— Sparse Rational Approximation —
Pole filtering by regularization & feature selection for complex matrix rational approximation

Pablo Ducru[a]

[a] *Nuclear Science & Engineering department, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA.*

Abstract

Mission:

- Meromorphic operators are the fundamental representation of resonance phenomena in physics, and Runge's theorem establishes rational fractions are universal approximators of meromorphic functions on any given region of the complex plane.
- Thus, myriad operators in physics or electrical engineering are best characterized by a complex rational fraction expansion.
- To this day, most methods of rational fraction approximation require to specify the order of the approximant: i.e. the number of roots and poles.
- When methods do seek to systematically find the number of poles, they presently take one of the following approaches:
  - solving with various number of poles and select best performance,
  - *ad hoc* problem specific poles filtering, such as low-band or high-frequency filters.
  - pole clustering and replacement with the number of clusters,
  - pole trimming, often with a hard-threshold manual way.
- In this research project, we propose and investigate a group LASSO regularization method for residues sparsification as a means to eliminate spurious poles.

What has been achieved:

- Kernel exploration: we searched for a RKHS structure to embed our problem in. This quest was inconclusive. The method is thus parametric.

- The major result of this research project is theorem 2, which generalizes the proximal forward backward splitting PFBS algorithm to the case of complex matrices. We demonstrate that in a given gradient sense (c.f. "cartesian differentiation" introduced in definition 1), the real-case PFBS algorithm can be formally extended to encompass complex matrices.
- The derived algorithm has implemented in the Julia Language and in a toy problem it was possible to eliminate a pole when LS was unable to.

SETBACK:

- Cauchy matrices condition numbers render any practical application out-of-reach.

WAY FORWARD:

- Write-up the Kernel exploration inconclusive results, to explain why this problem has no apparent RKHS structure (the "pole-residue" atoms are difficult to provide with a Hilbert structure).
- Implementation and performance of the complex Frobenius PFBS algorithm 1, to show the convergence, and verify theorem 2, should now take care of the ill-conditioned problem.
- Elastic-Net regularization seems the way to go, though QR factorization may also be of great use.

Keywords: Rational approximation, feature selection, Group LASSO, complex variable optimization.

---

1. Residues sparsification problem

Provided observed complex data points $(z_k)_{k\in[\![1,N_k]\!]} \in \mathbb{C}$ with weights $(\rho_k)_{k\in[\![1,N_k]\!]} \in \mathbb{R}_+$ (for homoscedastic case $\rho_k = \frac{1}{N_k}$) and their corresponding matrices $\boldsymbol{Y_k} \in \mathbb{C}^{q\times s}$, and given N_p fixed poles $(p_n)_{n\in[\![1,N_P]\!]} \in \mathbb{C}$, we search for the residues and affine coefficients matrices $(\boldsymbol{R_n})_{n\in[\![1,N_P]\!]}, \boldsymbol{C}, \boldsymbol{A} \in \mathbb{C}^{q\times s}$ that minimize the rational approximation regularized weighted least square (LS) problem:

$$\min_{\boldsymbol{R_n},\boldsymbol{C},\boldsymbol{A}} \left\{ \sum_{k=1}^{N_k} \rho_k \left\| \sum_{n=1}^{N_p} \frac{\boldsymbol{R_n}}{z_k - p_n} + \boldsymbol{C} + z_k \boldsymbol{A} - \boldsymbol{Y_k} \right\|_F^2 + \lambda \left\| \boldsymbol{R} \right\|_{\mathrm{L}_1 - F} \right\} \tag{1}$$

where $\boldsymbol{R} \coloneqq \left[\boldsymbol{R_1}, \ldots, \boldsymbol{R_n}, \ldots, \boldsymbol{R_{N_p}}\right]$ and $\|\boldsymbol{R}\|_{\mathrm{L}_1-F} \coloneqq \sum_{n=1}^{N_p} \|\boldsymbol{R_n}\|_F$ is a sparsity-based group-LASSO penalization, with $\|\cdot\|_F$ being the Frobenius norm associated to the Hilbert-Schmidt hermitian product. This $\mathrm{L}_1 - F$ regularization will drive low-incidence residue matrices to zero when solving problem (1). This

Email address: p_ducru@mit.edu (Pablo Ducru)

residues sparsification can then be used to discard superfluous poles by identifying the poles associated to zero-matrix residues.

The sparsity-inducing regularization in problem (1) is in essence a feature extraction by 'kitchen-sink approach': we prescribe a number N_p of poles deemed to be higher than the optimal number $N_p^{\text{opt}} \ll N_p$, and filter-out the poles associated to zero residues $\boldsymbol{R_n} = \boldsymbol{0}$, as prescribed in the final algorithm 2.

