmatrix}$$

$$\| \mathbf{B}_{[22]}^{(k+1)} \varphi^{(k+1)} - \mathbf{s}_2^{(k+1)} \|_2 \longrightarrow \min$$

$$\underbrace{\begin{pmatrix} * & * & * & & & & & \times & \times & \times \\ 0 & * & * & & & & & \times & \times & \times \\ 0 & 0 & * & & & & & \times & \times & \times \\ 0 & 0 & 0 & & & & & \star & \star & \star \\ 0 & 0 & 0 & & & & & 0 & \star & \star \\ 0 & 0 & 0 & & & & & 0 & 0 & \star \\ 0 & 0 & 0 & & & & & 0 & 0 & 0 \\ \hline & & & * & * & * & & \times & \times & \times \\ & & & 0 & * & * & & \times & \times & \times \\ & & & 0 & 0 & * & & \times & \times & \times \\ & & & 0 & 0 & 0 & & \star & \star & \star \\ & & & 0 & 0 & 0 & & 0 & \star & \star \\ & & & 0 & 0 & 0 & & 0 & 0 & \star \\ & & & 0 & 0 & 0 & & 0 & 0 & 0 \\ \hline & & & & & & * & \times & \times & \times \\ & & & & & & 0 & \times & \times & \times \\ & & & & & & 0 & \times & \times & \times \\ & & & & & & 0 & \star & \star & \star \\ & & & & & & 0 & 0 & \star & \star \\ & & & & & & 0 & 0 & 0 & \star \\ & & & & & & 0 & 0 & 0 & 0 \end{pmatrix}}_{(\mathcal{Q}^{(k+1)})^* \mathbf{A}^{(k+1)}}$$

Least Squares Solutions: A review

$K \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, $m \geq n$ ($m \gg n$). Solve $\|Kx - b\|_2 \longrightarrow \min$
$K = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^*$ the SVD of K;

$$\Sigma = \begin{pmatrix} \widehat{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \widehat{\Sigma} = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix}, \quad \sigma_1 \geq \cdots \geq \sigma_r > 0$$

Let $U_r = U(:, 1:r)$, $V_r = V(:, 1:r)$.

$$\begin{aligned} \|Kx - b\|_2 &= \|U\Sigma(V^*x) - b\|_2 = \|\Sigma y - c\|_2 \quad (y = V^*x, c = U^*b) \\ &= \left\| \begin{pmatrix} \widehat{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \right\|_2 \longrightarrow \min, \quad \underbrace{\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \widehat{\Sigma}^{-1} c_1 \\ \mathbf{0} \end{pmatrix}}_{\|y\|_2 \to \min} \end{aligned}$$

$x = Vy = V_r \widehat{\Sigma}^{-1} U_r^* b \equiv V \begin{pmatrix} \Sigma \\ \mathbf{0}_{m-n,n} \end{pmatrix}^\dagger U^* b = V(\Sigma^\dagger \ \mathbf{0}_{n,m-n}) U^* b \equiv K^\dagger b$ is with smallest $\|x\|_2$ such that $\|Kx - b\|_2 \longrightarrow \min$. $K^\dagger$ is the pseudoinverse of K. DO NOT USE IT EXPLICITLY!

Least Squares Solutions: A review

$$\underbrace{K}_{m\times n}\overbrace{P} = Q\begin{pmatrix} R \\ 0 \end{pmatrix},\quad R = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \hline 0 & 0 & \blacksquare & \bullet & \blacksquare & \blacklozenge \\ 0 & 0 & 0 & \bullet & \blacksquare & \blacklozenge \\ 0 & 0 & 0 & 0 & \blacksquare & \blacklozenge \\ 0 & 0 & 0 & 0 & 0 & \blacklozenge \end{pmatrix}$$

permutation

$$Q^*Q = I_m.$$

$$|R_{ii}| \geq \sqrt{\sum_{k=i}^{j} |R_{kj}|^2}, \quad \text{for all } 1 \leq i \leq j \leq n. \tag{2.1}$$

$$|R_{11}| \geq |R_{22}| \geq \cdots \geq |R_{\rho\rho}| \gg |R_{\rho+1,\rho+1}| \geq \cdots \geq |R_{nn}| \tag{2.2}$$

The structure (2.1), (2.2) may not be rank revealing but it must be guaranteed by the software (e.g. LAPACK, Matlab). Implemented in LINPACK in 1971., adopted by (Sca)LAPACK and used in many packages.

Least Squares Solutions: A review

Let $\mathrm{rank}(K) = r$,

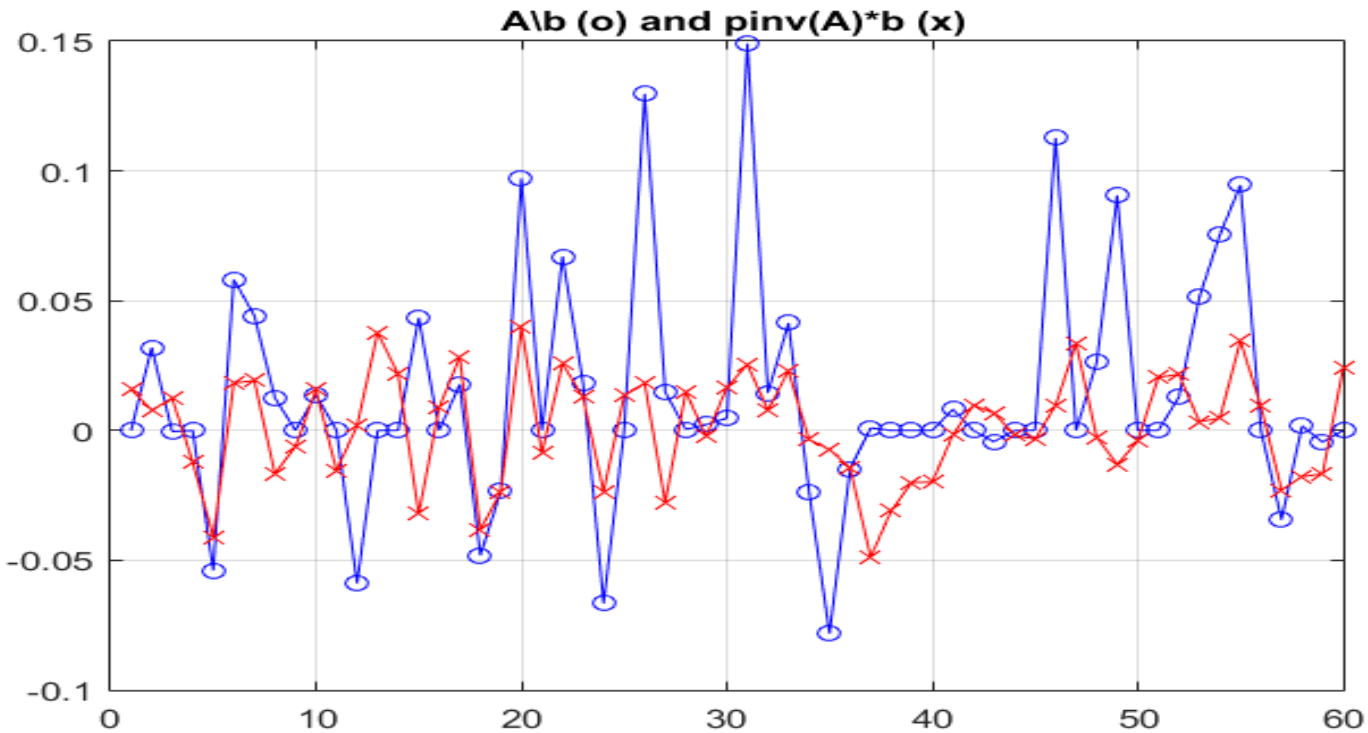
$$KP = QR = \begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad Q_r = Q(:, 1:r)$$

$$\begin{aligned} \|Kx - b\|_2 &= \|Q \begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} P^T x - b\|_2 \\ &= \| \begin{pmatrix} R_{[11]} & R_{[12]} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \|_2 \to \min, \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} R_{[11]}^{-1} c_1 \\ \mathbf{0} \end{pmatrix} \end{aligned}$$

$x = Py = P \begin{pmatrix} R_{[11]}^{-1} Q_r^* b \\ \mathbf{0} \end{pmatrix}$. x has at least $n - r$ zeros.

If $r = n$ then $x = K^\dagger b$ by virtue of uniqueness. Otherwise, this x is different from $K^\dagger b$.

Least Squares Solutions: Example, 100×60 of rank 40



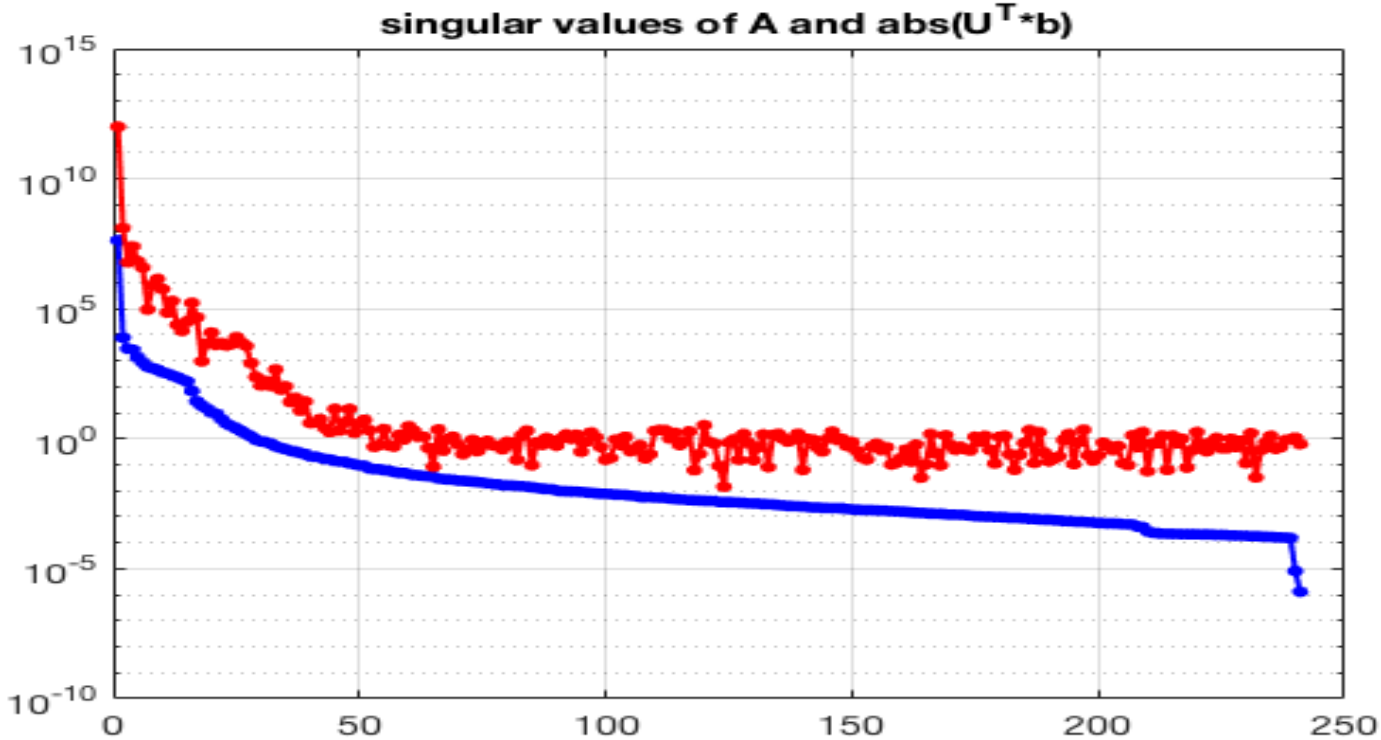
x1=A\b, x2=pinv(A)*b; norm(x1)=0.3539; norm(x2)=0.1599

norm(A*x1-b)=7.623047315933105e+00

norm(A*x2-b)=7.623047315933104e+00

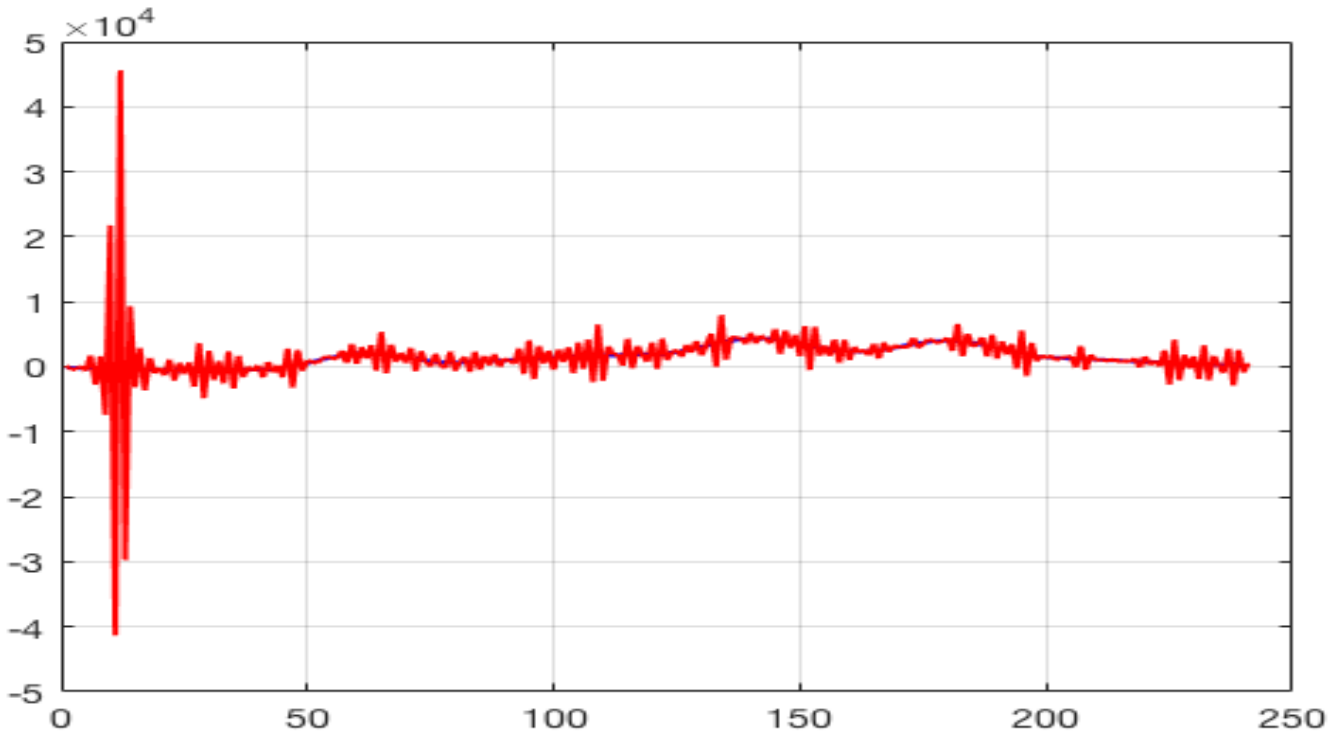
Truncating the SVD - Example (Phillips integral eq. 1962)

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = V\Sigma^{\dagger}U^{T}b = \sum_{i}^{240} v_i \frac{u_i^T b}{\sigma_i}$,



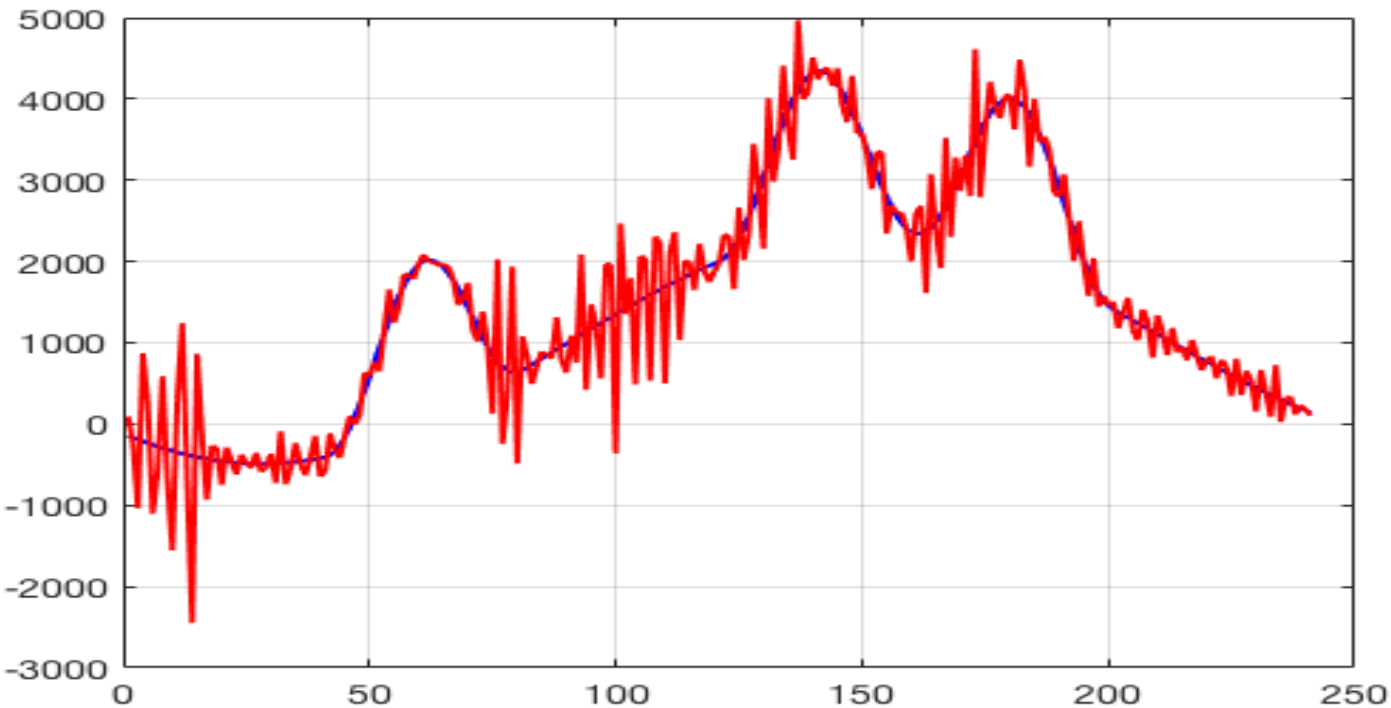
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{240} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution. What blue curve?!



