Decoupled isogeometric discretizations using low-rank approximation

Angelos Mantzaflaris

RICAM, Austrian Academy of Sciences

AANMPDE 10

Paleochora, Crete, Greece

October 5, 2017

Acknowledgements: B. Jüttler, B.N. Khoromskij, U. Langer and F. Scholz

Overview

1. Low-rank function approximation
   - Low-rank spline functions
   - Singular Value Decomposition
   - Adaptive Cross Approximation
   - Decoupling more than two variables
2. Isogeometric analysis on tensor-product geometry mappings
   - Model problem and variational formulation
   - Separation rank and Kronecker rank
   - Computational complexity
3. Benchmarks and applications

Overview

1. Low-rank function approximation
   - Low-rank spline functions
   - Singular Value Decomposition
   - Adaptive Cross Approximation
   - Decoupling more than two variables
2. Isogeometric analysis on tensor-product geometry mappings
   - Model problem and variational formulation
   - Separation rank and Kronecker rank
   - Computational complexity
3. Benchmarks and applications

Low-rank spline functions

- Tensor-product spline function:

$$f(\boldsymbol{x}) = \mathbf{A} : \mathbf{B}(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}\, \beta_i(x_1)\, \beta_i(x_2)$$

  - ❒ Coefficient matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{A} = [a_{ij}]$
  - ❒ Basis matrix $\mathbf{B}(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$
    $\mathbf{B}(\boldsymbol{x}) = \beta(x_1) \otimes \beta(x_2) = \left[\beta_i(x_1)\beta_j(x_2)\right]$

- Rank$-R$ spline function:
  - ❒ The coefficient matrix $\mathbf{A} = \sum_{r=1}^{R} \boldsymbol{u}_r \otimes \boldsymbol{v}_r$, $R \le n$

$$f(\boldsymbol{x}) = \mathbf{A} : \mathbf{B}(\boldsymbol{x}) = \sum_{r=1}^{R} (\boldsymbol{u}_r \otimes \boldsymbol{v}_r) : (\beta(x_1) \otimes \beta(x_2)) = \sum_{r=1}^{R} f_r^{(1)}(x_1)\, f_r^{(2)}(x_2)$$

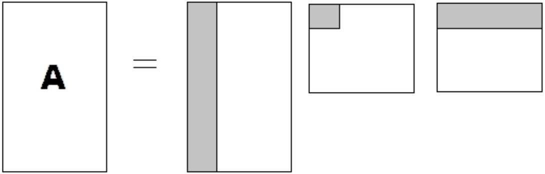
  - ❒ Evaluation at some $\boldsymbol{x}$: $\mathcal{O}(Rdp^2)$ vs $\mathcal{O}(p^{d+1})$

Low rank approximation by SVD

Singular value decomposition for $\mathbf{A} \in \mathbb{R}^{n \times m}$:

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

- $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$, $\sigma_k \geq 0$ and $\mathbf{U}, \mathbf{V}$ orthonormal matrices.
- Optimal approximation of $\mathbf{A}$ by $\mathbf{A}' \in \mathbb{R}^{n \times m}$ of rank R,
  Truncate $\Sigma' = \text{diag}(\sigma_1, \ldots, \sigma_R, 0, \ldots, 0)$

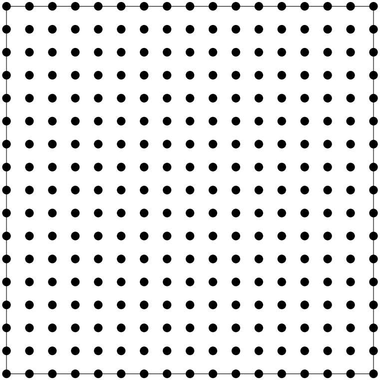


  Complexity $\mathcal{O}(\min(nm^2, mn^2))$, randomized $\mathcal{O}(R^2 \min(n, m))$
- Error estimate [Eckart-Young theorem] for $f(\boldsymbol{x}) = \mathbf{A} : \mathbf{B}(\boldsymbol{x})$,

$$\|f(\boldsymbol{x}) - \mathbf{A}' : \mathbf{B}(\boldsymbol{x})\|^2_{L^\infty} \leq \|A - A'\|^2_F = \sum_{r=R+1}^{n} \sigma_r^2$$

