# Reduced order modeling and numerical linear algebra

Akil Narayan[1]

[1] Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

February 7, 2020
ICERM

# Continuous ↔ discrete analogies

Most standard techniques for reduced basis methods can be understood from numerical linear algebra.

Kolmogorov $n$ widths ↔ Singular value decompositions

Reduced basis methods ↔ $QR$ decompositions

Empirical interpolation methods ↔ $LU$ decompositions

Kolmogorov $n$ widths are (essentially) singular values

# Singular value decompositions

Let $\boldsymbol{A} \in \mathbb{R}^{M \times N}$, with $M \gg N$.

We will think of the columns of $\boldsymbol{A}$ as snapshots.

$$\boldsymbol{A} := \left( \begin{array}{cccc} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ | & | & & | \end{array} \right)$$

The SVD of $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal $M \times M$ and $N \times N$ matrices, respectively.
$\boldsymbol{\Sigma}$ is a diagonal matrix with non-negative entries.

We'll use the following non-standard notation for the entries in $\boldsymbol{\Sigma}$:

$$\sigma_0 \geqslant \sigma_1 \geqslant \cdots \geqslant \sigma_{N-1}.$$

# Low-rank approximations

Among the nice properties of the SVD is its ability to form low-rank approximations,

$$\boldsymbol{A}_k := \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^T, \qquad\qquad 1 \leqslant k \leqslant N,$$

where $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ are $k$-column truncations, and $\boldsymbol{\Sigma}_k$ is a $k \times k$ principcal submatrix truncation.

With $\operatorname{rank}(\boldsymbol{A}_k) = k$, then

$$\boldsymbol{A}_k = \operatorname*{arg\,min}_{\operatorname{rank}(\boldsymbol{B}) \leqslant k} \left\| \boldsymbol{A} - \boldsymbol{B} \right\|_* ,$$

for $* = 2, F$.

# Low-rank approximations

Among the nice properties of the SVD is its ability to form low-rank approximations,

$$\boldsymbol{A}_k := \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^T, \qquad\qquad 1 \leqslant k \leqslant N,$$

where $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ are $k$-column truncations, and $\boldsymbol{\Sigma}_k$ is a $k \times k$ principcal submatrix truncation.

With $\mathrm{rank}(\boldsymbol{A}_k) = k$, then

$$\boldsymbol{A}_k = \underset{\mathrm{rank}(\boldsymbol{B}) \leqslant k}{\arg\min} \left\| \boldsymbol{A} - \boldsymbol{B} \right\|_* ,$$

for $* = 2, F$.

Equivalently, $\boldsymbol{A}_k$ is the projection of the columns of $\boldsymbol{A}$ onto $R(\boldsymbol{U}_k)$:

$$\boldsymbol{A}_k = \left( \begin{array}{cccc} | & | & & | \\ P_{R(\boldsymbol{U}_k)} \boldsymbol{a}_1 & P_{R(\boldsymbol{U}_k)} \boldsymbol{a}_2 & \cdots & P_{R(\boldsymbol{U}_k)} \boldsymbol{a}_N \\ | & | & & | \end{array} \right)$$

# Projections onto arbitrary spaces

What if we project $\boldsymbol{A}$ onto other spaces?

If $V \subset \mathbb{R}^M$ is any subspace, we could consider

$$\boldsymbol{P}_V \boldsymbol{A} := \begin{pmatrix} | & | & & | \\ \boldsymbol{P}_V \boldsymbol{a}_1 & \boldsymbol{P}_V \boldsymbol{a}_2 & \cdots & \boldsymbol{P}_V \boldsymbol{a}_N \\ | & | & & | \end{pmatrix}$$

# Projections onto arbitrary spaces

What if we project $\boldsymbol{A}$ onto other spaces?

If $V \subset \mathbb{R}^M$ is any subspace, we could consider

$$\boldsymbol{P}_V \boldsymbol{A} := \begin{pmatrix} | & | & & | \\ \boldsymbol{P}_V \boldsymbol{a}_1 & \boldsymbol{P}_V \boldsymbol{a}_2 & \cdots & \boldsymbol{P}_V \boldsymbol{a}_N \\ | & | & & | \end{pmatrix}$$

And we could ask about a certain type of error committed by this approximation

$$E(V) := \max_{\|x\|_2 = 1} \|\boldsymbol{A}x - \boldsymbol{P}_V \boldsymbol{A}x\|_2$$

We know $V = R(\boldsymbol{U}_k)$ does a pretty good job. What about other spaces?