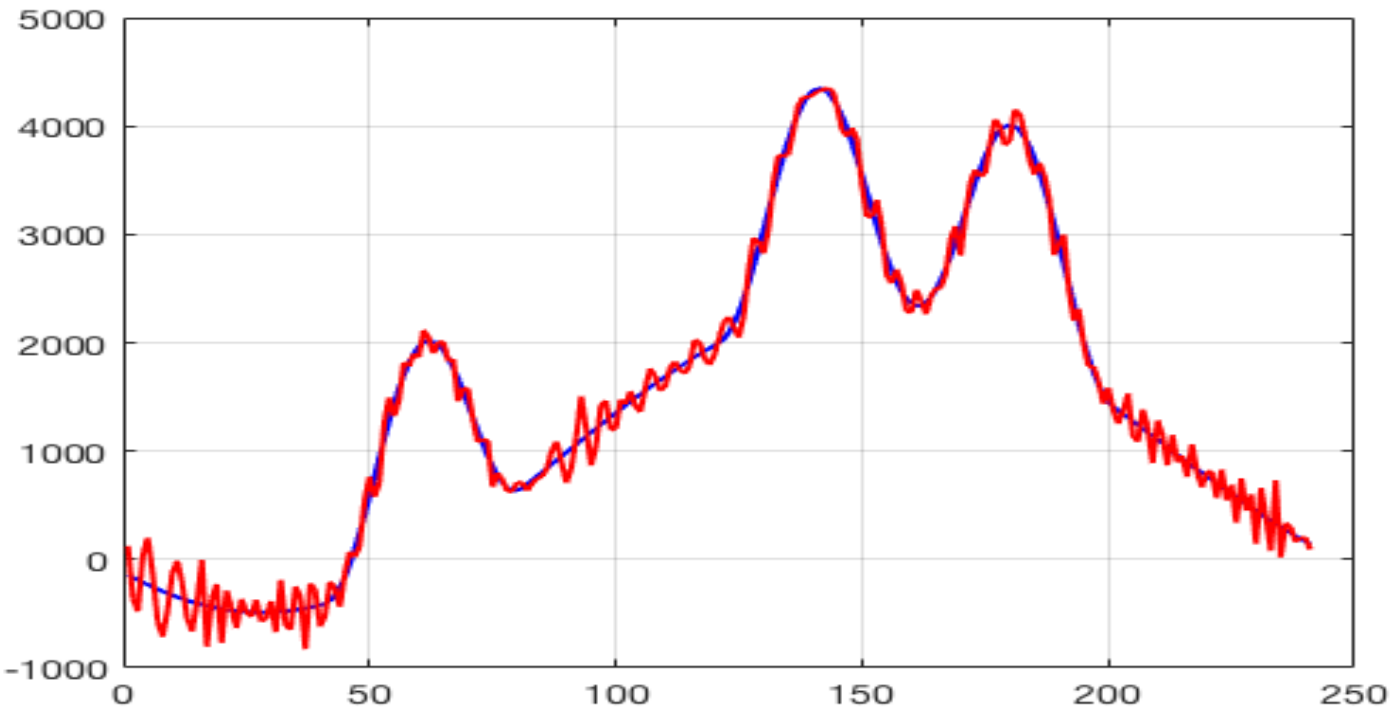
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{200} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution.



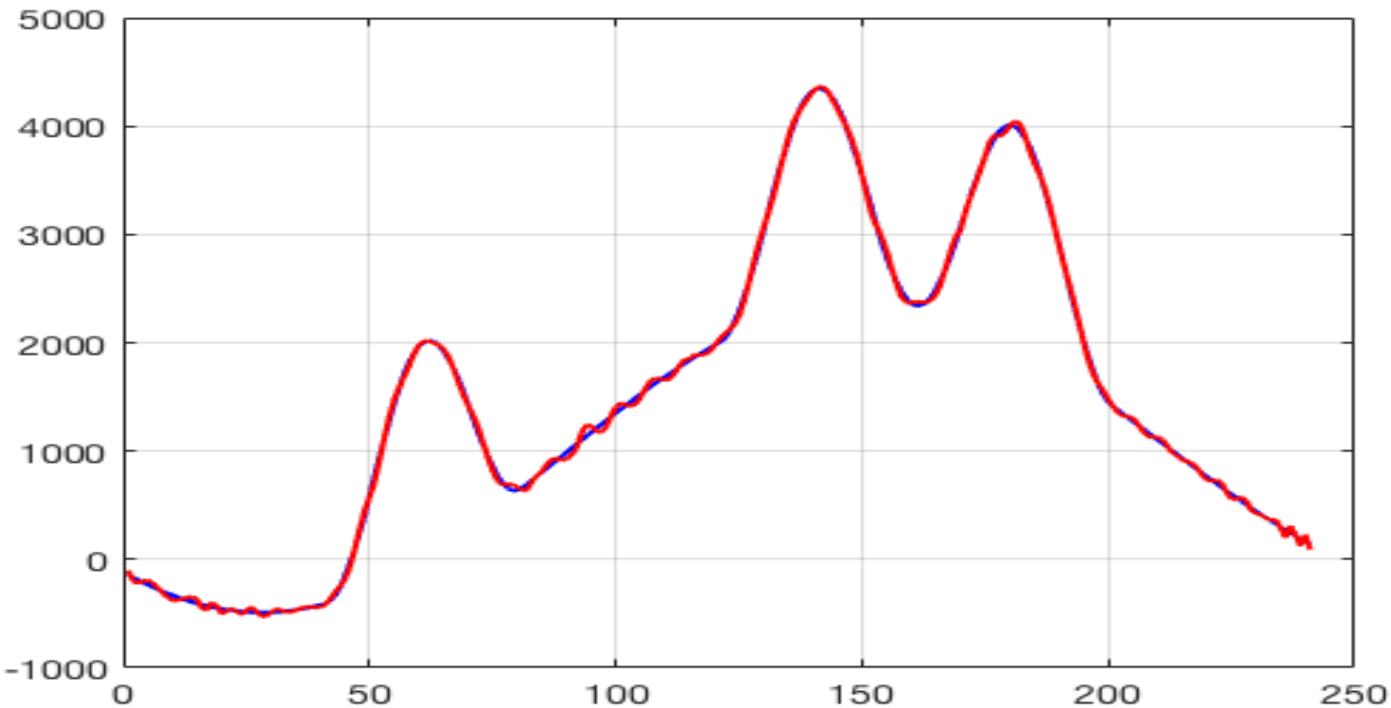
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{150} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution.



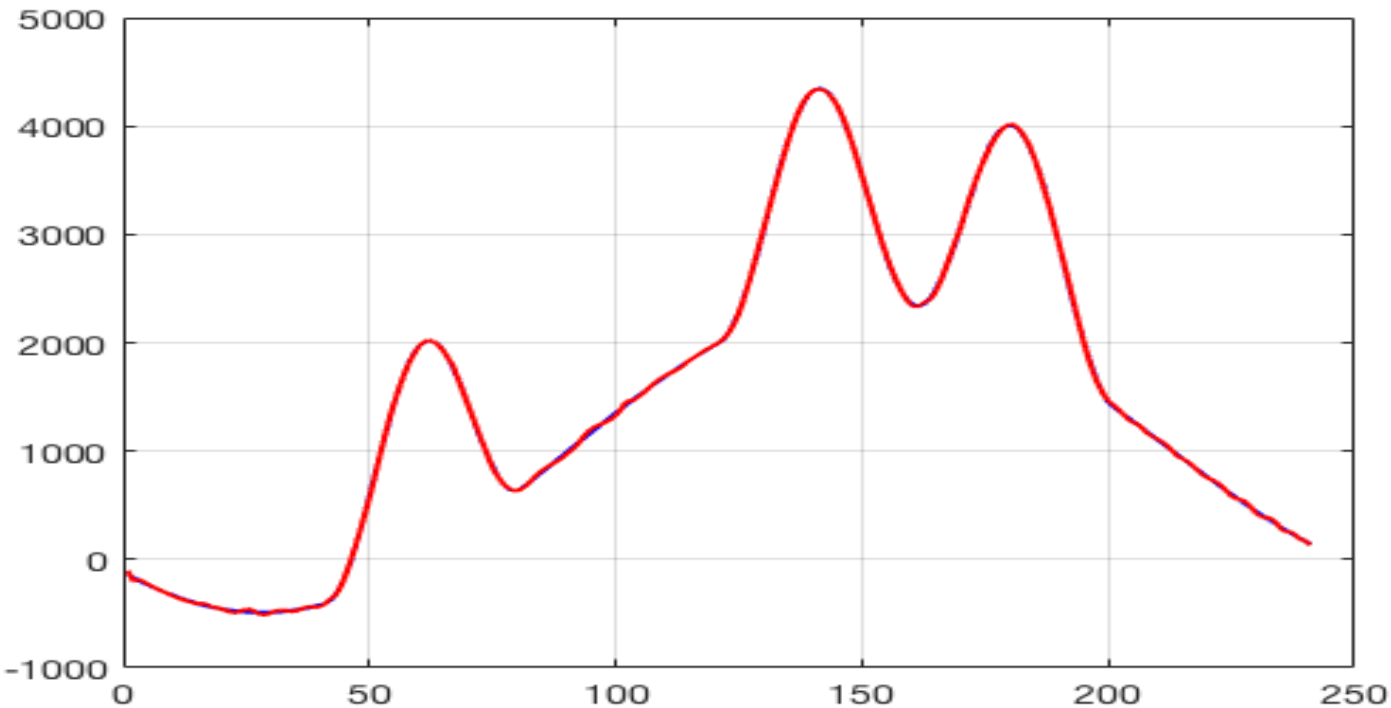
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{100} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution.



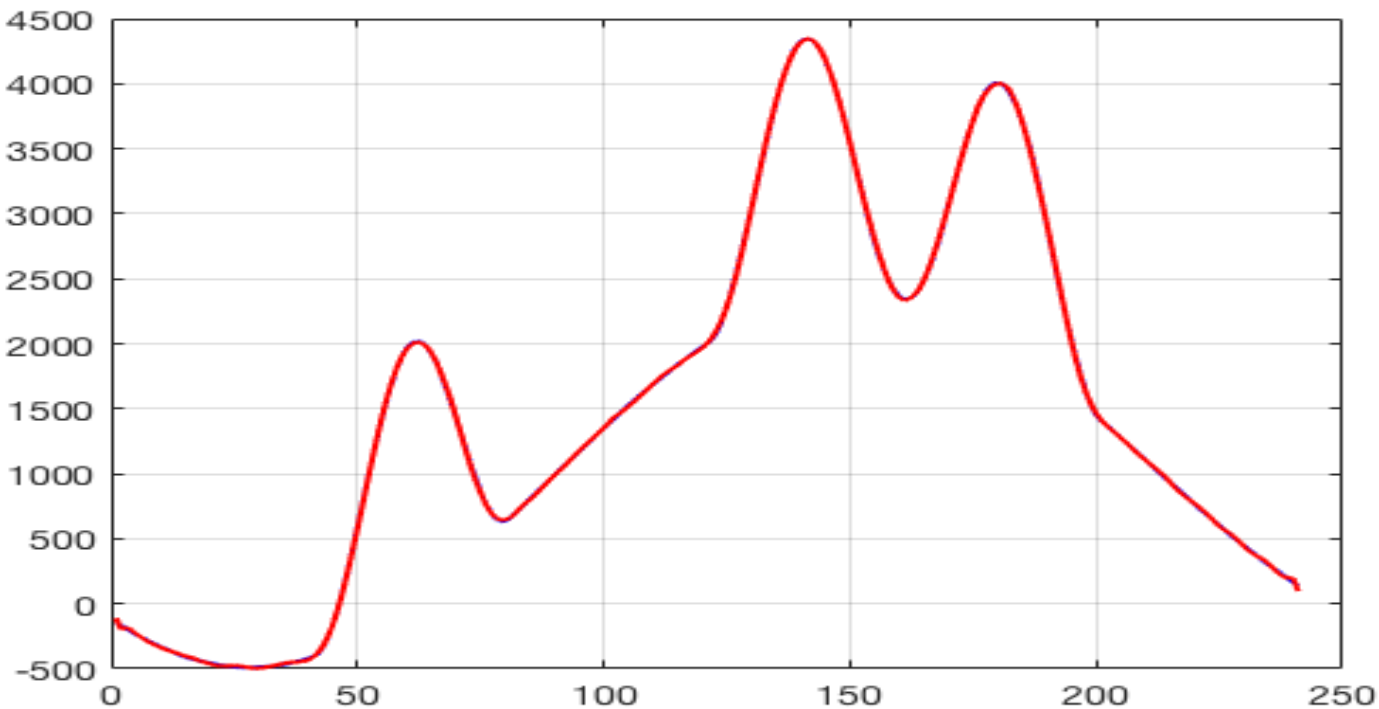
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{80} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution.



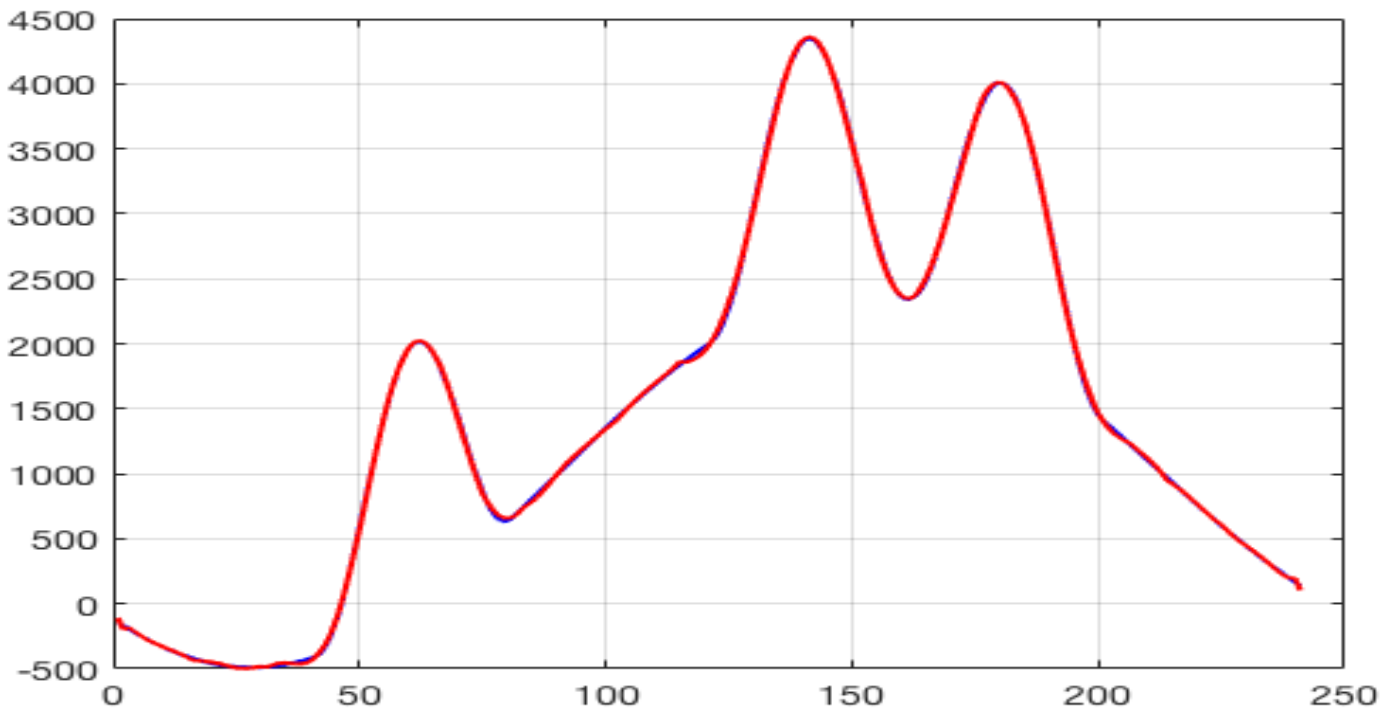
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{60} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution. What blue curve?