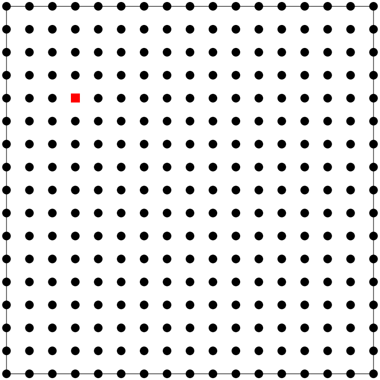
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A}, \; \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \dfrac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



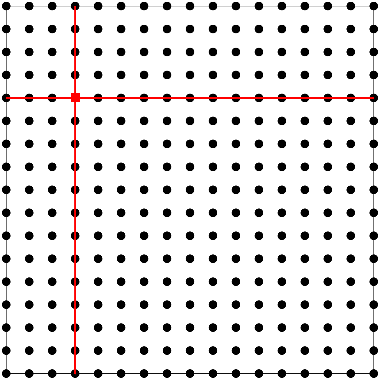
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A},\ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \frac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



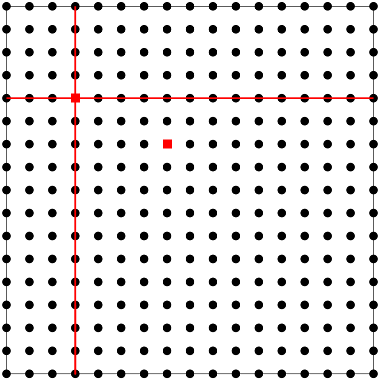
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A},\ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \dfrac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



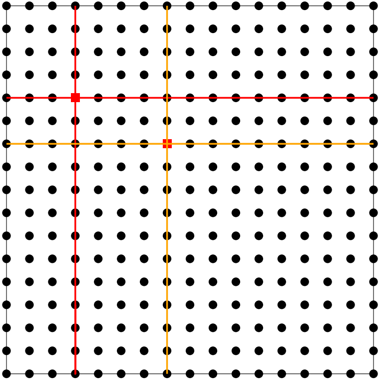
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A}, \ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \frac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



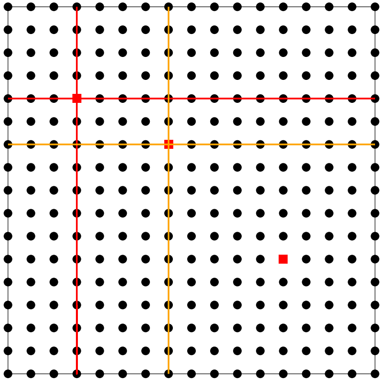
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A},\ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \dfrac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



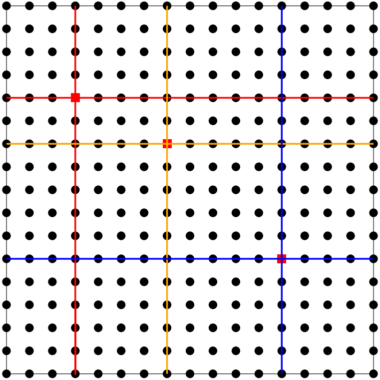
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A},\ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \frac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



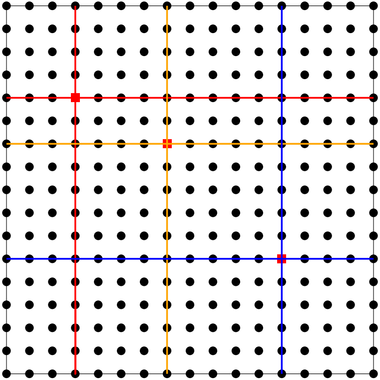
Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A},\ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \dfrac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$