# Optimal projections

For a given rank $k$, an "optimal" projection commits the smallest error:

$$E_k := \min_{V \subset \mathbb{R}^M} E(V)$$

# Optimal projections

For a given rank $k$, an "optimal" projection commits the smallest error:

$$E_k := \min_{V \subset \mathbb{R}^M} E(V)$$

So an extremal characterization of an SVD-based low rank approximation is

$$R(\boldsymbol{U}_k) = \arg\min_{V \subset \mathbb{R}^N} \max_{\|x\|_2 = 1} \left\| \boldsymbol{A} x - \boldsymbol{P} \boldsymbol{A} x \right\|_2 .$$

# Optimal projections

For a given rank $k$, an "optimal" projection commits the smallest error:

$$E_k := \min_{V \subset \mathbb{R}^M} E(V)$$

So an extremal characterization of an SVD-based low rank approximation is

$$R(\boldsymbol{U}_k) = \arg\min_{V \subset \mathbb{R}^N} \max_{\|x\|_2 = 1} \left\| \boldsymbol{A}x - \boldsymbol{P}\boldsymbol{A}x \right\|_2 .$$

Or, an (unnecessarily?) pedantic alternative:

$$E_k = \sigma_k(\boldsymbol{A}) = \min_{V \subset \mathbb{R}^N} \max_{\|x\|_2 = 1} \min_{v \in V} \left\| Ax - v \right\|_2$$

# SVD projections

Given $\boldsymbol{A} \in \mathbb{R}^{M \times N}$, the success of a low-rank projection is dictated by the approximation numbers

$$\sigma_k(\boldsymbol{A}) = \min_{V \subset \mathbb{R}^N} \max_{\|x\|_2 = 1} \min_{v \in V} \|Ax - v\|_2 .$$

More precisely, it is dictated by fast *decay* of these numbers as $k$ increases.

# SVD projections

Given $\boldsymbol{A} \in \mathbb{R}^{M \times N}$, the success of a low-rank projection is dictated by the approximation numbers

$$\sigma_k(\boldsymbol{A}) = \min_{V \subset \mathbb{R}^N} \max_{\|x\|_2 = 1} \min_{v \in V} \|Ax - v\|_2 .$$

More precisely, it is dictated by fast *decay* of these numbers as $k$ increases.

These numbers are defined by our choice of metric on "output" space $\mathbb{R}^M$, and our choice of metric on "measurement" space $\mathbb{R}^N$.

I.e., a generalization might look like

$$\sigma_k\left(\boldsymbol{A}; \ell^p\left(\mathbb{R}^M\right), \ell^q\left(\mathbb{R}^N\right)\right) = \min_{\dim V \leqslant k} \max_{\|x\|_q = 1} \min_{v \in V} \|Ax - v\|_p .$$

# Kolmogorov $n$ widths

$$\sigma_n\left(\boldsymbol{A}; \ell^p\left(\mathbb{R}^M\right), \ell^q\left(\mathbb{R}^N\right)\right) = \min_{\dim V \leqslant n} \max_{\|\boldsymbol{x}\|_q = 1} \min_{\boldsymbol{v} \in V} \left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{v}\right\|_p .$$

These numbers tell us how well the columns of $\boldsymbol{A}$ are $\ell^p$-approximated by a linear space using $\ell^q$ measurements.

Another definition might be the maximum column norm error:

$$\sigma_n\left(\boldsymbol{A}; \ell^p\left(\mathbb{R}^M\right)\right) = \min_{\dim V \leqslant n} \max_{i \in [N]} \min_{v \in V} \left\|\boldsymbol{A}\boldsymbol{e}_i - \boldsymbol{v}\right\|_p .$$

Great. How do we do all this with functions?

# Kolmogorov $n$ widths

$$\sigma_n\left(\boldsymbol{A}; \ell^p\left(\mathbb{R}^M\right), \ell^q\left(\mathbb{R}^N\right)\right) = \min_{\dim V \leqslant n} \max_{\|\boldsymbol{x}\|_q = 1} \min_{\boldsymbol{v} \in V} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{v}\|_p .$$

These numbers tell us how well the columns of $\boldsymbol{A}$ are $\ell^p$-approximated by a linear space using $\ell^q$ measurements.