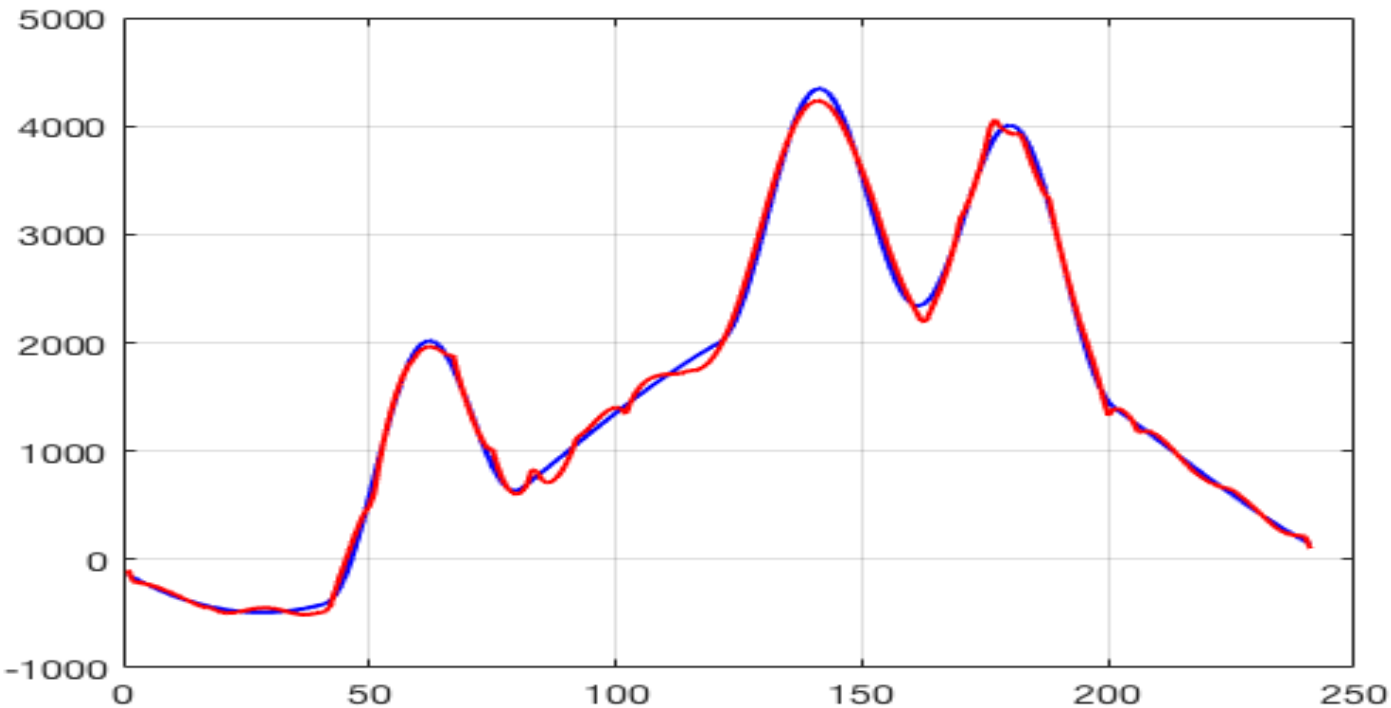
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{40} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution. What blue curve?



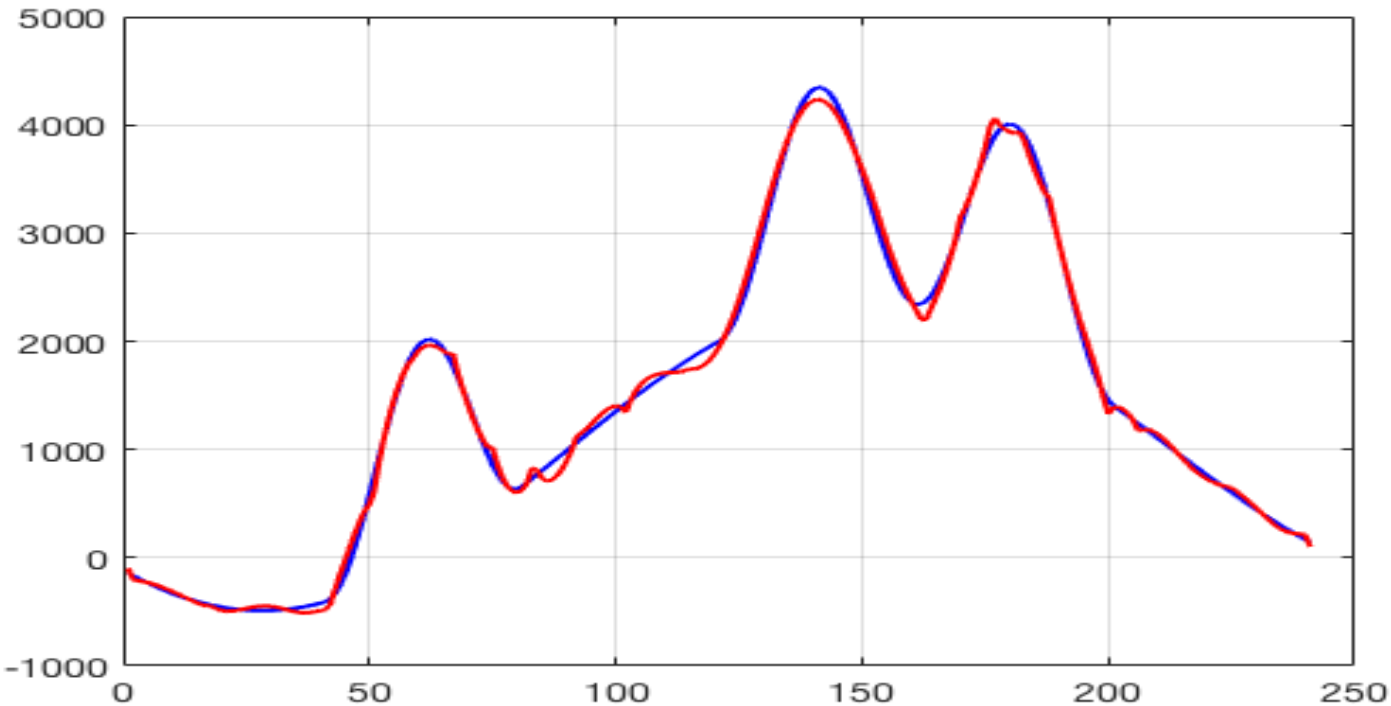
Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{30} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution.



Truncating the SVD - Example

Phillips integral equation discretized: $Ax \approx b$; $\mathrm{rank}(A) = 240$;
$x = \sum_i^{25} v_i \frac{u_i^T b}{\sigma_i}$, the blue curve is the solution.



See P.C. Hansen on L-curve theory. B. W. Rust on truncated SVD.

Computing the residues: $\Phi = (\mathbf{B}^{(k)}_{[11]})^{-1}\mathbf{s}^{(k)}_1$.

$\Updownarrow$ For $u = 1:p,\ v = 1:m,$

$$\|\mathcal{D}_\rho\left(\mathscr{C}^{(k+1)}\Phi^{(k+1)}(u,v,:) - \mathbb{S}(u,v,:)\right)\|_2 \longrightarrow \min\,.$$

To simplify the notation, write $\|\mathcal{D}_\rho\mathscr{C}x - h\|_2 \longrightarrow \min$, where $\mathscr{C} = \mathscr{C}_{\boldsymbol{\xi},\boldsymbol{\lambda}}$ is a Cauchy matrix as in (1.4), h is the corresponding scaled right-hand side, $\boldsymbol{\lambda}$ is closed under conjugation and and the solution vector should also be closed under conjugation. Such a constrained problem can be replaced by an equivalent unconstrained LS problem

$$\left\|\begin{pmatrix}\mathcal{D}_\rho\mathscr{C}_{\boldsymbol{\xi},\boldsymbol{\lambda}}\\ \mathcal{D}_\rho\mathscr{C}_{\overline{\boldsymbol{\xi}},\boldsymbol{\lambda}}\end{pmatrix}x - \begin{pmatrix}h\\ \overline{h}\end{pmatrix}\right\|_2 \equiv \|\widehat{\mathscr{C}}x - \widehat{h}\|_2 \longrightarrow \min \tag{2.3}$$

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

Ragularization: Tichonov and Morozov

Let $\widehat{\mathscr{C}} = W\Sigma V^*$ be the SVD and let the unique[1] 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.

Tichonov regularization

Choose $\mu \geq 0$ and use the solution of $\|\widehat{\mathscr{C}}x - \widehat{h}\|_2^2 + \mu^2\|x\|_2^2 \to \min$,

$$x_\mu = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \mu^2}(w_i^*\widehat{h})v_i. \tag{2.4}$$

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

[1] Since all nodes are distinct and the poles are assumed simple, the matrix is of full column rank.

SVD($D_1 \times$ Cauchy $\times D_2$)

Given Cauchy matrix $C = C(x, y)$ and any two diagonal matrices D_1, D_2, the SVD of $G = D_1 C D_2$ can be computed to nearly fully precision as follows (Demmel):

1. Compute the LDU, $P_1 G P_2 = LDU$ using explicit determinant based formulas to update the Schur complement. This is entry wise forward stable computation of L, D, U. Moreover, $\kappa(L)$, $\kappa(U)$ are moderate. (Small $\|\delta L\|/\|L\|$, $\|\delta U\|/\|U\|$, $|\delta D_{ii}|/|D_{ii}|$ is also OK) ($G = hilb(100)$, $\kappa_2(G) > \mathbf{10}^{150}$, $\kappa_2(L) = \kappa_2(U) \approx 72.24$, $\kappa_2(D) \approx 2.32 \cdot \mathbf{10}^{149}$)

2. Compute the SVD of the product LDU using a Jacobi type SVD algorithm (Drmač). The forward error is determined by $\max(\kappa(L), \kappa(U))$. The backward errors $\|\Delta L\|/\|L\|$, $\|\Delta U\|/\|U\|$, $\Delta D_{ii}/D_{ii}$ are small.

condition number(condition number) = condition number

```
>> cond(hilb(100))
ans = 4.6226e+19
```

LS with Cauchy matrix

```
function [L, D, U, indexr, indexc ] = Cauchy_RRD( x, y, Dr, Dc )
% LDU with complete pivoting of Cauchy matrix G=(Dr(i)*Dc(j)/(x(i)+y(j)))_{i=1:m,j=1:n}
m = max(size(x)) ; n = max(size(y)) ; indexr = 1:m ; indexc = 1:n ;
G = ( Dr * Dc.' )./( x*ones(1,n) + ones(m,1)*y.' ) ;
for k = 1 : min(m,n)
[ colmax, jcm] = max( abs( G(k:m,k:n) ) )                    ;
[ ~, jm ] = max( colmax ) ; im = jcm(jm)+k-1 ; jm = jm+k-1 ;
itmp = indexr(k) ; indexr(k) = indexr(im) ; indexr(im) = itmp ;
itmp = indexc(k) ; indexc(k) = indexc(jm) ; indexc(jm) = itmp ;
tmp = x(k) ; x(k) = x(im) ; x(im) = tmp ; tmp = y(k) ; y(k) = y(jm) ; y(jm) = tmp ;
vtmp = G(k,:) ; G(k,:) = G(im,:) ; G(im,:) = vtmp ;
vtmp = G(:,k) ; G(:,k) = G(:,jm) ; G(:,jm) = vtmp ;
for r = k + 1 : m
G(r,k+1:n)=(G(r,k+1:n).*(y(k+1:n).'-y(k))./(x(k)+y(k+1:n).'))*((x(r)-x(k))/(x(r)+y(k)));
end
end
D=diag(G); L=tril(G,-1)*diag(1./D)+eye(m,n); U=diag(1./D)*triu(G(1:n,1:n),1)+eye(n);
end
```

PSVD($X \cdot D \cdot Y^*$), D diagonal; X, Y well conditioned

1. $XDY^* = \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix} \Delta_X D \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix} = \widetilde{X}\widetilde{Y}^*$; $\widetilde{X}$ has unit columns; $\Delta_X = \mathrm{diag}(\|X(:,i)\|)$, $\widetilde{Y}^* = \Delta_X D Y^*$.
2. QRCP: $\widetilde{Y}P = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$; it holds that $XDY^* = (\widetilde{X}P) \begin{pmatrix} R^* & 0 \end{pmatrix} Q^*$;
3. Need SVD of $\widetilde{A} = (\widetilde{X}P)R^*$, where $R^* = \begin{pmatrix} \blacksquare & & \\ \blacksquare & \blacksquare & \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$ has dominant diagonal and $\min_{\Delta=\mathrm{diag}} \kappa_2(\widetilde{A}\Delta) \leq \kappa_2(\widetilde{X}) \min_{\Delta=\mathrm{diag}} \kappa_2(R^*\Delta)$
4. $[U, \Sigma, V_1]$=SVD($\widetilde{A}$); Jacobi SVD of **explicitly computed** $\widetilde{A}$
5. With $V = Q \begin{pmatrix} V_1 & 0 \\ 0 & I_{n-p} \end{pmatrix}$, the SVD is $XDY^* = U\Sigma V^*$

The SVD will be accurate if $\min_{\mathbf{\Delta}=\mathrm{diag}} \kappa_2(\mathbf{X\Delta})$ and $\min_{\mathbf{\Delta}=\mathrm{diag}} \kappa_2(\mathbf{Y\Delta})$ are moderate. Detailed analysis in Z.D. 1998.

PSVD application: Hankel SVD, $\sigma_i = \sqrt{\lambda_i(HM)}$

$\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t)$
Grammians $H = \int_0^\infty e^{tA}BB^Te^{tA^T}dt$, $M = \int_0^\infty e^{tA^T}C^TCe^{tA}dt$ via Lyapunov equations $AH + HA^T = -BB^T$, $A^TM + MA = -C^TC$.
Let $H = L_HL_H^T$, $M = L_ML_M^T$, where L_H, L_M are the Cholesky factors computed by the Hammarling algorithm. Solve $HMx = \lambda x$ via the SVD of $L_M^TL_H$, using the PSVD($L_M^TL_H^T$) algorithm. The algorithm solves

$$(H + \delta H)(M + \delta M)x = \tilde{\lambda}x \text{ exactly, with symmetric } \delta H,\ \delta M,$$

$$\frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}} \leq f(n)\cdot\varepsilon, \quad \frac{|\delta M_{ij}|}{\sqrt{M_{ii}M_{jj}}} \leq g(n)\cdot\varepsilon, \quad 1 \leq i,j \leq n$$
$$\frac{|\delta\lambda|}{\lambda} \leq h(n)(\sqrt{\|H_s^{-1}\|_2} + \sqrt{\|M_s^{-1}\|_2})\cdot\varepsilon, \quad \varepsilon = \mathrm{eps}.$$

$H_s = \mathrm{diag}(H)^{-1/2}H\mathrm{diag}(H)^{-1/2}$, $\kappa_2(H_s) \leq n\min_{D=diag}\kappa_2(DHD)$.
Accuracy invariant under changes of physical units in state variables.