Defining the coefficients matrix $\boldsymbol{K}$ we seek to find

$$\boldsymbol{K} \coloneqq \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{C} \\ \boldsymbol{A} \end{bmatrix} \tag{2}$$

and the weights matrices $\boldsymbol{W_k}$ composed of the Cauchy matrices at the poles (p_n) and data points (z_k):

$$\boldsymbol{W_k} \coloneqq \left[\frac{\mathbb{I}}{z_k - p_1}, \ldots, \frac{\mathbb{I}}{z_k - p_{N_p}}, \mathbb{I}, z_k \mathbb{I} \right] \tag{3}$$

where $\mathbb{I}$ designates the $\mathbb{C}^{q \times q}$ identity matrix, the residues sparsification problem (1) appears as a special case of the more general problem:

$$\min_{\boldsymbol{K}} \left\{ \sum_{k=1}^{N_k} \rho_k \left\| \boldsymbol{W_k K} - \boldsymbol{Y_k} \right\|_F^2 + \lambda \left\| \boldsymbol{R} \right\|_{\mathrm{L}_1 - F} \right\} \tag{4}$$

where the specific $\boldsymbol{W_k}$ weights matrices are tailored to the simple poles and residues feature space.

Defining the empirical error

$$\widehat{\mathcal{E}}\left(\boldsymbol{R}, \boldsymbol{C}, \boldsymbol{A}\right) \coloneqq \sum_{k=1}^{N_k} \rho_k \left\| \sum_{n=1}^{N_p} \frac{\boldsymbol{R_n}}{z_k - p_n} + \boldsymbol{C} + z_k \boldsymbol{A} - \boldsymbol{Y_k} \right\|_F^2 \tag{5}$$

or equivalently

$$\widehat{\mathcal{E}}\left(\boldsymbol{K}\right) \coloneqq \sum_{k=1}^{N_k} \rho_k \left\| \boldsymbol{W_k K} - \boldsymbol{Y_k} \right\|_F^2 \tag{6}$$

we are thus solving the regularization problem:

$$\min_{\boldsymbol{R}, \boldsymbol{C}, \boldsymbol{A}} \left\{ \widehat{\mathcal{E}}\left(\boldsymbol{R}, \boldsymbol{C}, \boldsymbol{A}\right) + \lambda \left\| \boldsymbol{R} \right\|_{\mathrm{L}_1 - F} \right\} \tag{7}$$

2. In quest of a reproducing kernel Hilbert space

We could however seek to perform this search of optimal complexity in an unsupervised way, by invoking a representer theorem and calling the reproducing kernel on the given training points $(z_k)_{k\in[\![1,N_k]\!]}$.

In this section, we present some research we undertook to exhibit a reproducing kernel Hilbert space structure to this problem. In effect, we failed to unveil a convincing RKHS structure.

The essential reason for this is that the "pole-residue" atoms are difficult to provide with a Hilbert space nature. The use of contour integrals and the residues theorem has enabled us to exhibit some sesquilinear hermitian operator with reproducing property, but the space is not stable through the kernel. We have not yet transcribed here these hand-written results.

3. Proximal Forward Backward Splitting Iteration for Complex Matrices

In this section we establish that the traditional Proximal Forward-Backward Splitting (PFBS) iteration can be generalized to deal with complex matrices in the case of problem (1). Our main result is theorem 2, where we entirely characterize a PFBS iteration for complex matrices, with a specific definition of the gradient — the carthesian differential — introduced in definition 1.

This is not an obvious result at all, because holomorphic differentiation with respect to the complex variable changes the nature and behavior of complex gradients. We however notice and demonstrate that the quadratic norm structure of problem (1) lends itself well to the use of the usually ill-suited definition of complex differential — here named Cartesian differential and specified in definition 1 — which then enables us to seamlessly extend the expression of a PFBS iteration to complex matrices.

Definition 1. Cartesian complex differential.
Let $z \in \mathbb{C}$, $z = x + \mathrm{i}y$, $x, y \in \mathbb{R}$. The holomorphic differential is defined as

$$\partial_z \coloneqq \frac{1}{2}\left(\partial_x - \mathrm{i}\partial_y\right) \tag{8}$$

Note that $\partial_z z = 1$, and $\partial_z \overline{z} = 0$, where $\overline{z} \coloneqq x - iy$ designates the complex conjugate.

