Computing the nearest stable matrix via optimization on matrix manifolds

Federico Poloni (University of Pisa)
Joint work with V. Noferini (Aalto university)

Oselot online seminars, October 2020
Two problems: “the ∃ problem”

Given Hurwitz stable $A \in \mathbb{C}^{n \times n}$, find nearest non-stable $B$.
More generally: given $A$ and closed region $\Omega \subseteq \mathbb{C}$, find

$$\min_{B \in \mathbb{C}^{n \times n}} \| A - B \|_F \quad \exists \lambda \in \Lambda(B) \cap \Omega$$

Application: how much noise can we add so that $\dot{x} = Ax$ stays stable?
Two problems: “the $\forall$ problem”

Given non-stable $A \in \mathbb{C}^{n \times n}$, find nearest stable $B$.

More generally: given $A$ and closed region $\Omega \subseteq \mathbb{C}$, find

$$\min_{B \in \mathbb{C}^{n \times n}, \Lambda(B) \subseteq \Omega} \| A - B \|_F$$

Application: noise made $\dot{x} = Ax$ unstable; how to ‘fix’ $A$?
Comparing the two problems

When $A$ is non-normal, there is no simple solution.

Previous work on these problems or variants: Benner, Burke, Byers, Gillis, Guglielmi, He, Hinrichsen, Karow, Kostić, Lewis, Meerbergen, Mehl, Mehrmann, Mengi, Michiels, Międlar, Mitchell, Nesterov, Overton, Pritchard, Protasov, Sharma, Stolwijk, Van Dooren, Watson, . . . (and surely I have missed many).

Keywords: nearest $\Omega$-stable matrix, pseudospectral abscissa, robust stability, distance to (in)stability.

Most focus on the Frobenius norm $\|M\|_F = \left(\sum_{i,j=1}^{n}|M_{ij}|^2\right)^{1/2}$.

The $\forall$ problem is considered more difficult; we need to juggle multiple eigenvalues at the same time.

In this talk: the $\forall$ problem, but the technique can be extended to the $\exists_k$ problem.
An “MO-hard” special case

Nearest matrix with all real eigenvalues: $\Omega = \mathbb{R}$.

Attempts to find a closed-form solution (without luck) on Matlab Central and Mathoverflow, dating back to 2010.
The background

Distance from a closed set is a classical topic in mathematical analysis.

Given $\Omega \subseteq \mathbb{R}^N$ (or also $\mathbb{C} \simeq \mathbb{R}^2$), study functions

$$d^2_\Omega(x) = \min_{y \in \Omega} \|x - y\|^2,$$
$$p_\Omega(x) = \arg \min_{y \in \Omega} \|x - y\|^2.$$

Known results: $d^2_\Omega$ is continuous, semiconcave, and differentiable everywhere apart from a (measure-zero) set where $p_\Omega(x)$ is not unique.
The background

Distance from a closed set is a classical topic in mathematical analysis.

Given $\Omega \subseteq \mathbb{R}^N$ (or also $\mathbb{C} \simeq \mathbb{R}^2$), study functions

$$d^2_\Omega(x) = \min_{y \in \Omega} \|x - y\|^2, \quad p_\Omega(x) = \arg \min_{y \in \Omega} \|x - y\|^2.$$  

**Known results:** $d^2_\Omega$ is continuous, semiconcave, and differentiable everywhere apart from a (measure-zero) set where $p_\Omega(x)$ is not unique.

![Graph showing the squared distance from $\Omega = \{0\} \cup [2, \infty)$ with two smooth segments and two equidistant, nonsmooth segments.](image)
The set of Hurwitz stable matrices

The set \( \{ X \in \mathbb{C}^{n \times n} : \Lambda(X) \subseteq \Omega \} \) is closed, so the same results hold for nearest \( \Omega \)-stable matrix problems.

**Challenge 1**: the set of Hurwitz stable matrices is non-smooth and non-convex, already for \( n = 2 \). Many local minima.