Example: ⊠⇝■⇜⇝■⇜⇝■ early loss of definiteness

The stiffness matrix of a mass spring system with 3 masses ⊠⇝■⇜⇝■⇜⇝■ with spring constants $k_1 = k_3 = 1$, $k_2 = \varepsilon/2$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{pmatrix},$$

Let $\varepsilon \leq \text{eps} = \textbf{round-off}$. Then the true and the computed matrix are

$$K = \begin{pmatrix} 1+\frac{\varepsilon}{2} & -\frac{\varepsilon}{2} & 0 \\ -\frac{\varepsilon}{2} & 1+\frac{\varepsilon}{2} & -1 \\ 0 & -1 & 1 \end{pmatrix}, \widetilde{K} = \begin{pmatrix} 1 & -\frac{\varepsilon}{2} & 0 \\ -\frac{\varepsilon}{2} & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}, \max_{i,j} \frac{|\widetilde{K}_{ij} - K_{ij}|}{|K_{ij}|} < \varepsilon/2.$$

$\widetilde{K}$ is the best machine representation of K. However:

K is **positive definite** with $\lambda_{\min}(K) \approx \varepsilon/4$,
$\tilde{K}$ is **indefinite** with $\lambda_{\min}(\widetilde{K}) \approx -\varepsilon^2/8$.
Too late for $\lambda_{\min}(K)$, even in exact computation with $\widetilde{K}$. :(

Example ... implicit formulation

On the other hand, $K = A^T A$ with

$$A = \begin{pmatrix} \sqrt{k_1} & 0 & 0 \\ -\sqrt{k_2} & \sqrt{k_2} & 0 \\ 0 & -\sqrt{k_3} & \sqrt{k_3} \end{pmatrix} = \begin{pmatrix} \sqrt{k_1} & 0 & 0 \\ 0 & \sqrt{k_2} & 0 \\ 0 & 0 & \sqrt{k_3} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

clearly separates physical parameters and the geometry of the connections. The problem reduces to the SVD of A.
Since A is bidiagonal, for any choice of k_1, k_2, k_3, the singular values of A can be computed (zero-shift bidiagonal QR SVD of Demmel and Kahan) to nearly the same number of accurate digits to which the spring constants are given. Hence, in this formulation, the initial eigenvalue problem $Kx = \lambda x$ is perfectly well conditioned.
This is an example of preserving the important qualitative property (definiteness) exactly, using an implicit formulation of the problem and an accurate algorithm.

LS solution for $\varphi^{(k+1)}$: Recall the QR factorizations

$$\begin{pmatrix} (R^{(k+1)})_{11} & (R_{uv}^{(k+1)})_{12} \\ 0 & (R_{uv}^{(k+1)})_{22} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} * & * & * & \times & \times & \times \\ 0 & * & * & \times & \times & \times \\ 0 & 0 & * & \times & \times & \times \\ 0 & 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & 0 & \star & \star \\ 0 & 0 & 0 & 0 & 0 & \star \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B}_{[22]}^{(k+1)} = \begin{pmatrix} (R_{11}^{(k+1)})_{22} \\ (R_{21}^{(k+1)})_{22} \\ \vdots \\ (R_{pm}^{(k+1)})_{22} \end{pmatrix}$$

$$\| \mathbf{B}_{[22]}^{(k+1)} \varphi^{(k+1)} - \mathbf{s}_2^{(k+1)} \|_2 \longrightarrow \min$$

$$\underbrace{\begin{pmatrix} * & * & * & & & & & \times & \times & \times \\ 0 & * & * & & & & & \times & \times & \times \\ 0 & 0 & * & & & & & \times & \times & \times \\ 0 & 0 & 0 & & & & & \star & \star & \star \\ 0 & 0 & 0 & & & & & 0 & \star & \star \\ 0 & 0 & 0 & & & & & 0 & 0 & \star \\ 0 & 0 & 0 & & & & & 0 & 0 & 0 \\ \hline & & & * & * & * & & \times & \times & \times \\ & & & 0 & * & * & & \times & \times & \times \\ & & & 0 & 0 & * & & \times & \times & \times \\ & & & 0 & 0 & 0 & & \star & \star & \star \\ & & & 0 & 0 & 0 & & 0 & \star & \star \\ & & & 0 & 0 & 0 & & 0 & 0 & \star \\ & & & 0 & 0 & 0 & & 0 & 0 & 0 \\ \hline & & & & & & * & \times & \times & \times \\ & & & & & & 0 & \times & \times & \times \\ & & & & & & 0 & \times & \times & \times \\ & & & & & & 0 & \star & \star & \star \\ & & & & & & 0 & 0 & \star & \star \\ & & & & & & 0 & 0 & 0 & \star \\ & & & & & & 0 & 0 & 0 & 0 \end{pmatrix}}_{(\mathcal{Q}^{(k+1)})^* \mathbf{A}^{(k+1)}}$$

Recall the QR factorizations

QR factorizations for $1 \leq u \leq p$, $1 \leq v \leq m$:

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

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

where the unitary matrix $Q_{uv}^{(k+1)}$ has been partitioned as $Q_{uv}^{(k+1)} = \begin{pmatrix} (Q^{(k+1)})_1 & (Q_{uv}^{(k+1)})_2 & (Q_{uv}^{(k+1)})_3 \end{pmatrix}$, with block columns of sizes $\ell \times r$, $\ell \times r$, $\ell \times (\ell - 2r)$, respectively. The leading r columns are *independent* of (u, v); hence the initial part of the factorization, $(Q^{(k+1)})_1 (R^{(k+1)})_{11} = \mathcal{D}_\rho \mathscr{C}^{(k+1)}$, need only be done once.

Convergence introduces noise by severe cancellations

– ease the notation and drop the iteration index k; $x = \Phi(u, v, 1:r)$
$\mathbb{S}(u, v, 1:\ell) \approx \mathscr{C}x + error = (Q)_1(R)_{11}x + error,$

Columns of $(R_{uv}^{(k+1)})_{22}$ and $\mathbf{B}_{[22]}^{(k+1)}$ may suffer severe cancellations

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

LS solution: Matlab's backslash

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

$$\varphi = P \begin{pmatrix} T_{[11]}^{-1} W_1^* \mathbf{s}_2 \\ 0 \end{pmatrix} = \mathbf{B}_{[22]} \backslash \mathbf{s}_2, \quad \text{different from } \mathbf{B}_{[22]}^{\dagger}\mathbf{s}_2 = \mathrm{pinv}(\mathbf{B}_{[22]})\mathbf{s}_2$$

The quandary of column scaling

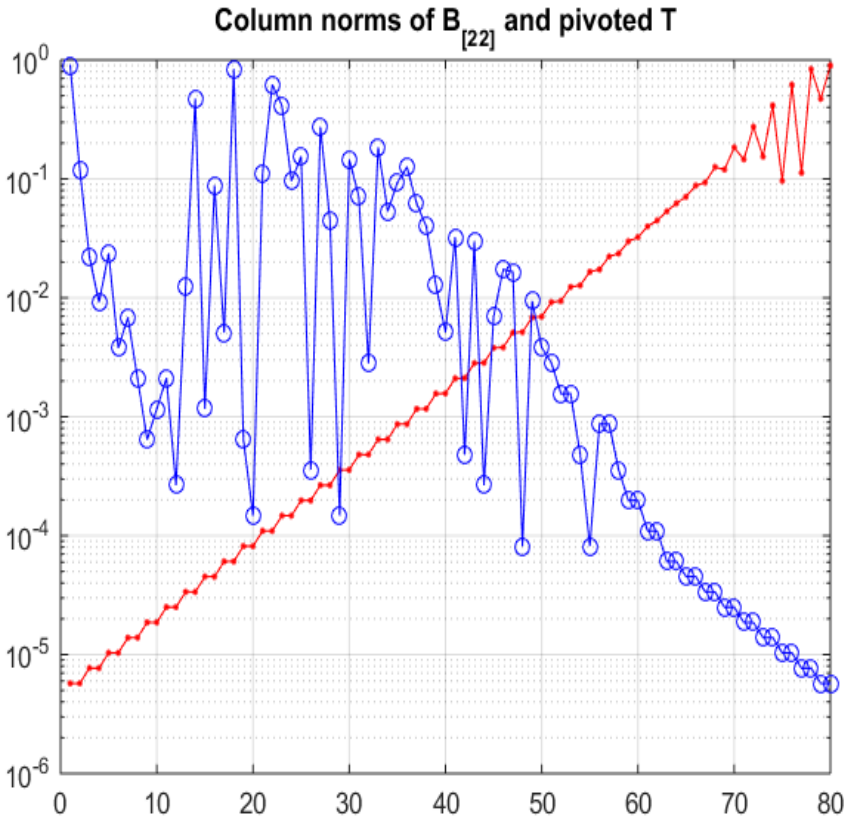
Scaling a noisy matrix

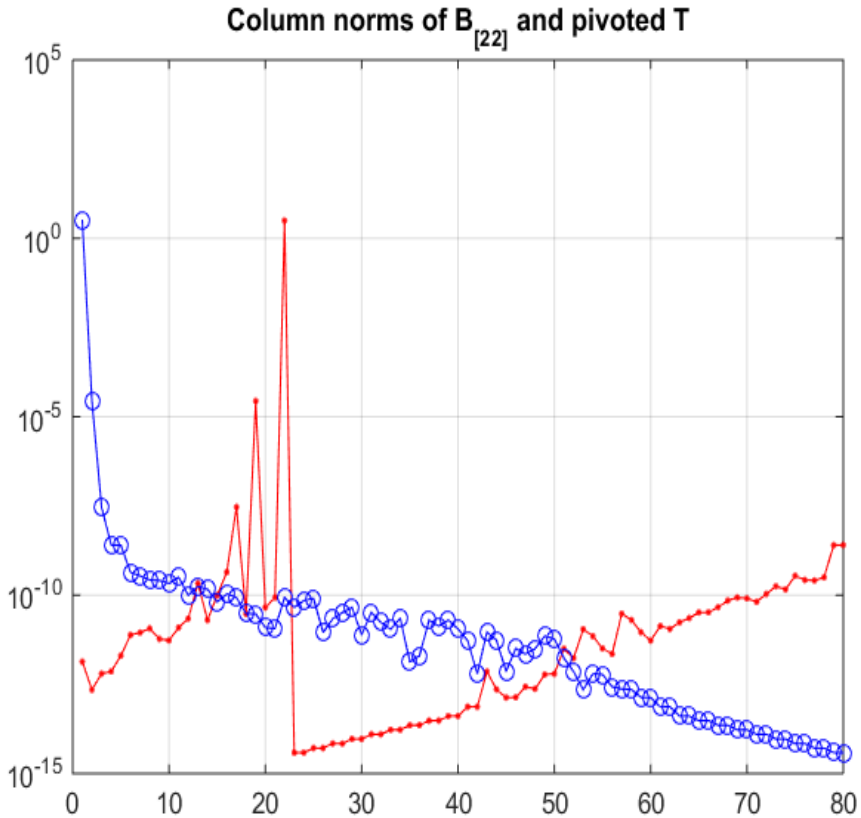
VF literature advocates scaling $\mathbf{B}_{[22]}$ to improve LS solution. However, scaling noisy matrix columns effectively increases the influence of noise, and allows these noisy columns to participate in the column pivoting process of the QR factorization, with a possibility that some of them become drafted and taken upfront as important. This, in turn, interferes with the rank revealing process

Example ($\frac{\partial}{\partial t}T(x,t) = \alpha \frac{\partial^2}{\partial x^2}T(x,t) + u(x,t), \quad 0 < x < 1, \quad t > 0$)

In this NICONET benchmak (n=200), we set $r = 80$, and show the column norms of $\mathbf{B}_{[22]}$ and its pivoted triangular factor T in the first two iterations.

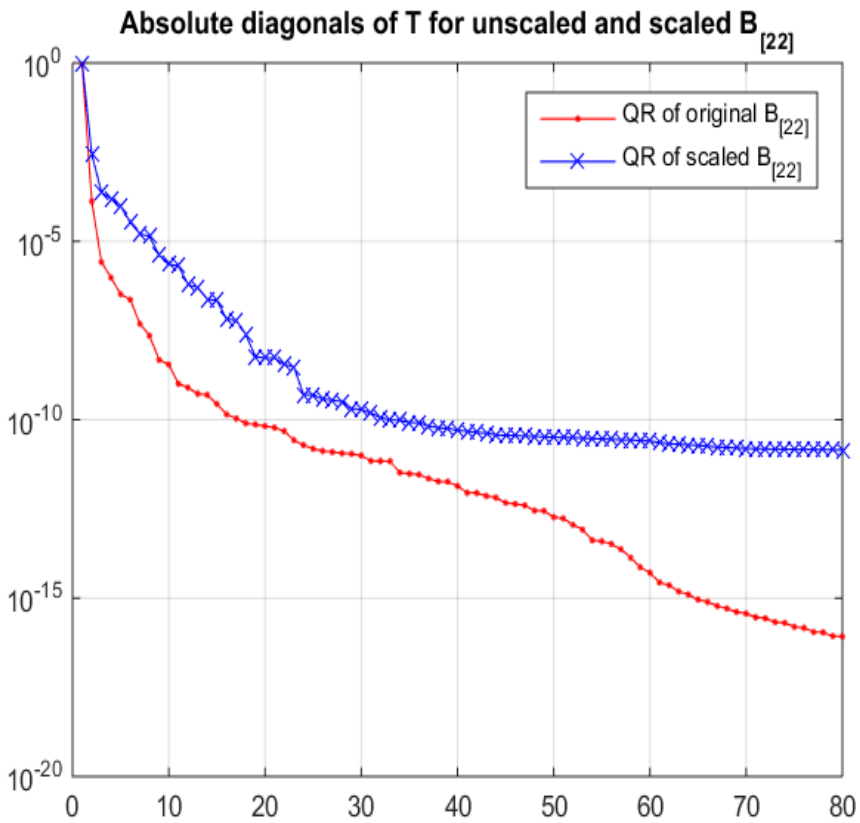
Column norms of $\mathbf{B}_{[22]}$ and T in the first two iterations

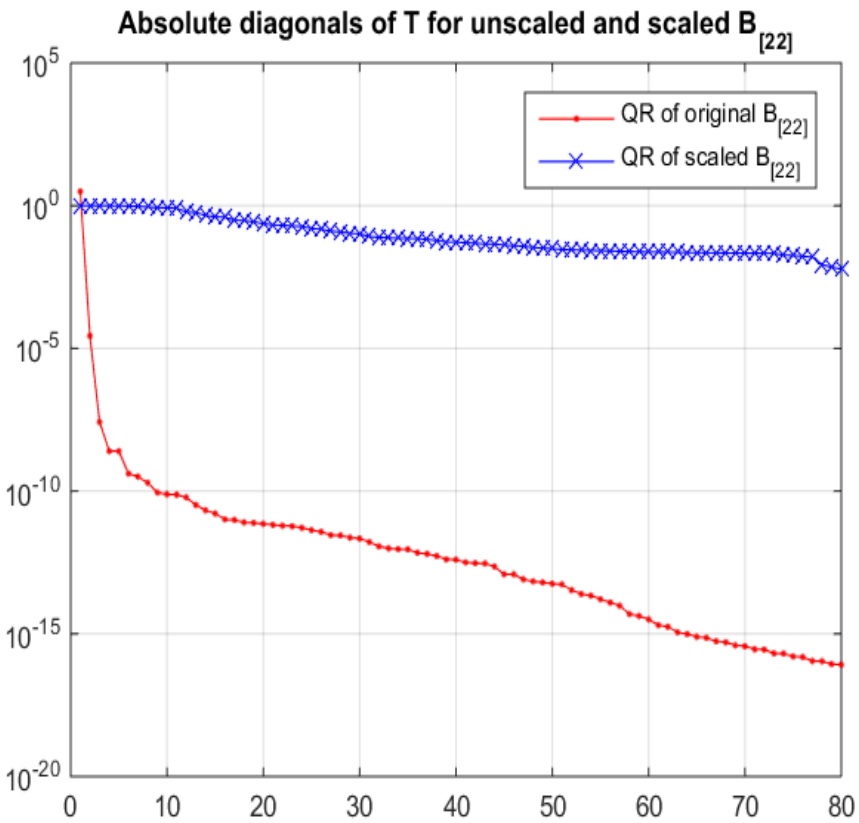




The column norms of $\mathbf{B}_{[22]}$ are marked with (red) $\cdot\!-$, and the column norms its triangular factor in the pivoted QR factorization are marked with (blue) $\circ\!-$. Left: 1. iteration. Right: 2. iteration.