Another definition might be the maximum column norm error:

$$\sigma_n\left(\boldsymbol{A}; \ell^p\left(\mathbb{R}^M\right)\right) = \min_{\dim V \leqslant n} \max_{i \in [N]} \min_{v \in V} \|\boldsymbol{A}\boldsymbol{e}_i - \boldsymbol{v}\|_p .$$

Great. How do we do all this with functions?

Let $\mathcal{A}$ be a collection of functions in a Hilbert space $\mathcal{H}$.

Then one way to talk about similar concepts to $(\ell^2)$ singular values is

$$\sigma_n(\mathcal{A}; \mathcal{H}) = \inf_{\dim V \leqslant n} \sup_{a \in \mathcal{A}} \inf_{v \in V} \|a - v\|$$

This is called the Kolmogorov $n$ width of $\mathcal{A}$ (with respect to $\mathcal{H}$).

Reduced basis methods (essentially) perform $QR$ decompositions

# Interpolative decompositions

One disadvantage of SVD-based low rank approximations,

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ | & | & & | \end{pmatrix} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

is that we need information from all columns of $\boldsymbol{A}$ to define $\boldsymbol{U}$.

# Interpolative decompositions

One disadvantage of SVD-based low rank approximations,

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \\ | & | & & | \end{pmatrix} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

is that we need information from all columns of $\boldsymbol{A}$ to define $\boldsymbol{U}$.

One alternative: Interpolative decompositions, or matrix skeletonizations.

Basic idea: project all columns of $\boldsymbol{A}$ onto a subspace spanned by a few columns.

A rank-$n$ column skeletonization of $\boldsymbol{A}$ is

$$\boldsymbol{B} = \underbrace{\boldsymbol{A}_S \left(\boldsymbol{A}_S^T \boldsymbol{A}_S\right)^\dagger \boldsymbol{A}_S^T}_{\boldsymbol{P}_{R(\boldsymbol{A}_S)}} \boldsymbol{A}, \qquad \boldsymbol{A}_S := \boldsymbol{A} \begin{pmatrix} | & | & & | \\ \boldsymbol{e}_{s_1} & \boldsymbol{e}_{s_2} & \cdots & \boldsymbol{e}_{s_n} \\ | & | & & | \end{pmatrix},$$

with $S = \{s_1, \ldots s_n\} \subset [N]$.

# Choosing the columns $S$

The problem of choosing $S$ that is optimal in some metric is the *column subset selection* problem.

For metrics of interest, it's NP-hard.

# Choosing the columns $S$

The problem of choosing $S$ that is optimal in some metric is the *column subset selection* problem.

For metrics of interest, it's NP-hard.

So let's do something else: Let's pick columns greedily:

Given $S \subset [N]$ of size $n$, we'll add a column index via the procedure

$$s_{n+1} = \operatorname*{arg\,max}_{j \in [N]} \left\| \boldsymbol{a}_j - \boldsymbol{P}_{R(\boldsymbol{A}_S)} \boldsymbol{a}_j \right\|_2 .$$

This is much cheaper since I need only to evaluate $N$ vector norms at each step.

# Choosing the columns $S$

The problem of choosing $S$ that is optimal in some metric is the *column subset selection* problem.

For metrics of interest, it's NP-hard.

So let's do something else: Let's pick columns greedily:

Given $S \subset [N]$ of size $n$, we'll add a column index via the procedure

$$s_{n+1} = \underset{j \in [N]}{\arg\max} \left\| \boldsymbol{a}_j - \boldsymbol{P}_{R(\boldsymbol{A}_S)} \boldsymbol{a}_j \right\|_2 .$$

This is much cheaper since I need only to evaluate $N$ vector norms at each step.

There's already a well-polished algorithm that does this: the $QR$ decomposition.

# The $QR$ decomposition (1/2)

The column-pivoted $QR$ decomposition iteratively computes orthonormal vectors in the range of $\boldsymbol{A}$.

At step $j$, the next column is identified as the one whose projected residual is largest.

$$\begin{aligned} \boldsymbol{P}_{j-1} &:= \boldsymbol{Q}_{j-1}\boldsymbol{Q}_{j-1}^T \\ s_j &= \operatorname*{arg\,max}_{j\in[N]} \left\|\boldsymbol{a}_j - \boldsymbol{P}_{j-1}\boldsymbol{a}_j\right\|_2 \\ \boldsymbol{q}_j &:= \frac{\boldsymbol{a}_{s_j}}{\|\boldsymbol{a}_{s_j}\|_2}, \quad \boldsymbol{Q}_j = \left[\boldsymbol{Q}_{j-1}, \quad \boldsymbol{q}_j\right] \end{aligned}$$

# The $QR$ decomposition (1/2)

The column-pivoted $QR$ decomposition iteratively computes orthonormal vectors in the range of $\boldsymbol{A}$.