Low rank approximation by ACA

Adaptive Cross Approximation: $\mathbf{E}^{[0]} = \mathbf{A},\ \mathbf{E}^{[r]} = \mathbf{E}^{[r-1]} - \dfrac{\boldsymbol{u}_i^{[r-1]} \otimes \boldsymbol{v}_i^{[r-1]}}{e_{ij}^{[r-1]}}$

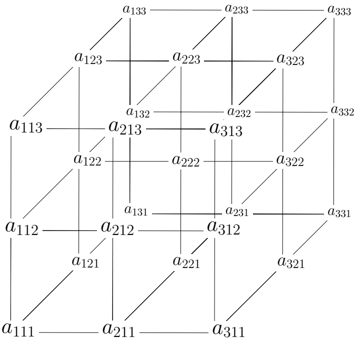


- $\mathbf{A} \approx U C^{-1} V$
- Choose pivots iteratively until $\|E^{[r]}\| \approx 0$
  - Full pivoting $\mathcal{O}(nm)$
  - Partial pivoting $\mathcal{O}(R \min(n, m))$
- Maximum-volume choice is optimal:

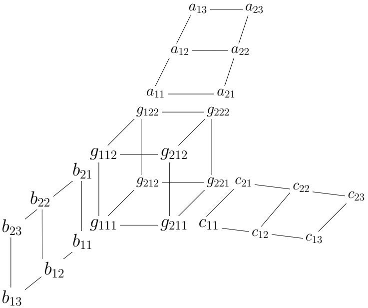
$$C^* = \underset{C \text{ submat. } \mathbf{A}}{\operatorname{argmax}} \left(|\det C| \right)$$

Low rank approximation : 3D, 4D, ...

Full tensor:



Tucker: generalizes matrix SVD (core+side matrices)

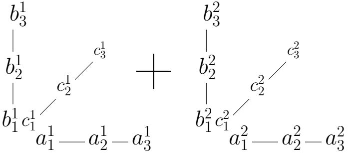


Algorithms for
low rank tensor approximation:

- Higher-order SVD
- Alternating least squares (ALS):

$$\min \|\mathbf{A} - \mathbf{A}'\|_F$$

Canonical: sum of rank-1 tensors (skeleton vectors)

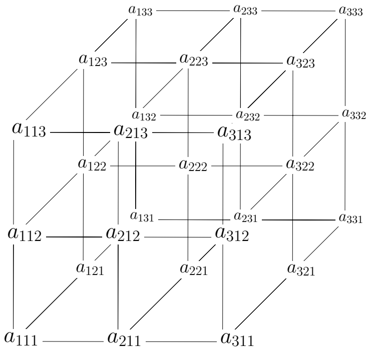


Partial decoupling for more than two variables

- Volumetric spline
  $f(\boldsymbol{x}) = \mathbf{A} : \mathbf{B}(\boldsymbol{x})$

- Tensor of order 3
  $\mathbf{A} = [a_{ijk}]$

  $\mathbf{B} = \beta(x_1) \otimes \beta(x_2) \otimes \beta(x_3)$



Unfoldings:

$$A^{(1)} = \begin{bmatrix} a_{111} & \cdots & a_{133} \\ a_{211} & \cdots & a_{233} \\ a_{311} & \cdots & a_{333} \end{bmatrix}, A^{(2)} = \begin{bmatrix} a_{111} & \cdots & a_{313} \\ a_{121} & \cdots & a_{323} \\ a_{131} & \cdots & a_{333} \end{bmatrix}, A^{(3)} = \begin{bmatrix} a_{111} & \cdots & a_{331} \\ a_{112} & \cdots & a_{332} \\ a_{113} & \cdots & a_{333} \end{bmatrix}$$

$$\sum_{r=1}^{R_1} f_r^{(1)}(x_1) f_r^{(23)}(x_2, x_3), \quad \sum_{r=1}^{R_2} f_r^{(2)}(x_2) f_r^{(13)}(x_1, x_3), \quad \sum_{r=1}^{R_3} f_r^{(3)}(x_3) f_r^{(12)}(x_1, x_2)$$