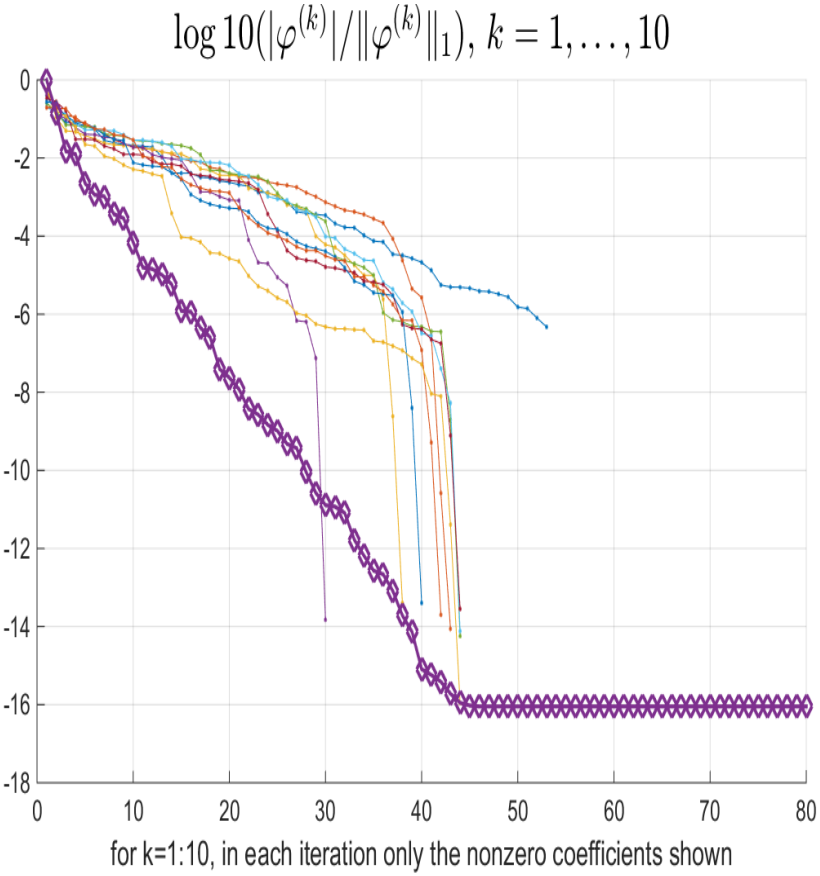
The diagonal of rank-revealing T in the first two iterations

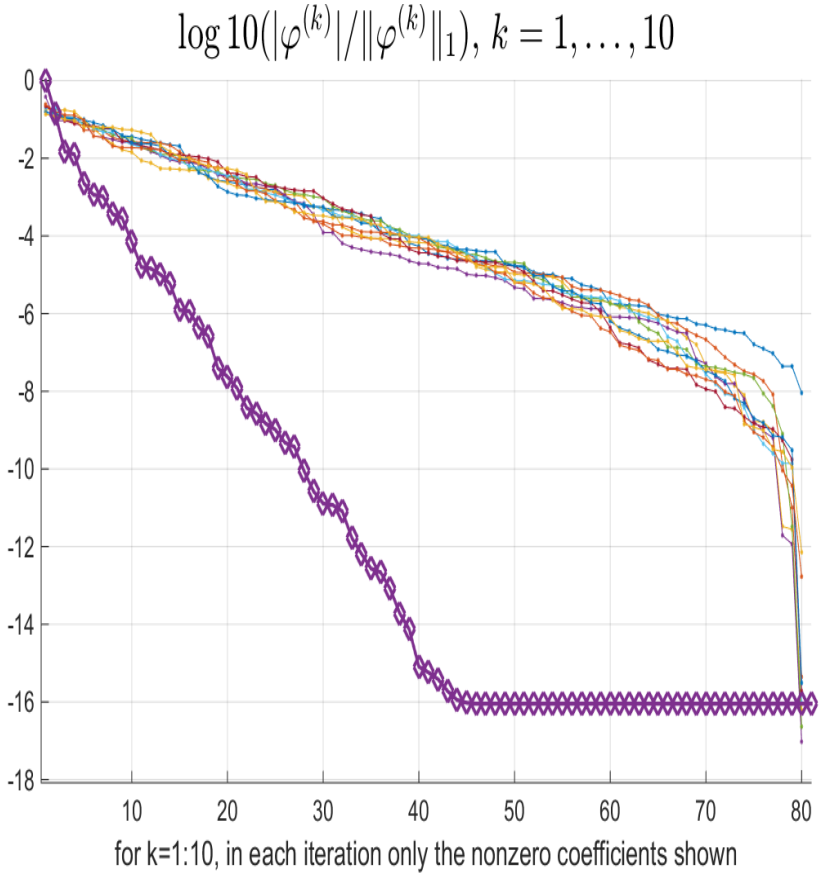




The values $|T_{ii}|$ of the unscaled $\mathbf{B}_{[22]}$ and of the column equilibrated $\mathbf{B}_{[22]}$. Note how scaling unrightfully rehabilitates the noise and precludes inferring numerical rank deficiency.

The diagonal of rank-revealing T in the first two iterations





Sorted $log_{10}(|\varphi_j^{(k)}|//\|\varphi^{(k)}\|_1)$ vs j, $k = 1:10$; only nonzero coefficients are shown. Normalized Hankel singular values are displayed as diamonds $\diamond$.
Left: no scaling. Right: with column scaling.

Checking contributions of individual poles

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

Theorem (Residue calculus)

$$\mathbf{G}(s) = \sum_{\varphi_j \neq 0} \frac{\widehat{R}_j}{s-\widehat{\lambda}_j} + \sum_{\varphi_j = 0} \frac{R_j}{s-\lambda_j}, \quad \textit{where}$$

$$R_j = \frac{\mathbf{\Phi}_j}{1+\sum_i \varphi_i/(\lambda_j-\lambda_i)}, \quad \textit{for } \varphi_j = 0$$

$$\widehat{R}_j = \frac{\prod_{\varphi_i\neq 0}(\widehat{\lambda}_j-\lambda_i)}{\prod_{i\neq j}(\widehat{\lambda}_j-\widehat{\lambda}_i)} \sum_{i=1}^{r} \frac{\mathbf{\Phi}_i}{\widehat{\lambda}_j-\lambda_i}, \quad \textit{for } \varphi_j \neq 0.$$

$$\max_{\omega\in\mathbb{R}} \sigma_{\max}\Big(\frac{R_j}{\mathrm{i}\omega-\lambda_j}\Big) \leq \max_{\omega\in\mathbb{R}} \frac{\|R_j\|_F}{|\mathrm{i}\omega-\lambda_j|} \leq \frac{\|R_j\|_F}{|\Re(\lambda_j)|}.$$

Test drive of the implementation mimoVF

For the resulting rational approximation $\mathbf{H}_r$, define $\mathbb{S}_r(:,:,i) = \mathbf{H}_r(\xi_i)$, $i = 1, \ldots, \ell$, and the relative LS error as

$$\gamma = \|\mathbb{S} - \mathbb{S}_r\|_F / \|\mathbb{S}\|_F.$$

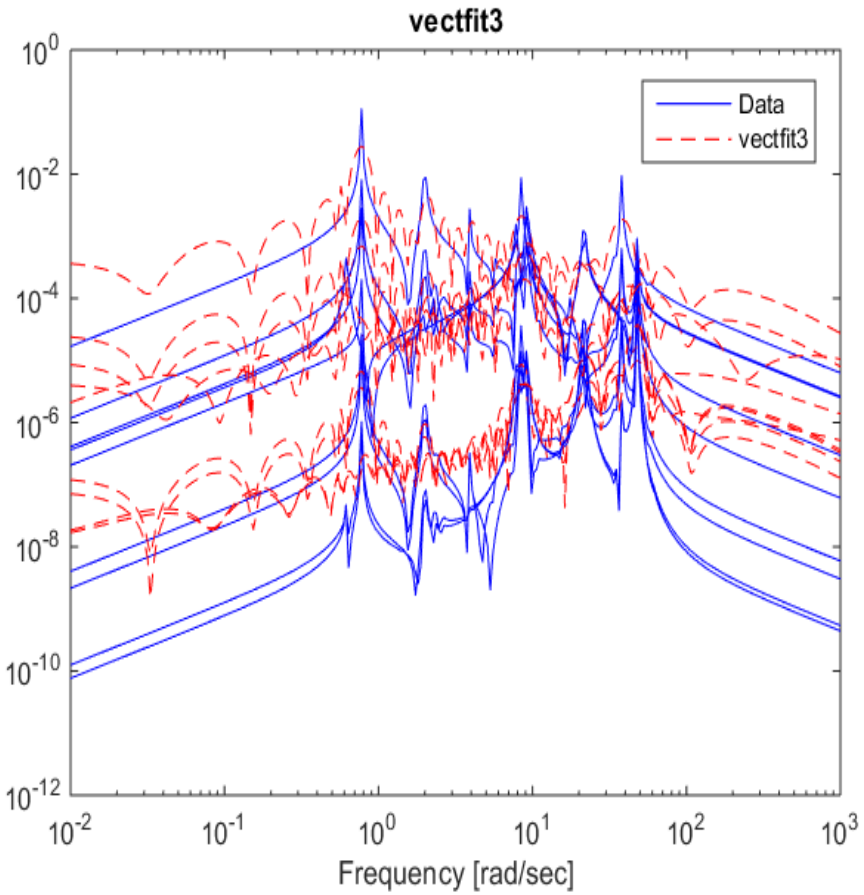
Recall that $\mathbb{S}(:,:,i) = \mathbf{H}(\xi_i) \in \mathbb{C}^{p\times m}$, $i = 1, \ldots, \ell$, contains the original samples that are either measurements, or computed from a state space realization of the underlying LTI dynamical system.

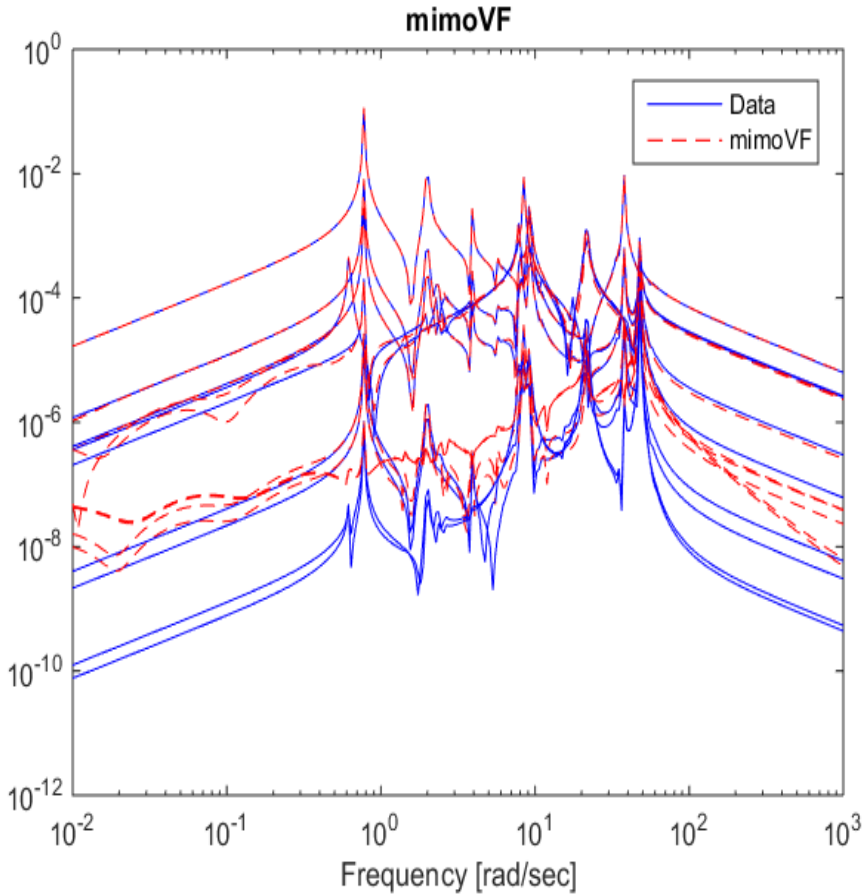
Example (1R module, International Space Station)

Dynamical system of order $n = 270$, with $m = 3$ inputs and $p = 3$ outputs. It is very hard to approximate and is challenging for model order reduction. We take $r = 50$ and use $\ell = 300$ samples.

Compare with vectfit3 (http://www.sintef.no/Projectweb/VECTFIT/)