We here define the Cartesian differential as:

$$\eth_z \coloneqq \partial_x + \mathrm{i}\partial_y \tag{9}$$

Note that $\eth_z z = 0$, and $\eth_z \overline{z} = 2$.

The Cartesian differential is thus *a priori* a poor choice for complex differentials. We however here demonstrate that it is well suited to our specific quadratic problem (1).

Lemma 1. Cartesian differential of Frobenius norm.
Let $\boldsymbol{A} \in \mathbb{C}^{n\times m}$ be a complex matrix, and $\|\cdot\|_F^2$ be the Frobenuis norm associated with the Hilbert-Schmidt hermitian product $\langle \boldsymbol{A}, \boldsymbol{B}\rangle \coloneqq \mathrm{Tr}\left(\boldsymbol{A}^*\boldsymbol{B}\right)$, with $\boldsymbol{A}^* \coloneqq \overline{\boldsymbol{A}}^\top$, then the following equalities hold:

$$\partial_{\boldsymbol{A}} \|\boldsymbol{A}\|_F^2 = \overline{\boldsymbol{A}} \quad \textit{and} \quad \partial_{\boldsymbol{A}} \|\boldsymbol{A}\|_F = \frac{1}{2}\frac{\overline{\boldsymbol{A}}}{\|\boldsymbol{A}\|_F} \tag{10}$$

while Cartesian differentiation yields:

$$\eth_{\boldsymbol{A}} \|\boldsymbol{A}\|_F^2 = 2\boldsymbol{A} \quad \textit{and} \quad \eth_{\boldsymbol{A}} \|\boldsymbol{A}\|_F = \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F} \tag{11}$$

Proof. Applying the Cartesian differential definition 1 to each element $a_{ij} \in \mathbb{C}$ of $\boldsymbol{A}$ in the expression of the norm $\|\boldsymbol{A}\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2 = \sum_{i=1}^n \sum_{j=1}^m \Re[a_{ij}]^2 + \Im[a_{ij}]^2$ readily yields the results. □

Lemma 1 shows that the Cartesian differential symbolically behaves on the norm of complex matrices like a real differential would behave on the norm of real matrices, all the while it does not behave like the real differential on the vector themselves (recall $\eth_{\boldsymbol{A}}\boldsymbol{A} = \boldsymbol{0}$). We can now use Lemma 1 to demonstrate the first main result: theorem 1.

Theorem 1. Proximal operator of Frobenius penalization for complex matrices.
Let $\boldsymbol{A} \in \mathbb{C}^{n\times m}$ be a complex matrix and $\lambda \in \mathbb{R}_+$, the proximal operator of a Frobenius norm penalization is defined as:

$$\mathrm{prox}_{\lambda\|\cdot\|_F}(\boldsymbol{A}) \coloneqq \underset{\boldsymbol{M}\in\mathbb{C}^{n\times m}}{\arg\min} \left\{ \frac{1}{2}\|\boldsymbol{M}-\boldsymbol{A}\|_F^2 + \lambda\|\boldsymbol{M}\|_F \right\} \tag{12}$$

In the Cartesian differential sense, the proximal operator has the closed explicit form:

$$\mathrm{prox}_{\lambda\|\cdot\|_F}(\boldsymbol{A}) = \left(\boldsymbol{A} - \lambda\frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F}\right)\mathbb{1}_{\left\{\|\boldsymbol{A}\|_F \geq \lambda\right\}} = \boldsymbol{A}\cdot\max\left\{0, 1-\lambda\frac{1}{\|\boldsymbol{A}\|_F}\right\} \tag{13}$$

Which is the formally the same expression as for real matrices.

Proof. The optimality condition on the subgradient, valid for both real and imaginary partial differentials, can thus also be expressed with the Cartesian differential from definition 1, i.e. let $\boldsymbol{M}_{\mathrm{opt}}$ be the minimizer, then $\boldsymbol{0} \in \eth_{\boldsymbol{M}}\left(\frac{1}{2}\|\boldsymbol{M}_{\mathrm{opt}}-\boldsymbol{A}\|_F^2 + \lambda\|\boldsymbol{M}_{\mathrm{opt}}\|_F\right)$. Applying lemma 1 to this Cartesian differential yields:

$\boldsymbol{A} \in \left(\mathbb{I} + \lambda \frac{\mathbb{I}}{\|\cdot\|_F}\right)\boldsymbol{M}_{\text{opt}}$, where $\mathbb{I} \in \mathbb{C}^{n\times m}$ indicates the identity matrix. Formally, we are thus seeking to invert: $\boldsymbol{M}_{\text{opt}} = \left(\mathbb{I} + \lambda \frac{\mathbb{I}}{\|\cdot\|_F}\right)^{-1}\boldsymbol{A}$. Let us now demonstrate that this inverse in given by: $\boldsymbol{M}_{\text{opt}} = \left(\boldsymbol{A} - \lambda \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F}\right)\mathbb{1}_{\left\{\|\boldsymbol{A}\|_F \geq \lambda\right\}}$. First, let us notice that since $\lambda \geq 0$, the real and imaginary parts of each element of $\boldsymbol{A}$ and of $\boldsymbol{M}_{\text{opt}}$ must be of same sign, and taking the norm yields $\|\boldsymbol{A}\|_F \geq \lambda$, unless $\boldsymbol{M}_{\text{opt}} = \boldsymbol{0}$ in which case $\boldsymbol{A} = \boldsymbol{0}$. In the case $\|\boldsymbol{A}\|_F \geq \lambda$ we can show (13) holds by noticing that $\boldsymbol{A} = \left(\mathbb{I} + \lambda \frac{\mathbb{I}}{\|\cdot\|_F}\right)\boldsymbol{M}_{\text{opt}}$ is equivalent to $\boldsymbol{A} = \boldsymbol{A} - \boldsymbol{\lambda}\frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_{\boldsymbol{F}}} + \lambda \frac{\boldsymbol{A} - \lambda \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F}}{\left\|\boldsymbol{A} - \lambda \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F}\right\|_F}$, i.e. $\frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F} = \frac{\boldsymbol{A} - \lambda \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F}}{\left\|\boldsymbol{A} - \lambda \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_F}\right\|_F}$, where this colinearity result is true iff $\|\boldsymbol{A}\|_F \geq \lambda$. This terminates the proof of (13). □

Having established the closed-form of the proximal operator in the Cartesian differential sense in theorem 1, we can now define a Cartesian differential proximal forward backward splitting iteration scheme:

Definition 2. Cartesian differential proximal forward backward splitting iteration.
For complex matrices, we define the Cartesian differential proximal forward backward splitting iteration for problem (7) with learning rate $\gamma \in \mathbb{R}_+$ as:

$$\begin{bmatrix} \boldsymbol{R_n}^{(t+1)} \\ \boldsymbol{C}^{(t+1)} \\ \boldsymbol{A}^{(t+1)} \end{bmatrix} \coloneqq \begin{bmatrix} \text{prox}_{\lambda\|\cdot\|_F}\left(\boldsymbol{R_n}^{(t)} - \gamma \eth_{\boldsymbol{R_n}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right)\right) \\ \boldsymbol{C}^{(t)} - \gamma \eth_{\boldsymbol{C}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right) \\ \boldsymbol{A}^{(t)} - \gamma \eth_{\boldsymbol{A}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right) \end{bmatrix} \tag{14}$$

Where the block expression stems from the block-property of the proximal operator with respect to the group penalization: $\|\boldsymbol{R}\|_{\text{L}_1-F} \coloneqq \sum_{n=1}^{N_p} \|\boldsymbol{R_n}\|_F$.

This definition is a direct translation to complex matrices of the real PFBS algorithm, where the real gradient $\partial(\cdot)$ has been replaced with the Cartesian derivative $\eth(\cdot)$.

We are now equiped to establish our main result, theorem 2:

Theorem 2. Proximal forward backward splitting algorihtm for Frobenius penalization of complex matrices.
The Cartesian differential proximal forward backward splitting iteration for problem (1) takes the form:

$$\begin{bmatrix} \boldsymbol{R_n}^{(t+1)} \\ \boldsymbol{C}^{(t+1)} \\ \boldsymbol{A}^{(t+1)} \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{R_n}^{(t)} - \gamma \eth_{\boldsymbol{R_n}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right)\right) \cdot \max\left\{0, 1 - \frac{\lambda}{\left\|\boldsymbol{R_n}^{(t)} - \gamma \eth_{\boldsymbol{R_n}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right)\right\|_F}\right\} \\ \boldsymbol{C}^{(t)} - \gamma \eth_{\boldsymbol{C}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right) \\ \boldsymbol{A}^{(t)} - \gamma \eth_{\boldsymbol{A}} \widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right) \end{bmatrix} \tag{15}$$

where:

$$\eth_{\boldsymbol{R_n}}\widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)},\boldsymbol{C}^{(t)},\boldsymbol{A}^{(t)}\right)=2\sum_{k=1}^{N_k}\rho_k\frac{1}{(z_k-p_n)^*}\left[\sum_{n=1}^{N_p}\frac{\boldsymbol{R_n}^{(t)}}{z_k-p_n}+\boldsymbol{C}^{(t)}+z_k\boldsymbol{A}^{(t)}-\boldsymbol{Y_k}\right] \tag{16}$$