Overview

1. Low-rank function approximation
   - Low-rank spline functions
   - Singular Value Decomposition
   - Adaptive Cross Approximation
   - Decoupling more than two variables

2. Isogeometric analysis on tensor-product geometry mappings
   - Model problem and variational formulation
   - Separation rank and Kronecker rank
   - Computational complexity

3. Benchmarks and applications

Model problem and variational formulation

- Differential operator

$$Lu = -\nabla \cdot (A(x)\nabla u) + c(x)u.$$

- Boundary value problem $G : \hat{\Omega} \to \Omega$,

$$\begin{cases} Lu &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{cases} \tag{1}$$

- Pull-back to $\hat{\Omega} = [0,1]^d$:

$$\begin{cases} \hat{L}\hat{u} &= \hat{f} \quad \text{in } \hat{\Omega}, \\ \hat{u} &= 0 \quad \text{on } \partial\hat{\Omega}. \end{cases}$$

$$\hat{L}\hat{u} = |\det J_G| L\left(\hat{u} \circ G^{-1}\right) = -\hat{\nabla} \cdot (K\hat{\nabla}\hat{u}) + \omega\hat{u},$$

$$\text{where } \begin{cases} K &= |\det(J_G)|\, J_G^{-1} A J_G^{-T}, \\ \omega &= |\det(J_G)|\, c, \\ \hat{f} &= |\det J_G|\, f \circ G \end{cases}$$

Separation rank and Kronecker rank

Consider the bilinear form associated to $\hat{L}$:

$$a(\hat{u}, \hat{v}) = \int_{\hat{\Omega}} \nabla \hat{u}^{\mathsf{T}} \cdot \boldsymbol{K} \cdot \nabla \hat{v} + \int_{\hat{\Omega}} \hat{u}\, \hat{v}\, \omega$$

- we need to evaluate the matrix $A_{ij} = a(B_i, B_j)$.
- tensor-product basis: $B_i = \beta_{i_1} \cdots \beta_{i_d}$.

$$a(B_i, B_j) \approx \sum_{i=1}^{R} a_r^{(1)}(\beta_{i_1}, \beta_{j_1}) \cdots a_r^{(d)}(\beta_{i_d}, \beta_{j_d})$$

This representation implies that the matrix A can be written in Kronecker product format:

$$A \approx \sum_{r=1}^{R} A_r^{(1)} \otimes \cdots \otimes A_r^{(d)}$$

We call the integer R the $\varepsilon-$Kronecker rank of A.

The Kronecker format of Galerkin matrices

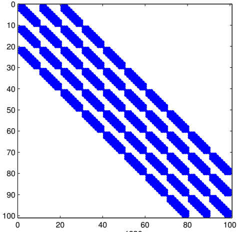
- *d*-D $\rightsquigarrow$ 1-D integrals: $\displaystyle\int_{\hat{\Omega}} \prod_{k=1}^{d} \omega^{(k)}(x_k)\,\mathrm{d}\boldsymbol{x} = \prod_{k=1}^{d} \int_0^1 \omega^{(k)}(x_k)\mathrm{d}x_k$
- Mass matrix:

$$M = \sum_{r=1}^{R} M_r^{(1)} \otimes M_r^{(2)}$$

- Stiffness matrix:

$$S = \sum_{r=1}^{R} S_r^{(1)} \otimes M_r^{(2)} + M_r^{(1)} \otimes S_r^{(2)} + U_r^{(1)} \otimes U_r^{(2)\top} + U_r^{(1)\top} \otimes U_r^{(2)}$$

$\displaystyle\sum_{r=1}^{R}$  $\otimes$  $=$



Computational complexity

- Quadrature with $\mathcal{O}(p)$ nodes per element/direction: $\mathcal{O}(n^d p^{3d})$.
- Lower bound for element-wise strategies: $\mathcal{O}(n^d p^{2d})$
- Optimal complexity with respect to output size $\mathcal{O}(n^d p^d)$.