Stress test: ISS R1, difficult poles, two iterations

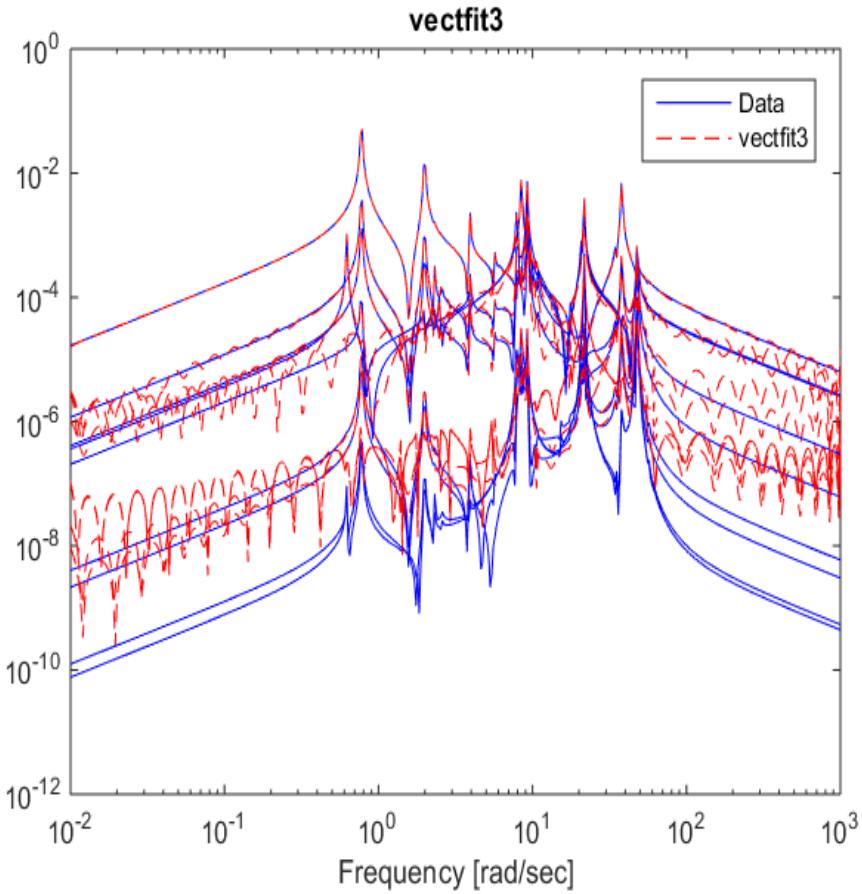


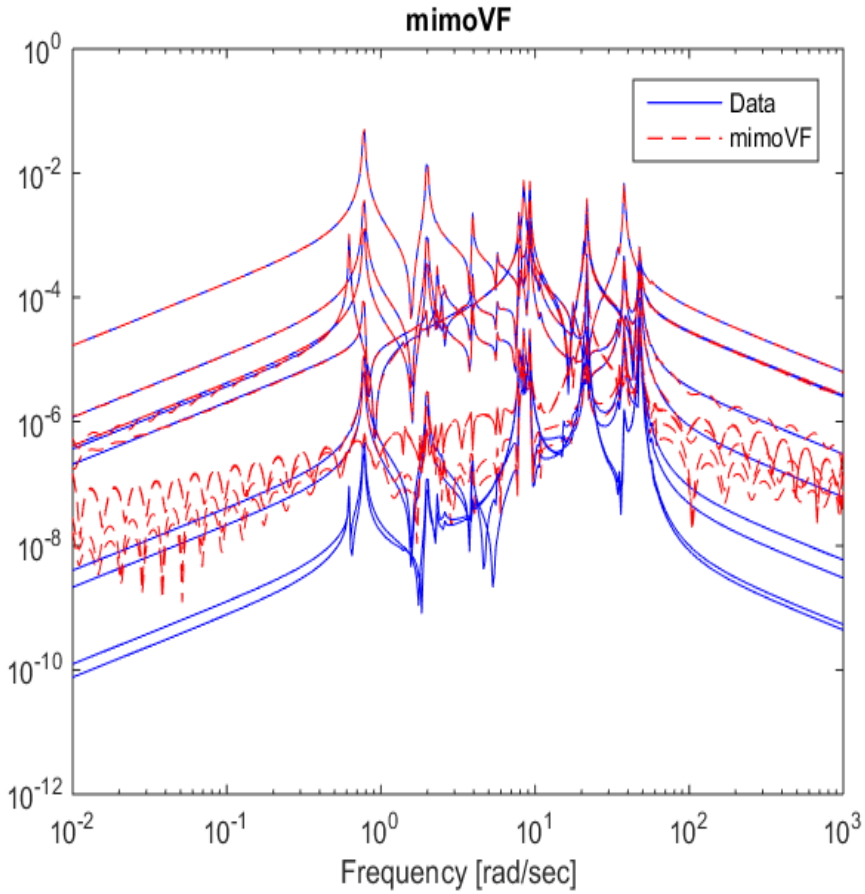


$\gamma(\text{vectfit3}) \approx 14.1,$

$\gamma(\text{mimoVF}) \approx 6.45 \cdot 10^{-3}.$

Stress test: ISS R1, good poles, $r = 100$, one iteration

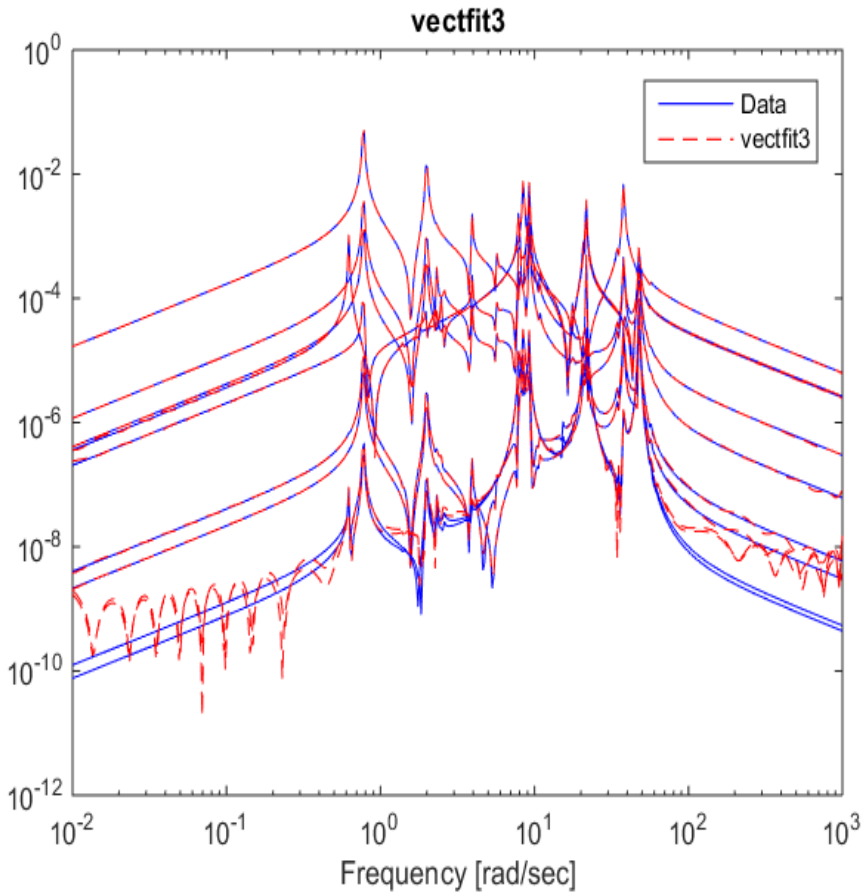


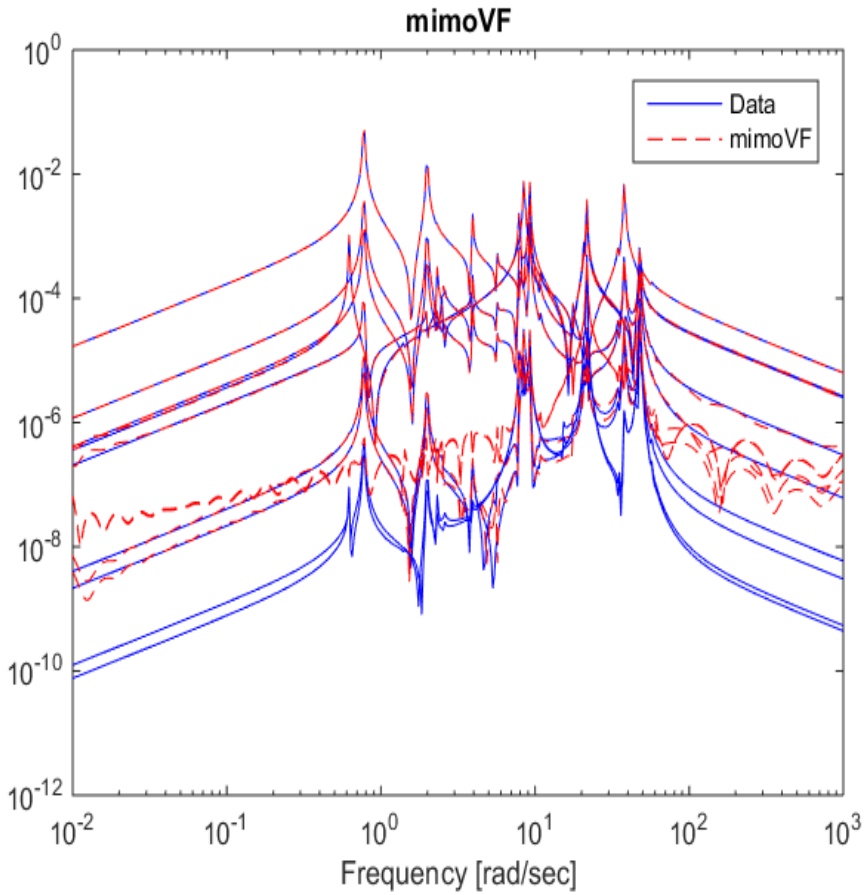


$$\gamma(\text{vectfit3}) \approx 9.47 \cdot 10^{-1},$$

$$\gamma(\text{mimoVF}) \approx 4.90 \cdot 10^{-3}.$$

Stress test: ISS R1, good poles, $r = 100$, one iteration





$\gamma(\text{vectfit3}) \approx 1.77 \cdot 10^{-1}$ (2 iterations), $\gamma(\text{mimoVF}) \approx 4.90 \cdot 10^{-3}$ (1 iter.).

A bigger picture: The Hardy space $\mathcal{H}_{2,+}^{p\times m}$

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 $\sup_{x>0} \int_{\infty}^{\infty} \|\mathbf{M}(x + \dot{\imath}y)\|_F^2 dy < \infty$.

$\mathcal{H}_{2,+}^{p\times m}$, $\langle \cdot, \cdot \rangle$

The space $\mathcal{H}_{2,+}^{p\times m}$is a Hilbert space with the associated inner product and norm defined by

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

By a Fatou theorem, $M(s)$ can be identified with its boundary function $M(\dot{\imath}\omega)$, $\omega \in \mathbb{R}$.

Distretized $\mathcal{H}_2$: Quadrature driven LS on $L_2(\dot{\imath}\mathbb{R})$

Discretized $\mathcal{H}_2$ error as algebraic least squares error

$$\int_{-\infty}^{+\infty} \|\mathbf{H}(\dot{\imath}\omega) - \mathbf{H}_r(\dot{\imath}\omega)\|_F^2 d\omega \approx \sum_{j=1}^{\ell} \rho_j^2 \|\mathbf{H}(\xi_j) - \mathbf{H}_r(\xi_j)\|_F^2 + \rho_+^2 M_+[|\mathbf{H} - \mathbf{H}_r|^2] + \rho_-^2 M_-[|\mathbf{H} - \mathbf{H}_r|^2]$$

An adapted Clenshaw-Curtis scheme (Boyd) :

$$\int_{-\infty}^{\infty} f(\omega) d\omega = \int_0^{\pi} f(L\cot t) \frac{dt}{\sin^2 t} \approx \sum_{j=0}^{\ell+1} w_j f(L\cot t_j), \quad t_j = \frac{j\pi}{\ell+1};$$

$$w_j = \begin{cases} \dfrac{L\pi}{(\ell+1)\sin^2 t_j}, & 0 < j < \ell+1 \\ \dfrac{L\pi}{(2\ell+2)\sin^2 t_j}, & j = 0,\ j = \ell+1 \end{cases}$$

Quad VF

Adapted to our setting, the Boyd/Clenshaw-Curtis (B/CC) formula is

$$\|\mathbf{H}(s)\|_{\mathcal{H}_2}^2 = \int_{-\infty}^{+\infty} |\mathbf{H}(\dot{\imath}\omega)|^2 \, d\omega = \int_0^{\pi} \frac{L}{\sin^2 t} |\mathbf{H}(\dot{\imath} L \cot t)|^2 \, dt$$

$$\approx \sum_{j=1}^{\ell} \frac{L\pi}{(\ell+1)\sin^2 t_j} |\mathbf{H}(\dot{\imath} L \cot t_j)|^2 + \frac{\pi}{2L(\ell+1)} \left(|M_+[\mathbf{H}]|^2 + |M_-[\mathbf{H}]|^2 \right).$$

$L > 0$ is a freely chosen scaling parameter, $t_j = \frac{j\pi}{\ell+1}$, for $j = 1, \ldots, \ell$, and

$$M_+[\mathbf{H}] = \lim_{\omega \to \infty} \dot{\imath}\omega \, \mathbf{H}(\dot{\imath}\omega) = \lim_{t \to 0^+} \frac{\mathbf{H}(\dot{\imath} L \cot t)}{\sin(t)} \cdot \dot{\imath} L$$

$$M_-[\mathbf{H}] = \lim_{\omega \to -\infty} \dot{\imath}\omega \, \mathbf{H}(\dot{\imath}\omega) = \lim_{t \to \pi^-} \frac{\mathbf{H}(\dot{\imath} L \cot t)}{\sin(t)} \cdot \dot{\imath} L \qquad (2.5)$$

For example, if $\mathbf{H}(s)$ is a strictly proper transfer function with realization, $\mathbf{H}(s) = \mathbf{C}(sI - \mathbf{F})^{-1}\mathbf{B}$, then $M_+[\mathbf{H}] = M_-[\mathbf{H}] = \mathbf{CB}$.

Quad VF: new weighted LS problem

$$\| \Delta_\rho \left(\mathcal{A}(\boldsymbol{\lambda}^{(k+1)}) x^{(k+1)} - h \right) \|_2 \to \min, \quad k = 0, 1, 2, \ldots,$$

where $x^{(k+1)} = \left(\begin{array}{cccccccc} \phi_1^{(k+1)} & \phi_2^{(k+1)} & \cdots & \phi_r^{(k+1)} & \varphi_1^{(k+1)} & \varphi_2^{(k+1)} & \cdots & \varphi_r^{(k+1)} \end{array} \right)^T$,

$$\mathcal{A}(\boldsymbol{\lambda}) = \begin{pmatrix} \frac{1}{\xi_1-\lambda_1} & \frac{1}{\xi_1-\lambda_2} & \cdots & \frac{1}{\xi_1-\lambda_r} & \frac{-H(\xi_1)}{\xi_1-\lambda_1} & \frac{-H(\xi_1)}{\xi_1-\lambda_2} & \cdots & \frac{-H(\xi_1)}{\xi_1-\lambda_r} \\ \frac{1}{\xi_2-\lambda_1} & \frac{1}{\xi_2-\lambda_2} & \cdots & \frac{1}{\xi_2-\lambda_r} & \frac{-H(\xi_2)}{\xi_2-\lambda_1} & \frac{-H(\xi_2)}{\xi_2-\lambda_2} & \cdots & \frac{-H(\xi_2)}{\xi_2-\lambda_r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\xi_{\ell-1}-\lambda_1} & \frac{1}{\xi_{\ell-1}-\lambda_2} & \cdots & \frac{1}{\xi_{\ell-1}-\lambda_r} & \frac{-H(\xi_{\ell-1})}{\xi_{\ell-1}-\lambda_1} & \frac{-H(\xi_{\ell-1})}{\xi_{\ell-1}-\lambda_2} & \cdots & \frac{-H(\xi_{\ell-1})}{\xi_{\ell-1}-\lambda_r} \\ \frac{1}{\xi_\ell-\lambda_1} & \frac{1}{\xi_\ell-\lambda_2} & \cdots & \frac{1}{\xi_\ell-\lambda_r} & \frac{-H(\xi_\ell)}{\xi_\ell-\lambda_1} & \frac{-H(\xi_\ell)}{\xi_\ell-\lambda_2} & \cdots & \frac{-H(\xi_\ell)}{\xi_\ell-\lambda_r} \\ 1 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \end{pmatrix} h = \begin{pmatrix} H(\xi_1) \\ H(\xi_2) \\ \vdots \\ H(\xi_{\ell-1}) \\ H(\xi_\ell) \\ M_+[H] \end{pmatrix}$$

and $\Delta_\rho = \operatorname{diag}(\rho_1, \rho_2, \ldots, \rho_\ell, \rho_+)$ with nodes $\xi_j = \imath L \cot\left(\frac{j\pi}{\ell+1}\right)$ and

weights $\rho_j = \csc\left(\frac{j\pi}{\ell+1}\right) \sqrt{\frac{L\pi}{(\ell+1)}}$ for $j = 1, \ldots, \ell$ and $\rho_+ = \sqrt{\frac{\pi}{L(\ell+1)}}$

Example: Heat model, QuadVF vs VF

- We take $\ell = 20$ samples; sampling points contained in $\dot{\imath}[2.7705 \times 10^{-2}, 2.4527]$;
- construct order $r = 4$ rational approximants
- The resulting relative $\mathcal{H}_2$ error norms are 8.4776×10^{-1} for VF and 6.9326×10^{-3} for QuadVF.
- The numbers for the relative $\mathcal{H}_\infty$ error norms were even more revealing: 1.6392 for VF and 6.7765×10^{-4} for the QuadVF.
- The algebraic LS relative residual norm for VF is 1.8943×10^{-3}, representing a very accurate solution to the discrete LS problem; for QuadVF, the relative residual norm is 3.5430×10^{-4},

VF does a great job in minimizing the least-squares error over the given samples; however the samples are local in nature and do not reflect the global $\mathcal{H}_2$ and/or $\mathcal{H}_\infty$ behavior. By choosing the sampling nodes from an appropriate quadrature rule, the discrete error that is minimized becomes a much better approximation to the true $\mathcal{H}_2$ error, leading ultimately to a better rational approximation.

Example: Heat model, QuadVF vs VF; $r = 2$

We use the Heat Model and construct order $r = 2$ rational approximants using QuadVF and IRKA. Let H, H_1, H_2 denote, respectively, the full-order model, the reduced model by IRKA and the reduced model by QuadVF. In Table below, we list the relative $\mathcal{H}_2$ distances as ℓ increases:

ℓ	$\frac{\|H_1 - H_2\|_{\mathcal{H}_2}}{\|H_1\|_{\mathcal{H}_2}}$	$\frac{\|H - H_1\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	$\frac{\|H - H_2\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$
10	1.1919×10^{-2}	3.9483×10^{-2}	4.1348×10^{-2}
100	3.8795×10^{-3}	3.9483×10^{-2}	3.9681×10^{-2}
1000	1.0239×10^{-3}	3.9483×10^{-2}	3.9497×10^{-2}
5000	5.2313×10^{-4}	3.9483×10^{-2}	3.9487×10^{-2}
15000	4.4926×10^{-4}	3.9483×10^{-2}	3.9486×10^{-2}

Table: Relative $\mathcal{H}_2$ distances vs ℓ

With $\ell = 15000$ VF has $\mathcal{H}_2$ relative error 0.98. This illustrates the value of sampling guided by a quadrature rule.

Controlling the McMillan Degree

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 $\mathbf{H}(s)$.

$\mathbf{H}_r$ can have McMillan degree as high as $r \cdot \min(p, m)$

The final approximant (assuming simple poles)

$$\mathbf{H}_r(s) = \sum_{j=1}^{r} \frac{\mathbf{\Phi}_j}{s - \lambda_j} = \begin{pmatrix} \mathbb{I}_p & \cdots & \mathbb{I}_p \end{pmatrix} \begin{pmatrix} \frac{\mathbb{I}_p}{s-\lambda_1} & & \\ & \ddots & \\ & & \frac{\mathbb{I}_p}{s-\lambda_r} \end{pmatrix} \begin{pmatrix} \mathbf{\Phi}_1 \\ \vdots \\ \mathbf{\Phi}_r \end{pmatrix} \tag{3.1}$$

has McMillan degree $\deg(\mathbf{H}_r(s)) = \sum_{j=1}^{r} \operatorname{rank}(\mathbf{\Phi}_j) \leq r \min(p, m)$.

Gustavsen and Semlyen proposed determining numerical ranks of the $\mathbf{\Phi}_j$'s and truncating them using the SVD. They also proposed a Gauss–Newton correction, without giving details.

Our first proposal: rank-one truncated residues + ALS correction

$$\widehat{\mathbf{H}}_r(s) = \sum_{j=1}^{r} \frac{c_j b_j^T}{s-\lambda_j} \equiv C(s\mathbb{I}-\Lambda)^{-1}B^T, \quad \begin{array}{l} C = \begin{pmatrix} c_1 & \dots & c_r \end{pmatrix} \\ B = \begin{pmatrix} b_1 & \dots & b_r \end{pmatrix}. \end{array}$$

$\Lambda = \mathrm{diag}(\lambda_j)_{j=1}^r$ Choose C and B so that $\widehat{\mathbf{H}}_r(s)$ satisfies

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

Initialize with $c_j b_j^T \approx \mathbf{\Phi}_j$, best rank-one approximation, and then use alternating least squares. In each step, solution for B and C based on matrix equations and computation rich in BLAS 3 operations.