**Challenge 2**: minimizers often have multiple eigenvalues \( \Rightarrow \) non-differentiability.
Yet another approach

Our approach: reformulation as optimization on matrix manifolds.

Basic idea simple enough that we can explain it in a few slides.

The problem

\[ B = \arg \min_{\Lambda(X) \subseteq \Omega} \| A - X \|_F. \]

Real and a complex version:

1. Nearest \( X \in \mathbb{C}^{n \times n} \) to a given \( A \in \mathbb{C}^{n \times n} \);
2. Nearest \( X \in \mathbb{R}^{n \times n} \) to a given \( A \in \mathbb{R}^{n \times n} \).

We start from the complex case.
On triangular matrices

Let us first solve a simpler problem: $X$ upper triangular.

$$
T(A) = \arg \min_{\lambda(T) \subseteq \Omega} \|A - T\|_F
$$

$T$ upper triangular

$$
= \arg \min_{T_{ii} \in \Omega} \left\| \begin{bmatrix}
    A_{11} - T_{11} & A_{12} - T_{12} & A_{13} - T_{13} & \ldots \\
    A_{21} & A_{22} - T_{22} & A_{23} - T_{23} & \ldots \\
    A_{31} & A_{32} & A_{33} - T_{33} & \ldots \\
    \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \right\|_F
$$

Clearly, the best choice is

- $T_{ij} = A_{ij}$ above the diagonal,
- $T_{ii} = \rho_{\Omega}(A_{ii})$ on the diagonal.
On triangular matrices

Let us first solve a simpler problem: $X$ upper triangular.

\[ T(A) = \arg \min_{\Lambda(T) \subseteq \Omega} \| A - T \|_F \]

$sT$ upper triangular

\[ = \arg \min_{T_{ii} \in \Omega} \left\| \begin{bmatrix} A_{11} - T_{11} & A_{12} - T_{12} & A_{13} - T_{13} & \cdots \\ A_{21} & A_{22} - T_{22} & A_{23} - T_{23} & \cdots \\ A_{31} & A_{32} & A_{33} - T_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \right\|_F \]

Clearly, the best choice is

- $T_{ij} = A_{ij}$ above the diagonal,
- $T_{ii} = p_{\Omega}(A_{ii})$ on the diagonal.
Triangular case

**Lemma**

The solution to

\[ \mathcal{T}(A) = \arg \min_{\Lambda(T) \subseteq \Omega} \|A - T\|_F \]

where \( T \) is upper triangular is

\[ \mathcal{T}(A)_{ij} = \begin{cases} A_{ij} & i < j, \\ p_\Omega(A_{ij}) & i = j, \\ 0 & i > j. \end{cases} \]

The optimum is \( \|\mathcal{L}(A)\|_F \), where \( \mathcal{L}(A) = A - \mathcal{T}(A) \) has entries

\[ \mathcal{L}(A)_{ij} = \begin{cases} 0 & i < j, \\ A_{ij} - p_\Omega(A_{ij}) & i = j, \\ A_{ij} & i > j. \end{cases} \]
Example

With \( \Omega = \{ \lambda : \text{Re} \lambda \leq 0 \} \) (nearest Hurwitz stable):

\[
\begin{bmatrix}
3 & -1 & 1 & 2 \\
-1 & -2 & 4 & 0 \\
2 & 1 & -1 & 1 \\
1 & 2 & 2 & 1
\end{bmatrix}
\begin{bmatrix}
0 & -1 & 1 & 2 \\
0 & -2 & 4 & 0 \\
0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
+ \begin{bmatrix}
3 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
1 & 2 & 2 & 1
\end{bmatrix}
\]

\( \mathcal{T}(A) \) is the upper triangular Hurwitz stable matrix nearest to \( A \), with distance \( \| \mathcal{L}(A) \|_F \).
Schur trick

In an unknown basis, the solution $X$ is upper triangular!