$$\eth_{\boldsymbol{C}}\widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)},\boldsymbol{C}^{(t)},\boldsymbol{A}^{(t)}\right)=2\sum_{k=1}^{N_k}\rho_k\left[\sum_{n=1}^{N_p}\frac{\boldsymbol{R_n}^{(t)}}{z_k-p_n}+\boldsymbol{C}^{(t)}+z_k\boldsymbol{A}^{(t)}-\boldsymbol{Y_k}^{(t)}\right] \tag{17}$$

$$\eth_{\boldsymbol{A}}\widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)},\boldsymbol{C}^{(t)},\boldsymbol{A}^{(t)}\right)=2\sum_{k=1}^{N_k}\rho_k z_k^*\left[\sum_{n=1}^{N_p}\frac{\boldsymbol{R_n}^{(t)}}{z_k-p_n}+\boldsymbol{C}^{(t)}+z_k\boldsymbol{A}^{(t)}-\boldsymbol{Y_k}^{(t)}\right] \tag{18}$$

Moreover, the Cartesian differential PFBS algorithm, which repeats iteration (15), is guaranteed to converge if the learning rate $\gamma\in\mathbb{R}_+$ respects:

$$\gamma\leq\frac{1}{2\max\left\{\mathcal{S}_p\left(\sum_{k=1}^{N_k}\rho_k{\boldsymbol{W_k}}^*\boldsymbol{W_k}\right)\right\}} \tag{19}$$

Proof. Expression (15) is a direct application of theorem 1 to express the proximal operator for complex matrices in the Cartesian differential sense.

To establish expressions (16), (17), and (18),we are going to show that the Cartesian differential satisfies (20), which is equivalent to expressions (16), (17), and (18) combined.

$$\eth_{\boldsymbol{K}}\widehat{\mathcal{E}}\left(\boldsymbol{K}^{(t)}\right)=2\sum_{k=1}^{N_k}\rho_k{\boldsymbol{W_k}}^*\left[\boldsymbol{W_k}\boldsymbol{K}^{(t)}-\boldsymbol{Y_k}\right] \tag{20}$$

We drop the $^{(t)}$ iteration superscript for clarity, and decompose into the real and imaginary components: $\widehat{\mathcal{E}}(\boldsymbol{K})=\sum_{k=1}^{N_k}\rho_k\Big(\Re[\boldsymbol{W_k}\boldsymbol{K}-\boldsymbol{Y_k}]^2+\Im[\boldsymbol{W_k}\boldsymbol{K}-\boldsymbol{Y_k}]^2\Big)$. Expressing the real and imaginary part $\boldsymbol{W_k}=\Re[\boldsymbol{W_k}]+\mathrm{i}\Im[\boldsymbol{W_k}]$ and $\boldsymbol{K}=\Re[\boldsymbol{K}]+\mathrm{i}\Im[\boldsymbol{K}]$ yields $\widehat{\mathcal{E}}(\boldsymbol{K})=\sum_{k=1}^{N_k}\rho_k\Bigg\{\Big(\Re[\boldsymbol{W_k}]\Re[\boldsymbol{K}]-\Im[\boldsymbol{W_k}]\Im[\boldsymbol{K}]-\Re[\boldsymbol{Y_k}]\Big)^2+\Big(\Im[\boldsymbol{W_k}]\Re[\boldsymbol{K}]+\Re[\boldsymbol{W_k}]\Im[\boldsymbol{K}]-\Re[\boldsymbol{Y_k}]\Big)^2\Bigg\}$. The Cartesian differential $\eth_{\boldsymbol{K}}\coloneqq\partial_{\Re[\boldsymbol{K}]}+\mathrm{i}\partial_{\Im[\boldsymbol{K}]}$ then yields $\eth_{\boldsymbol{K}}\widehat{\mathcal{E}}(\boldsymbol{K})=\sum_{k=1}^{N_k}\rho_k\Bigg\{2\Big(\Re[\boldsymbol{W_k}]\Re[\boldsymbol{K}]-\Im[\boldsymbol{W_k}]\Im[\boldsymbol{K}]-\Re[\boldsymbol{Y_k}]\Big)\cdot\Big(\Re[\boldsymbol{W_k}]-\mathrm{i}\Im[\boldsymbol{W_k}]\Big)+2\Big(\Im[\boldsymbol{W_k}]\Re[\boldsymbol{K}]+\Re[\boldsymbol{W_k}]\Im[\boldsymbol{K}]-\Re[\boldsymbol{Y_k}]\Big)\cdot\Big(\Im[\boldsymbol{W_k}]+\mathrm{i}\Re[\boldsymbol{W_k}]\Big)\Bigg\}$ i.e. $\eth_{\boldsymbol{K}}\widehat{\mathcal{E}}(\boldsymbol{K})=\sum_{k=1}^{N_k}\rho_k\Bigg\{2\Re[\boldsymbol{W_k}\boldsymbol{K}-\boldsymbol{Y_k}]\cdot\Big(\Re[\boldsymbol{W_k}]-\mathrm{i}\Im[\boldsymbol{W_k}]\Big)+2\mathrm{i}\Im[\boldsymbol{W_k}\boldsymbol{K}-\boldsymbol{Y_k}]\cdot\Big(\Re[\boldsymbol{W_k}]-\mathrm{i}\Im[\boldsymbol{W_k}]\Big)\Bigg\}$ that is, noticing ${\boldsymbol{W_k}}^*=\Big(\Re[\boldsymbol{W_k}]-\mathrm{i}\Im[\boldsymbol{W_k}]\Big)$, the Cartesian differential takes the form of equation (20), which establishes expressions (16), (17), and (18).