Correction of C

kth column ($k = 1, \ldots, m$) of the residual matrix as

$$\mathcal{M} \begin{pmatrix} (b_1)_k \mathbb{I} & & \\ & \ddots & \\ & & (b_r)_k \mathbb{I} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} \mathbf{H}(\xi_1) e_k \\ \vdots \\ \mathbf{H}(\xi_\ell) e_k \end{pmatrix}$$

$$= \mathcal{M}(\mathrm{diag}(B(k,:)) \otimes \mathbb{I}_p) \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} \mathbf{H}(\xi_1) e_k \\ \vdots \\ \mathbf{H}(\xi_\ell) e_k \end{pmatrix}.$$

Set $\Delta_i = \mathrm{diag}(B(i,:)), \quad i = 1, \ldots, m$
Impose conjugation symmetry constraints: c_j is real if λ_j is real and $c_k = \overline{c_j}$, if $\lambda_k = \overline{\lambda_j}$.
Stack all columns in a long vector to obtain the problem of minimizing the following residual:

Correction of C

$$\left\| \left(\begin{array}{c} \mathcal{M}(\Delta_1 \otimes \mathbb{I}_p) \\ \hline \vdots \\ \hline \mathcal{M}(\Delta_m \otimes \mathbb{I}_p) \end{array}\right) \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \left(\begin{array}{c} \mathbf{H}(\xi_1)e_1 \\ \vdots \\ \mathbf{H}(\xi_\ell)e_1 \\ \hline \vdots \\ \hline \mathbf{H}(\xi_1)e_m \\ \vdots \\ \mathbf{H}(\xi_\ell)e_m \end{array}\right) \right\|_F, \quad \Delta_i = \mathrm{diag}(B(i,:)), \quad i = 1, \ldots, m \tag{3.2}$$

Recall, $\mathcal{M} = \mathscr{C} \otimes \mathbb{I}_p$. Let $\mathscr{C} = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = \widehat{Q} R$ be the QR factorization, where $Q = \begin{pmatrix} \widehat{Q} & \check{Q} \end{pmatrix}$, $\widehat{Q} = Q(:, 1:r)$. Then $\mathcal{M} = (Q \otimes \mathbb{I}_p)(\begin{pmatrix} R \\ 0 \end{pmatrix} \otimes \mathbb{I}_p)$ is the QR factorization of $\mathcal{M}$. Multiplying the blocks in the residual (3.2) by $(Q^* \otimes \mathbb{I}_p)$ and using the block-partitioned structure of Q, we obtain an equivalent LS problem:

Correction of C

$$\left\| \left(\begin{array}{c} (R\otimes \mathbb{I}_p)(\Delta_1\otimes \mathbb{I}_p) \\ 0 \\ \hline \vdots \\ \hline (R\otimes \mathbb{I}_p)(\Delta_m\otimes \mathbb{I}_p) \\ 0 \end{array}\right) \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \left(\begin{array}{c} (\widehat{Q}^*\otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,1,:)) \\ (\check{Q}^*\otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,1,:)) \\ \hline \vdots \\ \hline (\widehat{Q}^*\otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,m,:)) \\ (\check{Q}^*\otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,m,:)) \end{array}\right) \right\|_F \longrightarrow \min . \tag{3.3}$$

The blocks $(\check{Q}^*\otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,i,:))$, $i=1,\ldots,m$, in the right-hand side constitute a part of the residual that cannot be influenced with any choice of the c_js 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 $\mathscr{C} = \widehat{Q}R$ 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

Correction of C

$$\left\| \left[\begin{pmatrix} R\Delta_1 \\ \hline \vdots \\ \hline R\Delta_m \end{pmatrix} \otimes \mathbb{I}_p \right] \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} - \begin{pmatrix} (\widehat{Q}^* \otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,1,:)) \\ \hline \vdots \\ \hline (\widehat{Q}^* \otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,m,:)) \end{pmatrix} \right\|_F \longrightarrow \min . \tag{3.4}$$

To solve (3.4) we compute the QR factorizations

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

and, using the partition $U = \begin{pmatrix} \widehat{U} & \check{U} \end{pmatrix}$, we reduce the problem to solving

$$(T \otimes \mathbb{I}_p) \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix} = (\widehat{U}^* \otimes \mathbb{I}_p) \begin{pmatrix} (\widehat{Q}^* \otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,1,:)) \\ \hline \vdots \\ \hline (\widehat{Q}^* \otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,m,:)) \end{pmatrix} . \tag{3.5}$$

Only the thin QRF $R_{\boxminus} = \widehat{U}T$ is needed.

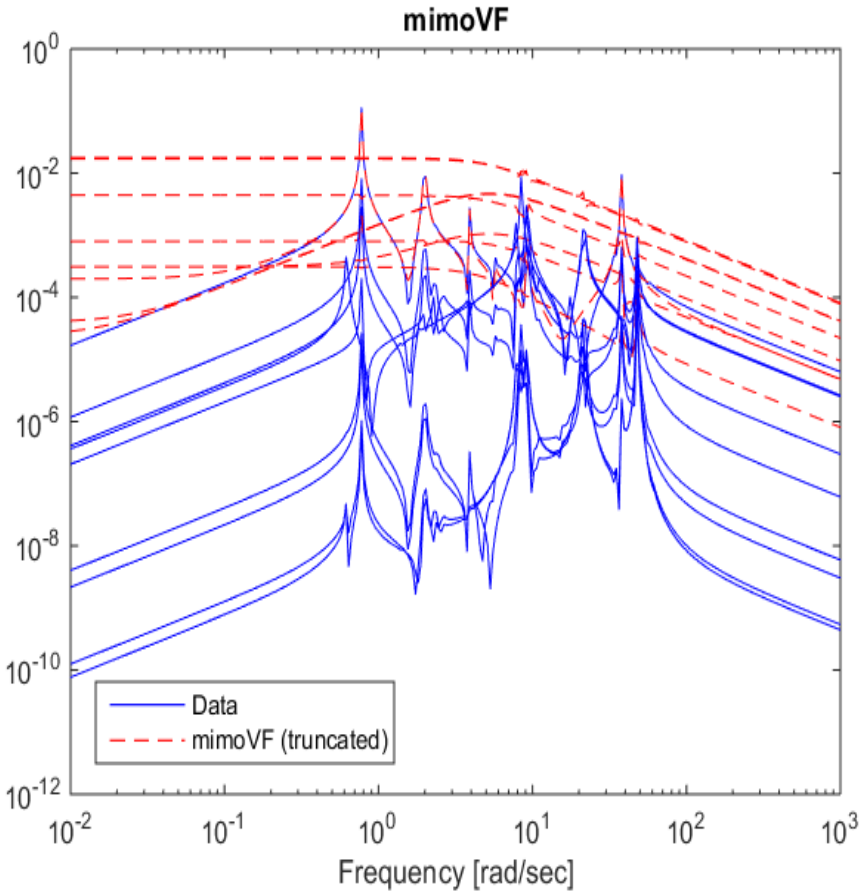
Correction of C

Folding the unknowns back into the structure of C we obtain, using that $(\widehat{Q}^* \otimes \mathbb{I}_p)\mathrm{vec}(\mathbb{S}(:,i,:)) = \mathrm{vec}(\mathbb{S}(:,i,:)\widehat{Q}^{*T})$,

$$\begin{aligned}
\mathrm{vec}(CT^T) &= (\widehat{U}^* \otimes \mathbb{I}_p) \left(\begin{array}{c} \mathrm{vec}(\mathbb{S}(:,1,:)\widehat{Q}^{*T}) \\ \hline \vdots \\ \hline \mathrm{vec}(\mathbb{S}(:,m,:)\widehat{Q}^{*T}) \end{array}\right) && (3.6)\\
&= (\widehat{U}^* \otimes \mathbb{I}_p)\mathrm{vec}(\left(\mathbb{S}(:,1,:)\widehat{Q}^{*T} \quad \ldots \quad \mathbb{S}(:,m,:)\widehat{Q}^{*T}\right)) \\
&= \mathrm{vec}(\left(\mathbb{S}(:,1,:)\widehat{Q}^{*T} \quad \ldots \quad \mathbb{S}(:,m,:)\widehat{Q}^{*T}\right)\widehat{U}^{*T}). && (3.7)
\end{aligned}$$

As an alternative to solving (3.5), C can be computed efficiently as the solution of a triangular matrix equation. The formula $C = \left(\mathbb{S}(:,1,:)\widehat{Q}^{*T} \quad \ldots \quad \mathbb{S}(:,m,:)\widehat{Q}^{*T}\right)\widehat{U}^{*T}T^{-T}$ 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.

Rank one truncation + ALS: the ISS R1 example



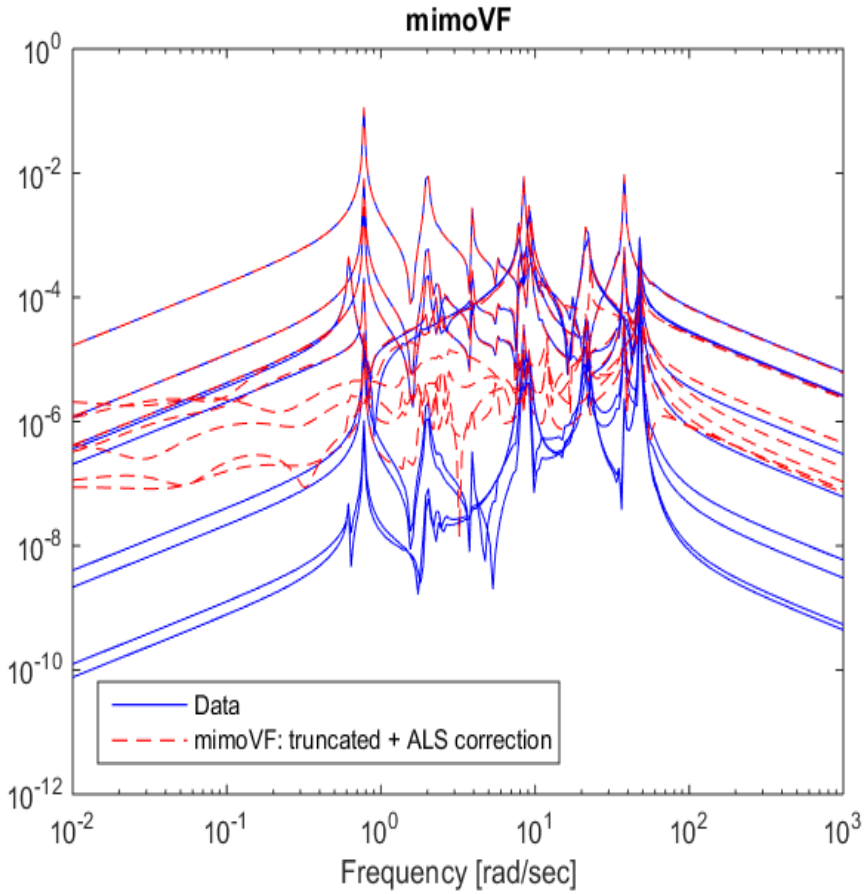


Illustration of the truncation of mimoVF and the ALS correction. Only one ALS iteration is used. The relative $\mathcal{H}_2$ error of $\widehat{\mathbf{H}}_r$ is $\approx 1.50e-01$.

Our second proposal: Order reduction in $\mathcal{H}_2$, $\mathcal{H}_\infty$ setting

$$\mathbf{H}_r(s) = \sum_{j=1}^{r} \frac{\mathbf{\Phi}_j}{s - \lambda_j} \tag{3.8}$$

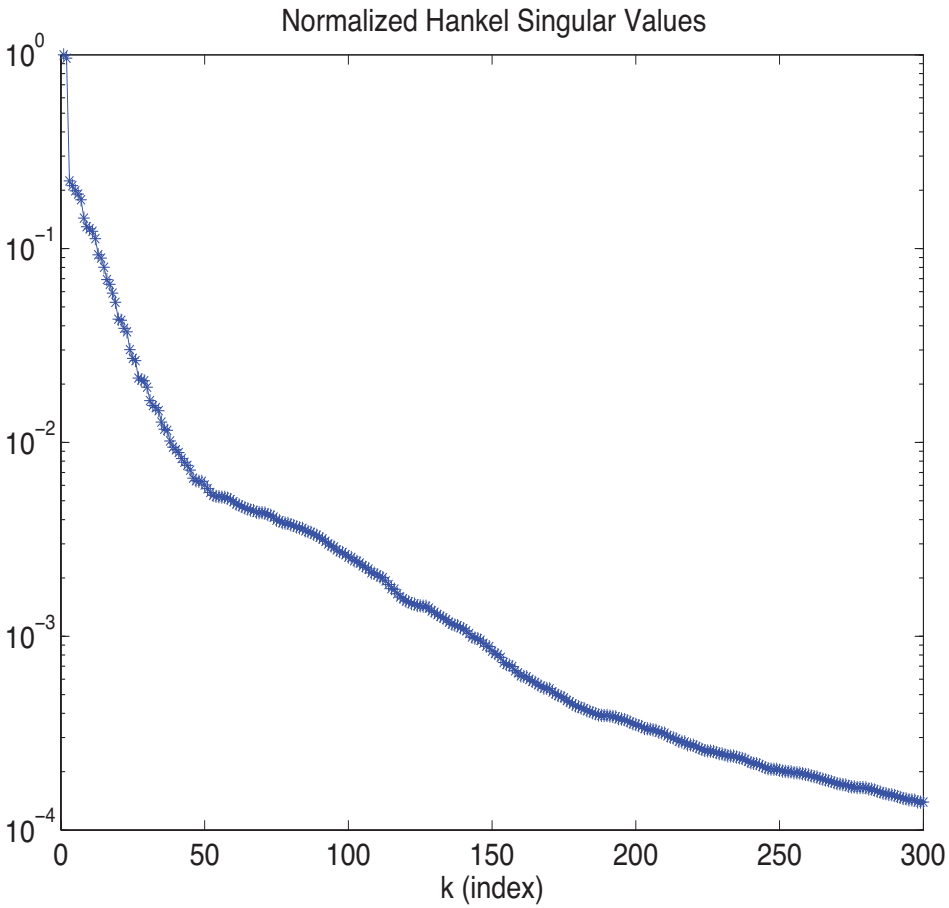
$\|\mathbf{H}\|_{\mathcal{H}_\infty} = \sup_{\omega\in\mathbb{R}} \|\mathbf{H}(\imath\omega)\|_2 \twoheadrightarrow$ Balanced Truncation

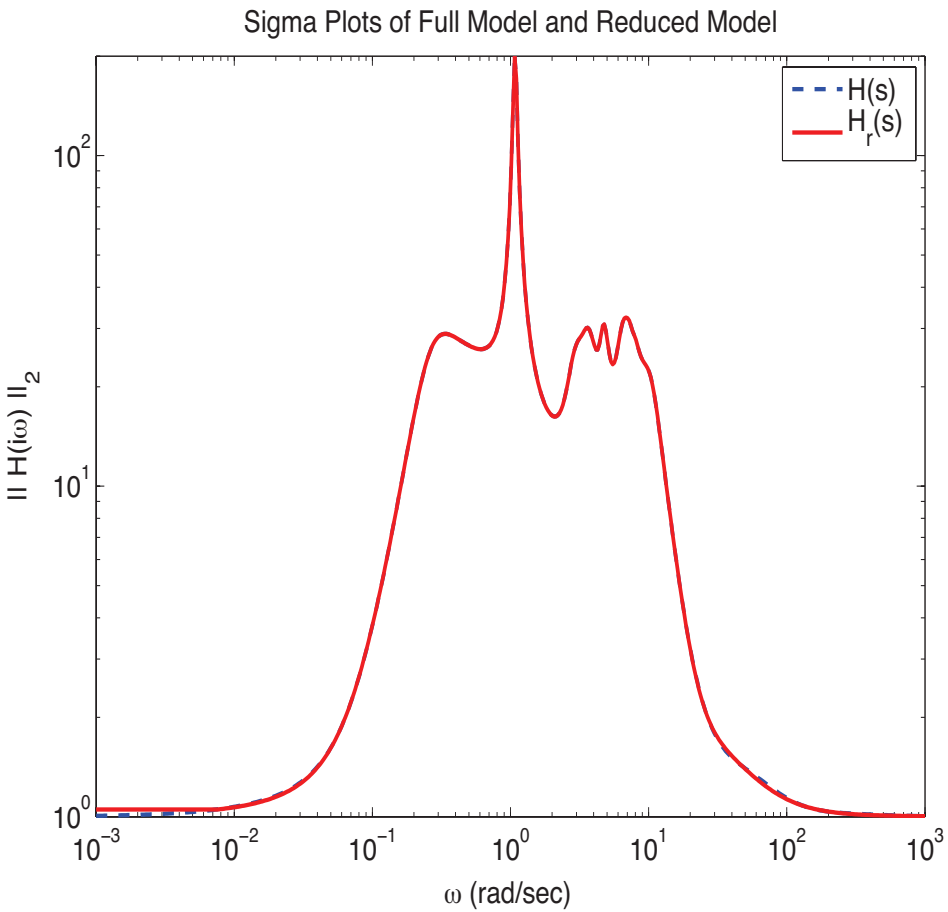
Use balanced truncation. Solve the Lyapunov equations efficiently using simple state space realization of (3.8). (Recall Monday lectures by Bart.)