**Schur form** $X = UTU^*$: $T$ upper triangular, $U \in \mathcal{U}_n$ (unitary matrices).

\[
\min_{\Lambda(X) \subseteq \Omega} \|A - X\|_F = \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega \atop T \text{ triangular}} \|A - UTU^*\|_F
\]

\[
= \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega \atop T \text{ triangular}} \|U^*AU - T\|_F
\]

\[
= \min_{U \in \mathcal{U}_n} \|\mathcal{L}(U^*AU)\|_F.
\]

We transformed a problem on \{\(\Lambda(X) \subseteq \Omega\)\} into one on \(\mathcal{U}_n\): simpler structure, half as many degrees of freedom.
Schur trick

In an unknown basis, the solution $X$ is upper triangular!

**Schur form** $X = UTU^*$: $T$ upper triangular, $U \in \mathcal{U}_n$ (unitary matrices).

\[
\min_{\Lambda(X) \subseteq \Omega} \|A - X\|_F = \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega} \|A - UTU^*\|_F
\]

\[
= \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega} \|U^*AU - T\|_F
\]

\[
= \min_{U \in \mathcal{U}_n} \|L(U^*AU)\|_F.
\]

We transformed a problem on $\{\Lambda(X) \subseteq \Omega\}$ into one on $\mathcal{U}_n$: simpler structure, half as many degrees of freedom.
Schur trick

In an unknown basis, the solution $X$ is upper triangular!

**Schur form** $X = UTU^*$: $T$ upper triangular, $U \in \mathcal{U}_n$ (unitary matrices).

$$
\min_{\Lambda(X) \subseteq \Omega} \|A - X\|_F = \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega \atop T \, \text{triangular}} \|A - UTU^*\|_F
$$

$$
= \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega \atop T \, \text{triangular}} \|U^*AU - T\|_F
$$

$$
= \min_{U \in \mathcal{U}_n} \|\mathcal{L}(U^*AU)\|_F.
$$

We transformed a problem on $\{\Lambda(X) \subseteq \Omega\}$ into one on $\mathcal{U}_n$: simpler structure, half as many degrees of freedom.
Schur trick

In an unknown basis, the solution $X$ is upper triangular!

Schur form $X = UTU^*$: $T$ upper triangular, $U \in \mathcal{U}_n$ (unitary matrices).

$$\min_{\Lambda(X) \subseteq \Omega} \|A - X\|_F = \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega} \|A - UTU^*\|_F$$

$$= \min_{U \in \mathcal{U}_n} \min_{\Lambda(T) \subseteq \Omega} \|U^*AU - T\|_F$$

$$= \min_{U \in \mathcal{U}_n} \|\mathcal{L}(U^*AU)\|_F.$$ 

We transformed a problem on $\{\Lambda(X) \subseteq \Omega\}$ into one on $\mathcal{U}_n$: simpler structure, half as many degrees of freedom.
Optimization on (matrix) manifolds

Optimization on matrix manifolds has been studied widely recently: see e.g. [Absil, Mahony, Sepulchre book].

Many first- and second-order methods available.

Key ideas:

- switch to Riemannian gradient and Hessian;
- the Riemannian gradient lives in the tangent space; we need a way to “retract” $x_k + g_k$ onto the manifold.
Optimization on manifolds: the set-up

We just use these algorithms as black box (for now).

- **Manifold**: $\mathcal{U}_n$ (unitary matrices).
- **Function**: $f(U) = \|\mathcal{L}(U^*AU)\|_F^2$, with
  \[
  \mathcal{L}(A)_{ij} = \begin{cases} 
  0 & i < j, \\
  A_{ij} - p_\Omega(A_{ij}) & i = j, \\
  A_{ij} & i > j.
  \end{cases}
  \]
- **Gradient**: $\nabla_U f = 2U \text{skew}(TL^* - L^*T)$, where $L = \mathcal{L}(U^*AU)$, $T = T(U^*AU)$, $\text{skew}(M) = \frac{1}{2}(M - M^*)$.
- **Algorithm**: quasi-Newton (trust-region).