At step $j$, the next column is identified as the one whose projected residual is largest.

$$\begin{aligned}
\boldsymbol{P}_{j-1} &:= \boldsymbol{Q}_{j-1}\boldsymbol{Q}_{j-1}^T \\
s_j &= \underset{j \in [N]}{\arg\max} \left\| \boldsymbol{a}_j - \boldsymbol{P}_{j-1}\boldsymbol{a}_j \right\|_2 \\
\boldsymbol{q}_j &:= \frac{\boldsymbol{a}_{s_j}}{\|\boldsymbol{a}_{s_j}\|_2}, \quad \boldsymbol{Q}_j = \left[\boldsymbol{Q}_{j-1}, \quad \boldsymbol{q}_j\right]
\end{aligned}$$

The residual

$$r_{j-1} := \left\| \boldsymbol{a}_{s_j} - \boldsymbol{P}_{j-1}\boldsymbol{a}_{s_j} \right\|_2,$$

is the largest ($\ell^2$-norm) column mistake we make by choosing $S = \{s_1, \ldots, s_{j-1}\}$, i.e., by replacing

$$\boldsymbol{A} \leftarrow \boldsymbol{P}_V \boldsymbol{A}, \qquad\qquad V := \mathrm{span}\{\boldsymbol{a}_{s_1}, \ldots, \boldsymbol{a}_{s_{j-1}}\}.$$

# The $QR$ decomposition (2/2)

This algorithm is a *greedy* algorithm: instead of all-at-once optimization, we optimize one at a time.

Clearly, we don't expect this to perform as well as the optimal SVD-based subspace.

But how well does this greedy procedure work in practice?

# Discrete greedy algorithms

In some cases, this greedy algorithm performs *comparably* to an optimal (SVD) algorithm.

In particular,

$$\sigma_r(\boldsymbol{A}) \lesssim \exp(-br) \quad \Longrightarrow \quad s_j \lesssim \exp(-cr),$$

where $c < b$.[Harbrecht, Peters, Schneider 2010]

# Back to the continuous world

Once more, let's put this into appropriate language for functions.

Let $\mathcal{A}$ be a collection of functions, parameterized by $\mu \in \mathbb{R}^d$,

$$\mathcal{A} = \left\{ u(\mu) \;\middle|\; \mu \in \Gamma \subset \mathbb{R}^d \right\}.$$

# Back to the continuous world

Once more, let's put this into appropriate language for functions.

Let $\mathcal{A}$ be a collection of functions, parameterized by $\mu \in \mathbb{R}^d$,

$$\mathcal{A} = \left\{ u(\mu) \;\middle|\; \mu \in \Gamma \subset \mathbb{R}^d \right\}.$$

A *greedy* (pivoted $QR$!) approach to determining a low-rank space for approximation is

$$\mu_j = \underset{\mu \in \Gamma}{\arg\max} \left\| u(\mu) - \mathcal{P}_{j-1} u(\mu) \right\|,$$

where $\mathcal{P}_{j-1}$ is the projection operator onto $\mathrm{span}\{u(\mu_1), \ldots, u(\mu_{j-1})\}$.

# Back to the continuous world

Once more, let's put this into appropriate language for functions.

Let $\mathcal{A}$ be a collection of functions, parameterized by $\mu \in \mathbb{R}^d$,

$$\mathcal{A} = \left\{ u(\mu) \;\middle|\; \mu \in \Gamma \subset \mathbb{R}^d \right\}.$$

A *greedy* (pivoted $QR$!) approach to determining a low-rank space for approximation is

$$\mu_j = \operatorname*{arg\,max}_{\mu \in \Gamma} \left\| u(\mu) - \mathcal{P}_{j-1} u(\mu) \right\|,$$

where $\mathcal{P}_{j-1}$ is the projection operator onto $\operatorname{span}\{u(\mu_1), \ldots, u(\mu_{j-1})\}$.

This is (essentially) the reduced basis method.