Using low-rank functions and Kronecker format:

- Tensor decomposition time:
  proportional to the (frequently coarse) geometry mesh
  and the rank parameter. Worst case: $\mathcal{O}(Rdn^d)$.
- "Univariate matrices" (mass, stiffness, advection,..): $\mathcal{O}(Rd\,n\,p^3)$.
- Sparse Kronecker product time $\mathcal{O}(Rn^d p^d)$ $\rightsquigarrow$ dominant cost
- Storage cost: $O(Rdnp)$ for KF vs $O(Rdn^d p^d)$
- Matrix-vector product cost: $O(Rdpn^d)$ for KF vs $O(p^d n^d)$

Error considerations

consistency error = error caused by numerical integration

Total (consistency) error = approximation error + separation error

- Spline projection error in $\Pi(f) = \mathbf{C} : \mathbf{B}$

$$\|f - \Pi(f)\|_\infty \leq \varepsilon_{f,\Pi}$$

- Overall (consistency) error

$$\|f - T_{\mathbb{S}}(\mathbf{C}) : \mathbf{B}\|_\infty \leq \varepsilon_{f,\Pi} + \tau_{\mathbf{C}}$$

eg. $\tau_{\mathbf{C}} \leq \sqrt{\sum_{r>R} \sigma_r^2}$ where σ_r are the singular values of $\mathbf{C}$.

- Both $\varepsilon_{f,\Pi}$ and $\tau_{\mathbf{C}}$ can be set to required tolerance,
- Knowledge of properties of f allows for a wise choice of Π, eg. $\boldsymbol{K}$, ω have known degree/continuity.

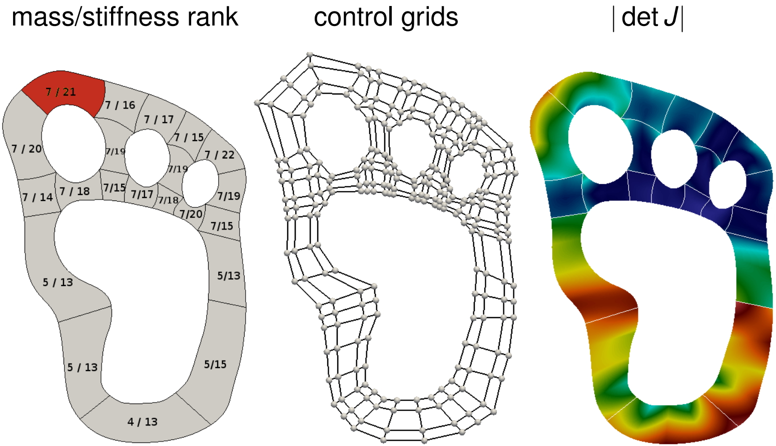
Overview

1. Low-rank function approximation
   - Low-rank spline functions
   - Singular Value Decomposition
   - Adaptive Cross Approximation
   - Decoupling more than two variables

2. Isogeometric analysis on tensor-product geometry mappings
   - Model problem and variational formulation
   - Separation rank and Kronecker rank
   - Computational complexity