Remark There is nothing that computes eigenvalues here. (!!!) The optimization procedure “does that” for us, and returns $X$ in Schur form.

Differentiable formulation: both $f$ and the constraint $U^*U = I$ are $C^1$ (outside of the medial axis).
Optimization on manifolds: the set-up

We just use these algorithms as black box (for now).

- **Manifold:** $U_n$ (unitary matrices).
- **Function:** $f(U) = \|L(U^*AU)\|_F^2$, with
  \[
  L(A)_{ij} = \begin{cases} 
  0 & i < j, \\
  A_{ij} - p_\Omega(A_{ij}) & i = j, \\
  A_{ij} & i > j.
  \end{cases}
  \]
- **Gradient:** $\nabla_U f = 2U \text{skew}(TL^* - L^*T)$, where
  \[
  L = L(U^*AU), \quad T = T(U^*AU), \quad \text{skew}(M) = \frac{1}{2}(M - M^*).
  \]
- **Algorithm:** quasi-Newton (trust-region).

**Remark** There is nothing that computes eigenvalues here. (!!) The optimization procedure “does that” for us, and returns $X$ in Schur form.

Differentiable formulation: both $f$ and the constraint $U^*U = I$ are $C^1$ (outside of the medial axis).
An aside: relation to Jacobi eigensolver

If we run the algorithm with $\Omega = \mathbb{C}$, the solution is $A = B = UTU^*$, i.e., the optimization algorithm just computes the Schur form of $A$.

This reminds of the Jacobi eigenvalue algorithm: apply a series of Givens rotations trying to zero out $\text{tril}(A) \iff$ coordinate descent on $U_n$.

\[
A \mapsto Q_{k\ell}^T A Q_{k\ell} = A'.
\]

In practice, coordinate descent did not perform well on this problem. However, many advanced computational tricks exist for eigensolvers; maybe we can borrow some.
The real case

The real case is more involved, because the real Schur form is more involved.

Easy case: $\mathcal{O} \subseteq \mathbb{R}$.

In this case, each admissible $X$ can be written as $X = QTQ^\top$, where $Q \in \mathcal{O}_n$ (orthogonal matrices) and $T$ is (truly) triangular. Everything works like in the complex case.

This works for the ‘nearest matrix with real eigenvalues’ problem, for instance.

Hard case: general $\mathcal{O}$.

We need to handle $2 \times 2$ blocks in the correct way.
General real case

Each real matrix is similar to

$$
\begin{bmatrix}
T_{11} & T_{12} & T_{13} & \ldots \\
0 & T_{22} & T_{23} & \ldots \\
0 & 0 & T_{33} & \ldots \\
\vdots & \vdots & \vdots & \ddots
\end{bmatrix}
$$

where all $T_{ii}$ are $2 \times 2$, except for a lone final entry if $n$ odd. (The $T_{ii}$ may have real eigenvalues.)

We define $\mathcal{T}(A)$, $\mathcal{L}(A)$ blockwise:

$$
\mathcal{T}(A)_{ij} = \begin{cases} 
A_{ij} & i < j, \\
p_\Omega(A_{ij}) & i = j, \\
0 & i > j,
\end{cases}
\quad
\mathcal{L}(A)_{ij} = \begin{cases} 
0 & i < j, \\
A_{ij} - p_\Omega(A_{ij}) & i = j, \\
A_{ij} & i > j.
\end{cases}
$$

($A_{ij}$ are $2 \times 2$ blocks here.)
Real case: $2 \times 2$ projection

We need a way to compute $p_\Omega(A_{ij})$, i.e., the ‘projection’ of $A_{ij} \in \mathbb{R}^{2 \times 2}$ onto $\{\Lambda(X) \subseteq \Omega\}$.