We now seek to establish the learning rate inequality (19), according to the well known property that maximum decrease is achieved with the learning rate $\gamma=\frac{1}{L}$. To do this, one must express the Hessian of

the loss function, and find its maximum eigenvalue, so as to obtain the corresponding Lipschitz constant L, according to $L = \max\left\{\mathcal{S}p\left(\text{Hess}\left(\widehat{\mathcal{E}}\right)\right)\right\}$. We remind that, since $\eth_z z = 0$, the Cartesian differential of (20) would be zero: $\eth^2_{\boldsymbol{K}}\widehat{\mathcal{E}}\left(\boldsymbol{K}^{(t)}\right) = 2\sum_{k=1}^{N_k}\rho_k \boldsymbol{W_k}^*\boldsymbol{W_k}\left(\eth_{\boldsymbol{K}}\boldsymbol{K}^{(t)}\right) = 0$. However, each derivative along the real or imaginary component yields: $\partial_{\Re[\boldsymbol{K}]}\eth_{\boldsymbol{K}}\widehat{\mathcal{E}}\left(\boldsymbol{K}^{(t)}\right) = 2\sum_{k=1}^{N_k}\rho_k \boldsymbol{W_k}^*\boldsymbol{W_k} = -\mathrm{i}\partial_{\Im[\boldsymbol{K}]}\eth_{\boldsymbol{K}}\widehat{\mathcal{E}}\left(\boldsymbol{K}^{(t)}\right)$. This means that the Hessian can be interpreted as $2\sum_{k=1}^{N_k}\rho_k \boldsymbol{W_k}^*\boldsymbol{W_k}$, which will yield real positive eigenvalues according to the spectral theorem. maximal eigenvalue of the Hessian yields the Lipschitz constant and thus learning rate γ according to (19), Q.E.D. □

4. Computational PFBS algorithm

Having established an analytical expression of the PFBS iteration in theorem 2, we here transcribe it into the computational Frobenius penalization complex matrix PFBS algorithm 1, to solve problem (1), for a given λ, provided a tolerance ϵ on the empirical error defined on the training set as in (5) or (6):

Algorithm 1 Complex Frobenius PBFS algorithm

Input: $\boldsymbol{R}^{(0)}$, $\boldsymbol{C}^{(0)}$, and $\boldsymbol{A}^{(0)}$ initial guesses ; λ meta-parameter ; ϵ accuracy ; γ learning rate from (19)
while $\widehat{\mathcal{E}}\left(\boldsymbol{R}^{(t)}, \boldsymbol{C}^{(t)}, \boldsymbol{A}^{(t)}\right) \leq \epsilon$ **do**
  Update $\boldsymbol{R_n}^{(t+1)}, \boldsymbol{C}^{(t+1)}, \boldsymbol{A}^{(t+1)}$ using equation (15)
  $t \to t+1$
end while
Output: $\boldsymbol{R}^{(\lambda)} = \left[\boldsymbol{R_n}^{(t_{\text{end}})}\right]_{n\in[\![1,N_p]\!]}$, $\boldsymbol{C}^{(\lambda)} = \boldsymbol{C}^{(t_{\text{end}})}$, and $\boldsymbol{A}^{(\lambda)} = \boldsymbol{A}^{(t_{\text{end}})}$.