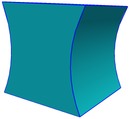
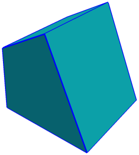
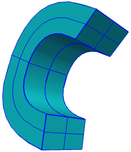
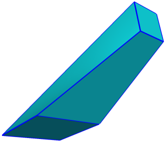
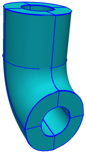
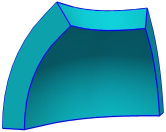
3. Benchmarks and applications

1. Tensor decomposition and rank profiles 2D

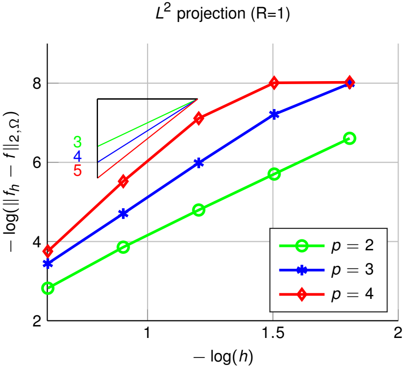


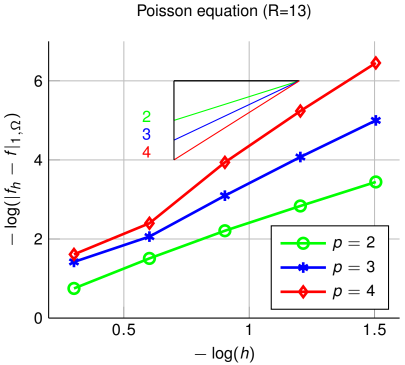
- The rank values lead to significant reduced computation costs
- Rank values can be optimized even further by re-parameterization

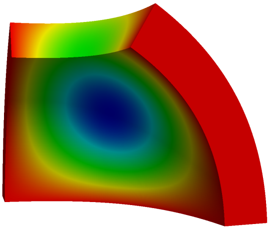
1. Tensor decomposition and rank profiles 3D

Shape	Rank profile $\omega(\hat{\boldsymbol{x}})$	Rank profile $K(\hat{\boldsymbol{x}})$	Shape	Rank profile $\omega(\hat{\boldsymbol{x}})$	Rank profile $K(\hat{\boldsymbol{x}})$
 $\boldsymbol{p} = (1, 1, 2),\ 2 \times 2 \times 3$	1 $6 \times 6 \times 3$	$\begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ $5 \times 5 \times 9$	 $\boldsymbol{p} = (1, 1, 1),\ 2 \times 2 \times 2$	2 $3 \times 3 \times 3$	$\begin{bmatrix} 4 & 4 & 0 \\ 4 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ $5 \times 5 \times 5$
 $\boldsymbol{p} = (2, 2, 2),\ 4 \times 4 \times 4$	1 $11 \times 11 \times 11$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $17 \times 17 \times 17$	 $\boldsymbol{p} = (1, 1, 1),\ 2 \times 2 \times 2$	2 $3 \times 3 \times 3$	$\begin{bmatrix} 3 & 3 & 3 \\ 3 & 8 & 4 \\ 3 & 4 & 4 \end{bmatrix}$ $5 \times 5 \times 5$
 $\boldsymbol{p} = (2, 1, 2),\ 9 \times 2 \times 5$	4 $24 \times 3 \times 12$	$\begin{bmatrix} 6 & 6 & 1 \\ 6 & 6 & 1 \\ 1 & 1 & 9 \end{bmatrix}$ $36 \times 5 \times 18$	 $\boldsymbol{p} = (2, 2, 1),\ 3 \times 3 \times 2$	1 $6 \times 6 \times 3$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 2 \end{bmatrix}$ $9 \times 9 \times 5$