I.e., we need a way to solve the $2 \times 2$ version of our problem.

This is more involved; we provide an implementation for the Hurwitz stable case.
Projection on Hurwitz stable $2 \times 2$ matrices

Let $A \in \mathbb{R}^{2 \times 2}$, and $B = p_\Omega(A)$ the nearest Hurwitz stable matrix to $A$.

**First result:** we can reduce to matrices with equal diagonal entries.

**Lemma**

Each $A$ is similar to an $\hat{A} = Q^\top AQ$ with $\hat{A}_{11} = \hat{A}_{22}$.

**Lemma**

If $A_{11} = A_{22}$, then $B_{11} = B_{22}$. 
# Projection on Hurwitz stable 2 × 2 matrices

**Second result:** casework based on trace and determinant.

**Lemma (Hurwitz)**

$X \in \mathbb{R}^{2 \times 2}$ Hurwitz stable iff $\text{Tr}(X) \leq 0$, $\text{det}(X) \geq 0$.

**Lemma**

When $A$ is not Hurwitz stable, $B$ is either:

1. a (local) minimizer on $\{\text{Tr}(X) = 0\}$,
2. a (local) minimizer on $\{\text{det}(X) = 0\}$,
3. a (local) minimizer on $\{\text{Tr}(X) = \text{det}(X) = 0\}$.

Minimizers in all three cases can be computed explicitly with a little work (for instance, truncated SVD solves case 2).
The set of $2 \times 2$ Hurwitz stable matrices

We can now make more sense of this picture.
Optimization on manifolds: the set-up

We can formulate a real analogue of the algorithm.

- **Manifold**: $\mathcal{O}_n$ (orthogonal matrices).
- **Function**: $f(Q) = \|L(Q^\top AQ)\|_F^2$, with
  \[
  L(A)_{ij} = \begin{cases} 
  0 & i < j, \\
  A_{ij} - p_{\Omega}(A_{ij}) & i = j, \text{ (the scalar version, if } \Omega \subseteq \mathbb{R}, \text{ or the} \\
  A_{ij} & i > j.
  \end{cases}
  \]
- **Gradient**: $\nabla_Q f = 2Q \text{skew}(TL^\top - L^\top T)$, where
  
  \[L = L(Q^\top AQ), T = T(Q^\top AQ), \text{ skew}(M) = \frac{1}{2}(M - M^\top).\]
- **Algorithm**: quasi-Newton (trust-region).
A conjecture

Let us consider the complex version of the problem

\[ B = \arg \min_{\Lambda(X) \subseteq \Omega} \| A - X \|_F. \]

Open problem

When \( A \) is a real matrix, is \( B \) also always a real matrix?

Experiments suggest so, at least for \( \Omega = \) Hurwitz stable.

If the answer is yes, then one can also use the complex version of the algorithm for the real case.

Pros : simpler to write; no need to solve the 2 × 2 case by hand.
Cons : no reduction in dimensionality of the problem.
A conjecture

Let us consider the complex version of the problem

\[ B = \arg \min_{\Lambda(X) \subseteq \Omega} \| A - X \|_F. \]

Open problem

When \( A \) is a real matrix, is \( B \) also always a real matrix?

Experiments suggest so, at least for \( \Omega = \text{Hurwitz stable} \).

If the answer is yes, then one can also use the complex version of the algorithm for the real case.

\textbf{Pros} : simpler to write; no need to solve the \( 2 \times 2 \) case by hand.

\textbf{Cons} : no reduction in dimensionality of the problem.
Numerical experiments: setup

**Tool** Manopt [Boumal, Mishra, Absil, Sepulchre], a Matlab toolbox.

**Competitors** Various algorithms available on N. Gillis’ home page:

- [Burke, Henrion, Lewis, Overton]: non-smooth quasi-Newton methods
- [Orbandexivry, Nesterov, Van Dooren]: convex approximation
- [Gillis, Sharma]: reformulation as dissipative Hamiltonian system

**Not** in these experiments, but some remarks later:

- [Guglielmi, Lubich, Manetta, Protasov]: reformulation as a system of ODEs (arguably the best algorithm available so far).