# Residuals?

One disadvantage of SVD-based low rank approximations is that we need all columns of $\boldsymbol{A}$.

("One disadvantage of Kolmogorov $n$-width low rank approximations is that we need all functions in $\mathcal{A}$.")

# Residuals?

One disadvantage of SVD-based low rank approximations is that we need all columns of $\boldsymbol{A}$.

(“One disadvantage of Kolmogorov $n$-width low rank approximations is that we need all functions in $\mathcal{A}$.”)

A “$QR$” approach still requires the residual

$$\mu_j = \underset{\mu \in \Gamma}{\arg\max} \left\| u(\mu) - \mathcal{P}_{j-1} u(\mu) \right\|,$$

which, naively, still requires $u(\mu)$.

# Residuals?

One disadvantage of SVD-based low rank approximations is that we need all columns of $\boldsymbol{A}$.

(“One disadvantage of Kolmogorov $n$-width low rank approximations is that we need all functions in $\mathcal{A}$.”)

A “$QR$” approach still requires the residual

$$\mu_j = \underset{\mu \in \Gamma}{\arg\max} \left\| u(\mu) - \mathcal{P}_{j-1} u(\mu) \right\|,$$

which, naively, still requires $u(\mu)$.

RBM methods get around this in the same way that one can get around knowing exact solutions to linear systems:

$$\boldsymbol{L}_j \boldsymbol{a}_j = \boldsymbol{b}_j \implies \|\boldsymbol{a}_j - \boldsymbol{z}\| \leqslant \frac{1}{\sigma_{\min}(\boldsymbol{L}_j)} \|\boldsymbol{b}_j - \boldsymbol{L}_j \boldsymbol{z}\|_2$$

# RBM and $QR$ decompositions

RBM algorithms perform snapshot-based model reduction via a $QR$ decomposition.

$$\mathcal{L}\left(u(\mu);\mu\right) = b(\mu)$$

$$\Downarrow$$

$$\left\| u(\mu) - \mathcal{P}_{j-1} u(\mu) \right\| \leqslant \frac{1}{\text{``}\sigma_{\min}(\mathcal{L})\text{''}} \left\| b(\mu) - \mathcal{L}\left(\mathcal{P}_{j-1} u(\mu);\mu\right) \right\|_2 \tag{1}$$

# RBM and $QR$ decompositions

RBM algorithms perform snapshot-based model reduction via a $QR$ decomposition.

$$\mathcal{L}\left(u(\mu);\mu\right) = b(\mu)$$

$$\Downarrow$$

$$\left\|u(\mu) - \mathcal{P}_{j-1}u(\mu)\right\| \leqslant \frac{1}{\text{``}\sigma_{\min}(\mathcal{L})\text{''}} \left\|b(\mu) - \mathcal{L}\left(\mathcal{P}_{j-1}u(\mu);\mu\right)\right\|_2 \qquad (1)$$

This residual:

- can be computed *without* computing $u$ if $\mathcal{L}(\cdot;\mu)$ depends on $\mu$ in an *affine* way,
- provides a rigorous bound on error committed if "$\sigma_{\min}(\mathcal{L})$" can be computed (*a posteriori* error estimates)

Even though (1) is only an inequality, this "weak" greedy algorithm still produces a good approximation, assuming the $n$ width decays quickly.
[Binev, Cohen, Dahmen, Devore, Petrova, Wojtaszczyk 2011], [Devore, Petrova, Wojtaszczyk 2013]

Empirical interpolation methods (essentially) perform $LU$ decompositions

# Affine dependence

Many times, $\mathcal{L}$ does not depend on $\mu$ in an affine way.

In particular, $\mathcal{L}$ may contain functions of $\mu$, e.g.,

$$\mathcal{L}(u;\mu) = -\nabla_x \cdot \left(\ell(x;\mu)\nabla_x\right) u.$$

This is affine only if

$$\ell(x;\mu) = \sum_{i=1}^{d} f_i(\mu)\ell_i(x).$$

# Affine dependence

Many times, $\mathcal{L}$ does not depend on $\mu$ in an affine way.