Defining the cross validation error as the empirical L_2 loss on a cross-validation grid $(z_\ell)_{\ell\in[\![1,N_{\text{CV}}]\!]} \in \mathbb{C}$:

$$\widehat{\mathcal{E}}_{\text{CV}}\left(\boldsymbol{R}^{(\lambda)}, \boldsymbol{C}^{(\lambda)}, \boldsymbol{A}^{(\lambda)}\right) \coloneqq \sum_{\ell=1}^{N_{\text{CV}}}\rho_\ell\left\|\sum_{n=1}^{N_p}\frac{\boldsymbol{R_n}^{(\lambda)}}{z_\ell - p_n} + \boldsymbol{C}^{(\lambda)} + z_\ell\boldsymbol{A}^{(\lambda)} - \boldsymbol{Y_\ell}\right\|_F^2 \tag{21}$$

where $\boldsymbol{R}^{(\lambda)}$, $\boldsymbol{C}^{(\lambda)}$, and $\boldsymbol{A}^{(\lambda)}$ are the solutions of problem (1) for a given λ, the meta-optimization process of λ selection can then be performed by selecting the λ as that minimizes the empirical L_2 norm on a cross-validation grid $(z_\ell)_{\ell\in[\![1,N_{\text{CV}}]\!]} \in \mathbb{C}$: i.e.

$$\lambda_{\text{CV}} \coloneqq \underset{\lambda\in\mathbb{R}_+}{\arg\min}\left\{\widehat{\mathcal{E}}_{\text{CV}}\left(\boldsymbol{R}^{(\lambda)}, \boldsymbol{C}^{(\lambda)}, \boldsymbol{A}^{(\lambda)}\right)\right\} \tag{22}$$

In practice, we can solve the meta-optimization problem (22) on a logarithmic grid of $\lambda_m \in [\lambda_{\min}, \lambda_{\max}]$

values comprised in the interval

$$0 \leq \lambda_m \leq \frac{1}{2\max\left\{\mathcal{S}_p\left(\sum_{k=1}^{N_k} \rho_k \boldsymbol{W_k}^* \boldsymbol{W_k}\right)\right\}} \tag{23}$$

The order in which we span this grid is important: we start with $\lambda_{\max}$ and then solve problem (1) by running algorithm 1, yielding $\left(\boldsymbol{R}^{(\lambda_{\max})}, \boldsymbol{C}^{(\lambda_{\max})}, \boldsymbol{A}^{(\lambda_{\max})}\right)$, on which we can calculate $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_{\max})}, \boldsymbol{C}^{(\lambda_{\max})}, \boldsymbol{A}^{(\lambda_{\max})}\right)$. We then solve problem (1) with the next lower $\lambda_{\max-1}$ value, by running algorithm 1 with initial guesses $\left(\boldsymbol{R}^{(\lambda_{\max})}, \boldsymbol{C}^{(\lambda_{\max})}, \boldsymbol{A}^{(\lambda_{\max})}\right)$. This will yield $\left(\boldsymbol{R_n}^{(\lambda_{\max-1})}, \boldsymbol{C}^{(\lambda_{\max-1})}, \boldsymbol{A}^{(\lambda_{\max-1})}\right)$, from which we can compute $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_{\max-1})}, \boldsymbol{C}^{(\lambda_{\max-1})}, \boldsymbol{A}^{(\lambda_{\max-1})}\right)$. We then repeat the process going down the λ_m grid as long as $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_{m-1})}, \boldsymbol{C}^{(\lambda_{m-1})}, \boldsymbol{A}^{(\lambda_{m-1})}\right) \leq \widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_m)}, \boldsymbol{C}^{(\lambda_m)}, \boldsymbol{A}^{(\lambda_m)}\right)$, until we find a minimum when $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_{m-1})}, \boldsymbol{C}^{(\lambda_{m-1})}, \boldsymbol{A}^{(\lambda_{m-1})}\right) > \widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_m)}, \boldsymbol{C}^{(\lambda_m)}, \boldsymbol{A}^{(\lambda_m)}\right)$.

At each new λ_m iteration, some $\boldsymbol{R_n}$ residues will have been drawn to zero, and we can then discard the corresponding poles p_n. We can then re-calculate the learning rate

$$\gamma_m = \frac{1}{2\max\left\{\mathcal{S}_p\left(\sum_{k=1}^{N_k} \rho_k \boldsymbol{W_k}^{(m)^*} \boldsymbol{W_k}^{(m)}\right)\right\}} \tag{24}$$

where $\boldsymbol{W_k}^{(m)} \coloneqq \left[\frac{\mathbb{I}}{z_k - p_1}, \ldots, \frac{\mathbb{I}}{z_k - p_{N_p^{(m)}}}, \mathbb{I}, z_k\mathbb{I}\right]$ with $N_p^{(m)} \leq N_p^{(m-1)}$, where $(p_j)_{j \in [\![1, N_p^{(m-1)}]\!]}$ is the set of remaining poles, from which the spurious poles have been discarded.