All algorithms promise only local minima.
Numerical experiments: results

type = 1, n = 10

CPU time / s
\|A - X_k\|_F
type = 1, n = 10

CPU time / s
\|A - X_k\|_F
type = 2, n = 10

CPU time / s
\|A - X_k\|_F
type = 3, n = 10

CPU time / s
\|A - X_k\|_F
type = 4, n = 10

BCD Grad FGM SuccConv BFGS Orth

CPU time / s
\|A - X_k\|_F
type = 1, n = 20

CPU time / s
\|A - X_k\|_F
type = 2, n = 20

CPU time / s
\|A - X_k\|_F
type = 3, n = 20

CPU time / s
\|A - X_k\|_F
type = 4, n = 20
Numerical experiments: results

- Type 1, n = 50
- Type 2, n = 50
- Type 3, n = 50
- Type 4, n = 50

- Type 1, n = 100
- Type 2, n = 100
- Type 3, n = 100
- Type 4, n = 100

CPU time / s

∥A − X_k∥_F

type = 1, n = 50
BCD Grad FGM SuccConv BFGS Orth

type = 2, n = 50
BCD Grad FGM SuccConv BFGS Orth

type = 3, n = 50
BCD Grad FGM SuccConv BFGS Orth

type = 4, n = 50
BCD Grad FGM SuccConv BFGS Orth

F. Poloni (U Pisa)
Numerical experiments: quality of local minima found

Figure: Performance profile of the values of $\|A - X\|_F$ obtained by the algorithms on 100 random $10 \times 10$ matrices (equal split of `rand` and `randn`).
Multiple eigenvalues

Empirical observation: often the other algorithms (especially BCD and Grad) cannot find local minima with multiple zero eigenvalues.

Eigenvalues of minimizer $B$ for a random $6 \times 6$ matrix $A$

Related: in Orth, $\text{diag}(T)$ gives multiple eigenvalues much more accurately than $\text{eig}(B)$ (accuracy $u^{1/k}$ from perturbation theory).
Comparison with ODE approach

No extensive comparison yet with ODE approach [Guglielmi, Lubich] (due to code availability).

😄 On a difficult small example (30 × 30 Grcar matrix), we seem to win both in terms of CPU time and quality of minimum $\|A - B\|_F$ (5.65 vs 6.50, by finding a minimizer with a pair of complex conjugate eigenvalues of multiplicity 14!).

😢 On large-scale problems (e.g. one with $n = 800$), the optimizer from Manopt does not converge.

😢 ODE method can handle various matrix structures and we cannot.
Conclusions

- New framework to attack nearest-stable-matrix problems via optimization on matrix manifolds.
- Avoids some of the main troubles with the problem: tricky feasible region, numerical difficulties with eigenvalue computation.
- Great numerical results for small matrices. Still work needed for larger matrices ($n \approx 100 - 1000$).
- Design space to explore: choose good initial value; fine-tune the optimization method; borrow tricks from eigensolvers.
- The approach works for a generic $\Omega$, and can be generalized to variants (e.g., nearest matrix with at least $k$ eigenvalues in $\Omega$).

Thanks for your attention!
Conclusions

- New framework to attack nearest-stable-matrix problems via optimization on matrix manifolds.
- Avoids some of the main troubles with the problem: tricky feasible region, numerical difficulties with eigenvalue computation.
- Great numerical results for small matrices. Still work needed for larger matrices ($n \approx 100 – 1000$).
- Design space to explore: choose good initial value; fine-tune the optimization method; borrow tricks from eigensolvers.
- The approach works for a generic $\Omega$, and can be generalized to variants (e.g., nearest matrix with at least $k$ eigenvalues in $\Omega$).

Thanks for your attention!