In particular, $\mathcal{L}$ may contain functions of $\mu$, e.g.,

$$\mathcal{L}(u; \mu) = -\nabla_x \cdot (\ell(x;\mu)\nabla_x)\, u.$$

This is affine only if

$$\ell(x;\mu) = \sum_{i=1}^{d} f_i(\mu)\ell_i(x).$$

An affine approximation for $\mathcal{L}$ (i.e., for $\ell$) is often accomplished via *empirical interpolation*.[Barrault, Maday, Nguyen, Patera 2004]

# Empirical interpolation

Once again, let's understand this in the discrete setting:

$$\boldsymbol{L} = \begin{pmatrix} | & | & & | \\ \boldsymbol{\ell}_1 & \boldsymbol{\ell}_2 & \cdots & \boldsymbol{\ell}_N \\ | & | & & | \end{pmatrix}$$

One consequence of continuous problem practicalities: want to avoid computing column norms.

# Empirical interpolation

Once again, let's understand this in the discrete setting:

$$\boldsymbol{L} = \begin{pmatrix} | & | & & | \\ \boldsymbol{\ell}_1 & \boldsymbol{\ell}_2 & \cdots & \boldsymbol{\ell}_N \\ | & | & & | \end{pmatrix}$$

One consequence of continuous problem practicalities: want to avoid computing column norms.

One strategy is an "incomplete" $LU$ factorization. A (complete-pivoting) factorization is

$$\boldsymbol{P}\boldsymbol{L}\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{U},$$

where $\boldsymbol{Z}$ is lower triangular, and $\boldsymbol{P}$ and $\boldsymbol{Q}$ are permutation matrices.

# Empirical interpolation

Once again, let's understand this in the discrete setting:

$$\boldsymbol{L} = \begin{pmatrix} | & | & & | \\ \boldsymbol{\ell}_1 & \boldsymbol{\ell}_2 & \cdots & \boldsymbol{\ell}_N \\ | & | & & | \end{pmatrix}$$

One consequence of continuous problem practicalities: want to avoid computing column norms.

One strategy is an "incomplete" $LU$ factorization. A (complete-pivoting) factorization is

$$\boldsymbol{P}\boldsymbol{L}\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{U},$$

where $\boldsymbol{Z}$ is lower triangular, and $\boldsymbol{P}$ and $\boldsymbol{Q}$ are permutation matrices. An approximation would be an incomplete factorization:

$$\boldsymbol{P}\boldsymbol{L}\boldsymbol{Q} \approx \boldsymbol{Z}_d\boldsymbol{U}_d,$$

where $\boldsymbol{Z}_d$ ($\boldsymbol{U}_d$) is a principal $d$-column (-row) truncation.

# Empirical interpolation

Once again, let's understand this in the discrete setting:

$$\boldsymbol{L} = \begin{pmatrix} | & | & & | \\ \boldsymbol{\ell}_1 & \boldsymbol{\ell}_2 & \cdots & \boldsymbol{\ell}_N \\ | & | & & | \end{pmatrix}$$

One consequence of continuous problem practicalities: want to avoid computing column norms.

One strategy is an "incomplete" $LU$ factorization. A (complete-pivoting) factorization is

$$\boldsymbol{PLQ} = \boldsymbol{ZU},$$

where $\boldsymbol{Z}$ is lower triangular, and $\boldsymbol{P}$ and $\boldsymbol{Q}$ are permutation matrices. An approximation would be an incomplete factorization:

$$\boldsymbol{PLQ} \approx \boldsymbol{Z}_d \boldsymbol{U}_d,$$

where $\boldsymbol{Z}_d$ ($\boldsymbol{U}_d$) is a principal $d$-column (-row) truncation. In the continuous setting, this is called the *empirical interpolation method* (EIM).

$\boldsymbol{P}$: Spatial points for interpolation
$\boldsymbol{Q}$: Parameter values defining snapshots used for spatial interpolation

# Continuous ↔ discrete analogies

Kolmogorov $n$ widths ↔ Singular value decompositions

Reduced basis methods ↔ $QR$ decompositions

Empirical interpolation methods ↔ $LU$ decompositions

# Continuous ↔ discrete analogies

Kolmogorov $n$ widths ↔ Singular value decompositions

Reduced basis methods ↔ $QR$ decompositions

Empirical interpolation methods ↔ $LU$ decompositions

Bonus! *Why* do Kolmogorov $n$ widths decay quickly? (for "nice" problems)

# Polynomial approximations

Recall some complex analysis:

Suppose $f : \mathbb{C} \to \mathbb{C}$ is a holomorphic function in some open disc $D$ of the complex plane.

Let $\Gamma$ be a subset of $D$, with $\bar{\Gamma} \subset D$, and $d(\Gamma, \partial D) \geqslant r$.