With this process, the meta-optimization is made faster, by pre-converging the initial guesses and sparsity exhibited by sub-optimal λ_m values. This yields the complex Frobenius PFBS meta-algorithm 2

Algorithm 2 Complex Frobenius PBFS meta-algorithm

Input: $\boldsymbol{R}^{(0)}$, $\boldsymbol{C}^{(0)}$, and $\boldsymbol{A}^{(0)}$ initial guesses ; $[\lambda_{\max}, \lambda_{\max}]$ meta-parameter grid ; ϵ accuracy
while $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_{m-1})}, \boldsymbol{C}^{(\lambda_{m-1})}, \boldsymbol{A}^{(\lambda_{m-1})}\right) \leq \widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_m)}, \boldsymbol{C}^{(\lambda_m)}, \boldsymbol{A}^{(\lambda_m)}\right)$ **do**
  run algorithm 1 with initial guesses $\left(\boldsymbol{R}^{(\lambda_{m-1})}, \boldsymbol{C}^{(\lambda_{m-1})}, \boldsymbol{A}^{(\lambda_{m-1})}\right)$, learning rate γ_m from (24)
  eliminate spurious poles (p_n) corresponding to zero residues $\boldsymbol{R_n} = \boldsymbol{0}$
  compute cross validation error $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_m)}, \boldsymbol{C}^{(\lambda_m)}, \boldsymbol{A}^{(\lambda_m)}\right)$
  $m \to m - 1$
end while
Output: $\left(\boldsymbol{R}^{(\lambda_{\mathrm{opt}})}, \boldsymbol{C}^{(\lambda_{\mathrm{opt}})}, \boldsymbol{A}^{(\lambda_{\mathrm{opt}})}\right)$, optimal meta-parameter λ_{opt}, lowest cross validation error $\widehat{\mathcal{E}}_{\mathrm{CV}}\left(\boldsymbol{R}^{(\lambda_{\mathrm{opt}})}, \boldsymbol{C}^{(\lambda_{\mathrm{opt}})}, \boldsymbol{A}^{(\lambda_{\mathrm{opt}})}\right)$.

5. PFBS algorithm : implementation & computational performance

To test the validity of our theorem (2), and the subsequent computational complex Frobenius PFBS algorithm 1 and meta-algorithm 2, we devise a test set of points $(z_k)_{k\in[\![1,N_k]\!]} \in \mathbb{C}$ and matrices $\boldsymbol{Y_k} \in \mathbb{C}_{q\times s}$ with weights $(\rho_k)_{k\in[\![1,N_k]\!]} \in \mathbb{R}_+$, which we will have generated from a known reference complex matrix operator $\boldsymbol{Y}(z) = \frac{\boldsymbol{R_n}^{\text{ref}}}{z_k - p_n} + \boldsymbol{C}^{\text{ref}} + z_k \boldsymbol{A}^{\text{ref}}$, so that $\boldsymbol{Y_k} \coloneqq \boldsymbol{Y}(z_k)$.

We then searched for poles with and without noise: i.e. $p_n \coloneqq p_n + \epsilon$. When no noise was introduced, the LS method is clearly superior, and manages to find all the residues (including the zero ones). However, when the poles are perturbed by less than 10%, the group-LASSO was able to eliminate the corresponding residue. This shows that our method is theoretically valid, albeit more work has to be done to palliate the issue of extreme condition numbers of the Cauchy matrices. Elastic net or QR seems the way to go.

Note that we could not have introduced the noise in the residues, i.e. $\boldsymbol{Y_k} \coloneqq \boldsymbol{Y}(z_k) + \epsilon_{\mathcal{N}}$, with Gaussian noise $\epsilon_{\mathcal{N}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$.

6. Conclusions

- We theoretically generalized the PFBS algorithm with L_1-Frobenius group LASSO penalization to the case of complex matrices.
- We implemented it in Julia and proved it write on the polynomial and residues case.
- We stumbled upon very significant ill-conditioning problems that made it very hard to continue
- Elastic Net regularization to address this issue is our next step.
- De-bug for all cases.
- Make sure the regularization does trim the right poles !
- study the relation between number of points and conditioning.
- Code Elastic net for conditioning