$\|\mathbf{H}\|_{\mathcal{H}_2} \twoheadrightarrow$ realization independent IRKA (Beattie and Gugercin)

- Gugercin, Antoulas, Beattie: IRKA – Locally $\mathcal{H}_2$-optimal order–r approximation $\widehat{\mathbf{H}}_r$ of $\mathbf{H}_r$. (Recall Monday/Tuesday lectures by Serkan.)
- Beattie, Gugercin: realization independent IRKA – only requires function evaluations, which is simple using (3.8).

Interconect power system; $m = p = 46$, $n = 13250$





With $r = 80$ and $\ell = 200$ samples $\mathcal{H}_2$ error $1.51 \cdot 10^{-1}$.
IRKA $\mathcal{H}_2$ error $1.13 \cdot 10^{-1}$.

Tangential LS fitting of MIMO data

- Given a MIMO system $\mathbf{H}(s)$ with m-inputs and p-outputs, assume that the following data is available:

$$\mathbf{H}(\xi_i)\mathbf{r}_i \quad \text{and} \quad \boldsymbol{\ell}_i^T\mathbf{H}(\xi_i) \quad \text{for} \quad i = 1, \ldots, \ell$$

  where $\xi_i \in \mathbb{C}$ are the sampling points, $\mathbf{r}_i \in \mathbb{C}^m$ are the right-tangential and $\boldsymbol{\ell}_i \in \mathbb{C}^p$ are the left-tangential directions.
- Note that we are not assuming access to full $\mathbf{H}(\xi_i)$, instead we are assuming that $\mathbf{H}(\xi_i)$ is available only along certain directions. Even for the cases when the full-matrix data $\mathbf{H}(\xi_i)$ is available, one can choose the tangential directions separate from the measurement stage. For example, one common choice is to pick the leading right and left singular vectors of $\mathbf{H}(\xi_i)$ as the right and left directions, respectively.

Tangential LS fitting of MIMO data

Problem formulation

$$\min_{\mathbf{H}_r \in \mathcal{R}_r} \left\{ \sum_{i=1}^{\ell} \|\mathbf{H}(\xi_i)\mathbf{r}_i - \mathbf{H}_r(\xi_i)\mathbf{r}_i\|_2^2 + \sum_{i=1}^{\ell} \left\| \ell_i^T \mathbf{H}(\xi_i) - \ell_i^T \mathbf{H}_r(\xi_i) \right\|_2^2 \right\}$$

We take $\mathbf{H}_r(s)$ to be of the form

$$\mathbf{H}_r(s) = \frac{\displaystyle\sum_{j=1}^{r} \frac{c_j b_j^T}{s - \lambda_j}}{\displaystyle\sum_{j=1}^{r} \frac{\beta_j}{s - \lambda_j} + 1} = \frac{\mathbf{N}(s)}{d(s)}$$

where $b_j \in \mathbb{C}^m$, $c_j \in \mathbb{C}^p$, $\beta_j, \gamma_j \in \mathbb{C}$, for $j = 1, \ldots, r$, are the variables to be determined. Note that we are enforcing the rank-1 residue structure directly at the beginning; thus the resulting approximation is truly a rational function of degree-r.

Tangential crisscross ALS

To ease the notation, we drop the iteration index k and use the following convention: the quantities marked with tilde such as $\widetilde{c}_j$, $\widetilde{b}_j$, $\widetilde{\beta}_j$, $\widetilde{\lambda}_j$ correspond to the index $k+1$, while c_j, b_j are from the kth iteration. Let $\widetilde{B} = \begin{pmatrix} \widetilde{b}_1 & \dots & \widetilde{b}_r \end{pmatrix}$, $\widetilde{C} = \begin{pmatrix} \widetilde{c}_1 & \dots & \widetilde{c}_r \end{pmatrix}$, $\widetilde{\mathbf{b}} = \mathrm{vec}(\widetilde{B})$, $\widetilde{\mathbf{c}} = \mathrm{vec}(\widetilde{C})$.

$$\widetilde{\epsilon}^2 = \sum_{i=1}^{\ell} \tau_i \left\| \sum_{j=1}^{r} \frac{\widetilde{c}_j b_j^T}{\xi_i - \widetilde{\lambda}_j} \mathbf{r}_i - \sum_{j=1}^{r} \frac{\widetilde{\beta}_j}{\xi_i - \widetilde{\lambda}_j} \mathbf{H}(\xi_i)\mathbf{r}_i - \mathbf{H}(\xi_i)\mathbf{r}_i \right\|_2^2$$

$$\widetilde{\delta}^2 = \sum_{i=1}^{\ell} \rho_i \left\| \sum_{j=1}^{r} \boldsymbol{\ell}_i^T \frac{c_j \widetilde{b}_j^T}{\xi_i - \widetilde{\lambda}_j} - \sum_{j=1}^{r} \frac{\widetilde{\beta}_j}{\xi_i - \widetilde{\lambda}_j} \boldsymbol{\ell}_i^T \mathbf{H}(\xi_i) - \boldsymbol{\ell}_i^T \mathbf{H}(\xi_i) \right\|_2^2$$

One iteration

Define $\mathscr{C}_{c\ell} \in \mathbb{C}^{\ell\times r}$, $\mathscr{C}_{b\mathbf{r}} \in \mathbb{C}^{\ell\times r}$, $\mathscr{C}_{\ell} \in \mathbb{C}^{\ell m\times r}$, $\mathscr{C}_{\mathbf{r}} \in \mathbb{C}^{\ell p\times r}$ as

$$(\mathscr{C}_{c\ell})_{ij} = \frac{c_j^T \ell_i}{\xi_i - \widetilde{\lambda}_j}, \quad (\mathscr{C}_{\ell})_{ij} = \frac{\mathbf{H}(\xi_i)^T \ell_i}{\xi_i - \widetilde{\lambda}_j}, \quad i = 1,\ldots,\ell,\ j = 1,\ldots,r.$$

$$(\mathscr{C}_{b\mathbf{r}})_{ij} = \frac{b_j^T \mathbf{r}_i}{\xi_i - \widetilde{\lambda}_j}, \quad (\mathscr{C}_{\mathbf{r}})_{ij} = \frac{\mathbf{H}(\xi_i)\mathbf{r}_i}{\xi_i - \widetilde{\lambda}_j}, \quad i = 1,\ldots,\ell,\ j = 1,\ldots,r.$$

and $\mathbf{h}_{\mathbf{r}} = \begin{pmatrix} \mathbf{H}(\xi_1)\mathbf{r}_1 \\ \vdots \\ \mathbf{H}(\xi_\ell)\mathbf{r}_\ell \end{pmatrix}$, $\mathbf{h}_{\ell} = \begin{pmatrix} \mathbf{H}(\xi_1)^T\ell_1 \\ \vdots \\ \mathbf{H}(\xi_\ell)^T\ell_\ell \end{pmatrix}$ to obtain

$$\left\| \left(\begin{array}{c|c|c} \mathscr{C}_{b\mathbf{r}} \otimes \mathbb{I}_p & \mathbf{O} & -\mathscr{C}_{\mathbf{r}} \\ \hline \mathbf{O} & \mathscr{C}_{c\ell} \otimes \mathbb{I}_m & -\mathscr{C}_{\ell} \end{array} \right) \left(\begin{array}{c} \widetilde{\mathbf{c}} \\ \hline \widetilde{\mathbf{b}} \\ \hline \widetilde{\boldsymbol{\beta}} \end{array} \right) - \left(\begin{array}{c} \mathbf{h}_{\mathbf{r}} \\ \hline \mathbf{h}_{\ell} \end{array} \right) \right\|_2^2 \longrightarrow \min$$

QR factorizations and Kronecker structure

Note that

- The number of equations is $(m+p)\cdot\ell$ (in MIMO VF $m\cdot p\cdot\ell$)
- The number of unknowns is $(m+p)\cdot r$ (in MIMO VF $m\cdot p\cdot r$)

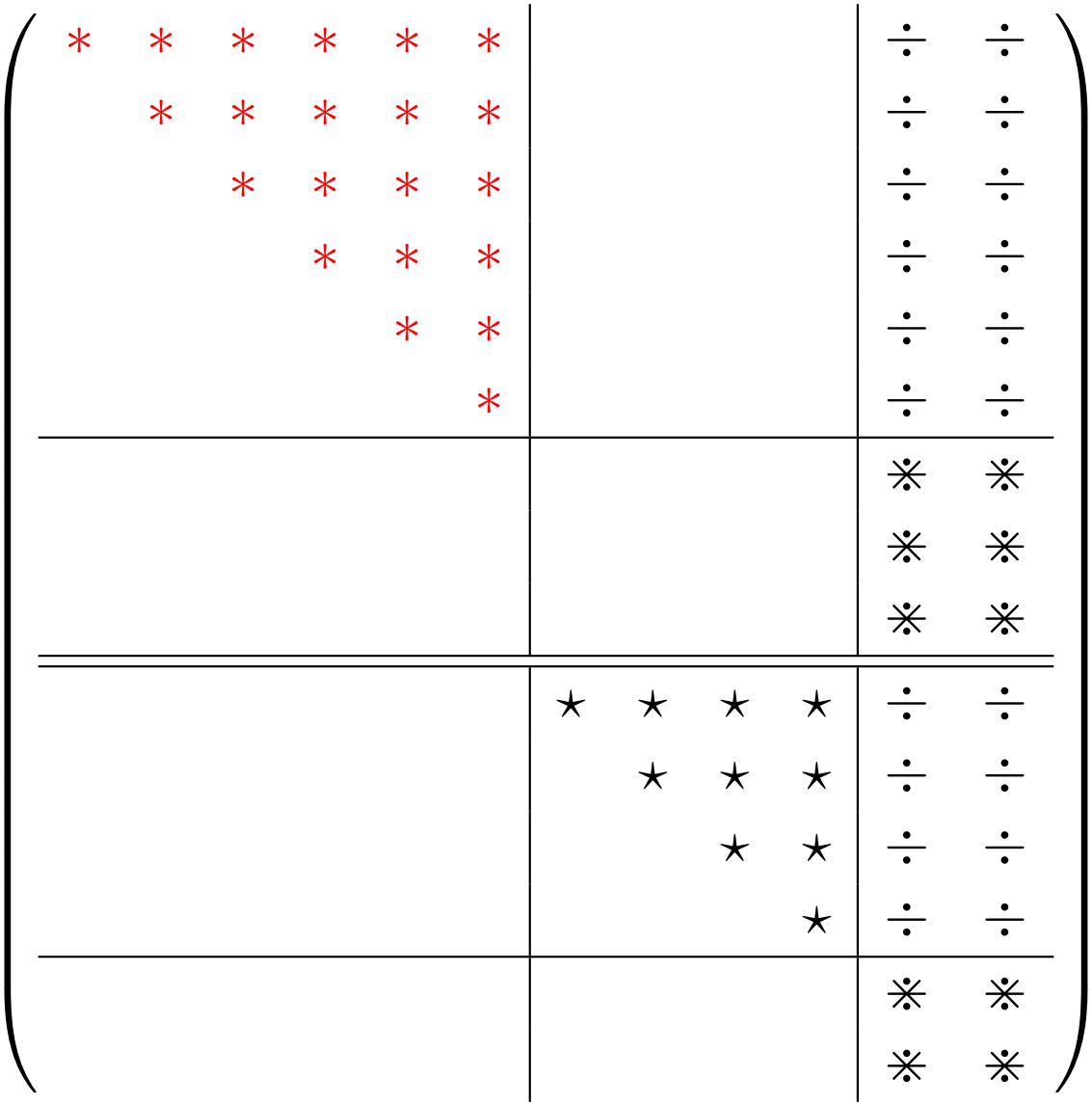
Consider the QR factorizations $\mathscr{C}_{\mathbf{br}} = Q_{\mathbf{br}} \left(\begin{smallmatrix} R_{\mathbf{br}} \\ 0 \end{smallmatrix}\right)$, $\mathscr{C}_{\mathbf{c\ell}} = Q_{\mathbf{c\ell}} \left(\begin{smallmatrix} R_{\mathbf{c\ell}} \\ 0 \end{smallmatrix}\right)$. Then we have the QR factorization

$$\mathscr{C}_{\mathbf{br}} \otimes \mathbb{I}_p = (Q_{\mathbf{br}} \otimes \mathbb{I}_p)(\left(\begin{smallmatrix} R_{\mathbf{br}} \\ 0 \end{smallmatrix}\right) \otimes \mathbb{I}_p), \quad \mathscr{C}_{\mathbf{c\ell}} \otimes \mathbb{I}_m = (Q_{\mathbf{c\ell}} \otimes \mathbb{I}_m)(\left(\begin{smallmatrix} R_{\mathbf{c\ell}} \\ 0 \end{smallmatrix}\right) \otimes \mathbb{I}_m)$$

$$\left(\begin{array}{c|c|c} \left(\begin{smallmatrix} R_{\mathbf{br}} \\ 0 \end{smallmatrix}\right) \otimes \mathbb{I}_p & \mathbf{O} & -(Q_{\mathbf{br}}^* \otimes \mathbb{I}_p)\mathscr{C}_{\mathbf{r}} \\ \hline \mathbf{O} & \left(\begin{smallmatrix} R_{\mathbf{c\ell}} \\ 0 \end{smallmatrix}\right) \otimes \mathbb{I}_m & -(Q_{\mathbf{c\ell}}^* \otimes \mathbb{I}_m)\mathscr{C}_{\ell} \end{array}\right)$$

Only two $\ell \times r$ QR factorizations. Can pivot for numerical robustness. (Similar discussions as before!)

QR factorizations and Kronecker structure



Lab Exercise:

For test data: google "NICONET benchmarks" or go directly to http://slicot.org/20-site/126-benchmark-examples-for-model-reduction

1. Implement Vector Fitting SISO. Use SISO examples from NICONET benchmarks as follows:
   - Pick an example (e.g. clamped beam) and use it to generate samples.
   - Construct a VF aproximation. Try different orders r, different sampling frequencies (equidistant, log spaced, quadrature based).
2. Implement Vector Fitting MIMO. Use MIMO examples from NICONET benchmarks as follows:
   - Pick an example and use it to generate samples (e.g. ISS).
   - Construct a VF aproximation. Try different orders r, different sampling frequencies (equidistant, log spaced).
3. Combine MIMO VF with MOR methods:
   - Alternating Least Squares for new rank-one residues
   - Balanced trunctation
   - MIMO IRKA

In all cases, use the known underlying model to compute errors, compare Bode plots.

References


C. Beattie and S. Gugercin, *Realization–independent H_2 approximation*, in Proc. of the 51st IEEE Conf. on Decision & Control, IEEE, 2012, pp. 4953–4958.


D. Deschrijver, B. Gustavsen, and T. Dhaene, *Advancements in iterative methods for rational approximation in the frequency domain*, Power Delivery, IEEE Transactions on, 22 (2007), pp. 1633–1642.


Z. Drmač, S. Gugercin, and C. Beattie, *Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation*, SIAM J. Sci. Comp. (to appear), (2015).


Z. Drmač, S. Gugercin, and C. Beattie, *Vector Fitting for matrix-valued rational approximation*, SIAM J. Sci. Comp. (submitted), (2015).


S. Gugercin, A. C. Antoulas, and C. Beattie, *$\mathcal{H}_2$ model reduction for large-scale linear dynamical systems*, SIAM SIMAX, 30 (2008), pp. 609–638.


B. Gustavsen and A. Semlyen, *Rational approximation of frequency domain responses by vector fitting*, IEEE Trans. on Power Del., 14 (1999), pp. 1052–1061.


———, *A robust approach for system identification in the frequency domain*, IEEE Transactions on Power Delivery, 19 (2004), pp. 1167–1173.