# Polynomial approximations

Recall some complex analysis:

Suppose $f : \mathbb{C} \to \mathbb{C}$ is a holomorphic function in some open disc $D$ of the complex plane.

Let $\Gamma$ be a subset of $D$, with $\bar{\Gamma} \subset D$, and $d(\Gamma, \partial D) \geqslant r$.

Then Taylor's theorem implies that if $p$ is the degree-$n$ Taylor polynomial centered around any $z_0 \in \Gamma$ then

$$\sup_{z \in \Gamma} \| f(z) - p(z) \| \lesssim r^{-n}.$$

# Polynomial approximations

Recall some complex analysis:

Suppose $f : \mathbb{C} \to \mathbb{C}$ is a holomorphic function in some open disc $D$ of the complex plane.

Let $\Gamma$ be a subset of $D$, with $\bar{\Gamma} \subset D$, and $d(\Gamma, \partial D) \geqslant r$.

Then Taylor's theorem implies that if $p$ is the degree-$n$ Taylor polynomial centered around any $z_0 \in \Gamma$ then

$$\sup_{z \in \Gamma} \| f(z) - p(z) \| \lesssim r^{-n}.$$

I.e., polynomial approximations are exponentially accurate for smooth functions.

# Parameterized elliptic PDEs (1/2)

Now consider the elliptic PDE

$$-\nabla_x \left(\ell(x;\mu)\nabla_x\right) = b(x;\mu).$$

Suppose $\ell(x;\mu)$ is continuous, is $\mu$-uniformly bounded, depends on $\mu$ in an affine way, and

$$\inf_x \ell(x,;\mu) > r_{\min} > 0,$$

uniformly for $\mu \in \Gamma \subset \mathbb{R}^d$. Let $0 \in \Gamma$.

Then the solution $\mu \mapsto u(\mu)$ exists and is well-defined in some Hilbert space $H$.

# Parameterized elliptic PDEs (1/2)

Now consider the elliptic PDE

$$-\nabla_x \left(\ell(x;\mu)\nabla_x\right) = b(x;\mu).$$

Suppose $\ell(x;\mu)$ is continuous, is $\mu$-uniformly bounded, depends on $\mu$ in an affine way, and

$$\inf_x \ell(x,;\mu) > r_{\min} > 0,$$

uniformly for $\mu \in \Gamma \subset \mathbb{R}^d$. Let $0 \in \Gamma$.

Then the solution $\mu \mapsto u(\mu)$ exists and is well-defined in some Hilbert space $H$. Under these conditions, then

$$\mu \mapsto u(\cdot,\mu)$$

is (complex) differentiable in an open disc $D$, with $\mathrm{dist}(\Gamma, \partial D) \sim r_{\min}$.

In particular, all $\mu$-derivatives of $u$ at $\mu = 0$ exist and are $H$-valued.

# Parameterized elliptic PDEs (2/2)

Since $\mu \mapsto u(\mu)$ is complex differentiable in $\Gamma$ with radius $\mathrm{r}_{\min}$:

Taylor's Theorem guarantees a degree-$n$, $d$-variate polynomial approximation $p_n$ with $N \lesssim n^d$ degrees of freedom such that

$$\sup_{\mu \in O} \|u(\mu) - p_n(z)\| \lesssim r_{\min}^{-n} \sim r_{\min}^{-N^{(1/d)}}$$

Hence, the Kolmogorov width of the manifold of solutions (in $H$) decays in $N$, but suffers the curse of dimensionality.[Cohen, Devore 2015]

# Parameterized elliptic PDEs (2/2)

Since $\mu \mapsto u(\mu)$ is complex differentiable in $\Gamma$ with radius $\mathrm{r}_{\min}$:

Taylor's Theorem guarantees a degree-$n$, $d$-variate polynomial approximation $p_n$ with $N \lesssim n^d$ degrees of freedom such that

$$\sup_{\mu \in O} \|u(\mu) - p_n(z)\| \lesssim r_{\min}^{-n} \sim r_{\min}^{-N^{(1/d)}}$$

Hence, the Kolmogorov width of the manifold of solutions (in $H$) decays in $N$, but suffers the curse of dimensionality.[Cohen, Devore 2015]

In short, Kolmogorov widths decay quickly when $u$ depends *smoothly* on the parameter, but suffer from (classical) approximation limitations.