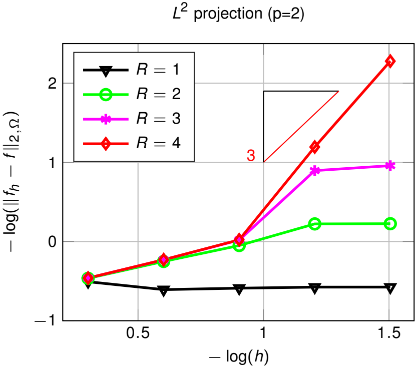
2. Convergence rates

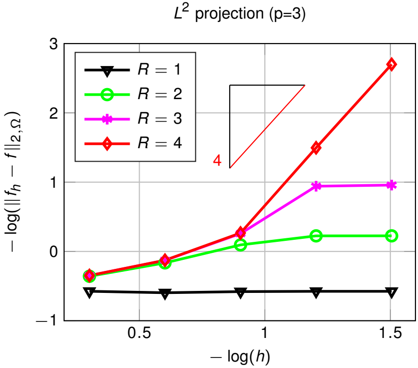


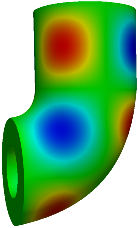




3. Rank truncation



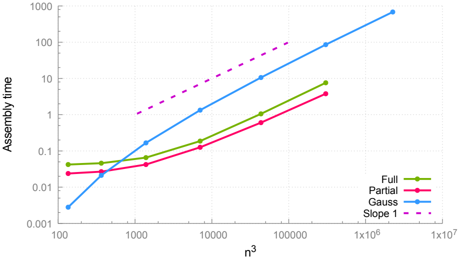




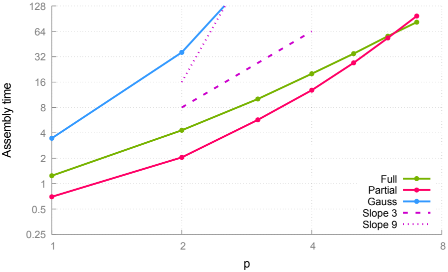
4. Computing times for Stiffness matrix

❐ Computation of “full” matrix

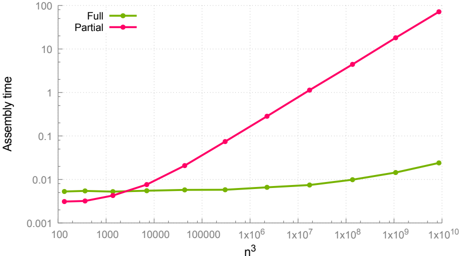
Fix $p = 2$

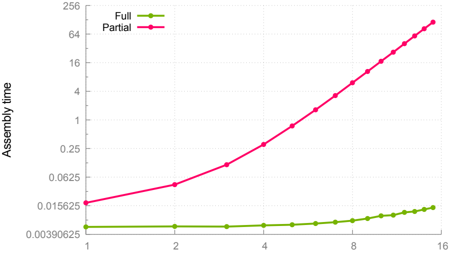


Fix $50 \times 50 \times 50$ DOFs



❐ Computation of Kronecker factors only





Space-time parabolic problem – work in progress

- Application to 3D/4D (space-time) problems of the form

$$\frac{\partial u}{\partial t} - \nabla \cdot (b(x,t)\nabla u) = f \text{ on } Q = \Omega \times [0,T].$$

- Variational form

$$a(u_h, w_h) = \int_{\hat{\Omega}\times[0,\hat{T}]} \hat{b}(\hat{x},\hat{t})\, \nabla_{\hat{x}} u_h \, K \, \nabla_{\hat{x}} w_h \, |J_G| \, d\hat{x}\, d\hat{t}\,,$$

with $w_h = v_h + \theta h \partial_t v_h$, $\quad u_h,\ v_h \in \mathbb{S}_p$.

- $(d+1)$-splitting of the diffusion term

$$\hat{b}(\hat{x},\hat{t}) = \sum_{r=1}^{R} \hat{b}_r^1(\hat{x})\, \hat{b}_r^2(\hat{t})$$

- System matrix for a single patch

$$A = \sum_{r=1}^{R} X^{(r)} \otimes Y^{(r)}$$