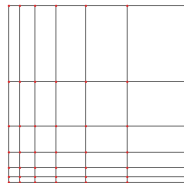


Block encoding for the finite element method with non uniform mesh

Arnaud Rémi, François Damanet, Christophe Geuzaine

University of Liège, Belgium

QTADE 2026



Finite element method in a nutshell

- ▶ Partial differential equation

$$\mathcal{L}[u(\mathbf{x})] = f(\mathbf{x}) \quad \& \quad \text{BCs}$$

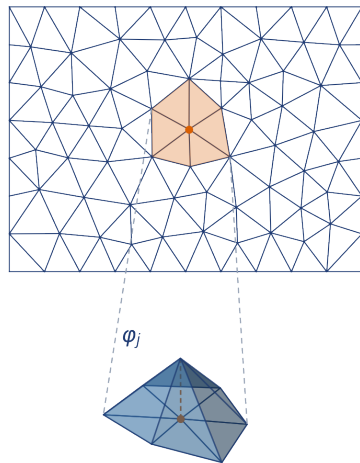
- ▶ Weak formulation

$$\int_{\Omega} (\mathcal{L}[u(\mathbf{x})] - f(\mathbf{x}))v \, d\Omega = 0, \quad \forall v$$

- ▶ Approximation of the solution within a basis of **shape functions**

$$u(\mathbf{x}) \approx \sum u_j \varphi_j(\mathbf{x}), \quad \text{choose } v \in \{\varphi_j\}$$

- ▶ Linear system $A\mathbf{u} = \mathbf{f}$



Example: Poisson problem

$$\begin{aligned} -\nabla^2 u(x, y) &= \sin(x) \sin(y) \quad \text{in } \Omega \\ u(x, y) &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

► **Weak formulation:**

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} \sin(x) \sin(y) v \, d\Omega,$$

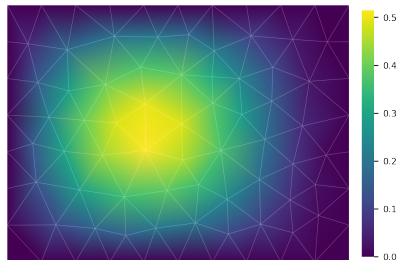
► Choice of **Shape functions**

$$u(\mathbf{x}) \approx \sum_j u_j \varphi_j(\mathbf{x})$$

► linear system $A\mathbf{u} = \mathbf{f}$

$$A_{ij} = \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, d\Omega,$$

$$f_i = \int_{\Omega} f \varphi_i \, d\Omega$$



Classical solvers for $Au = f$

N : unknowns count
 κ : condition number
 ε : precision

Some usual classical solvers :

2D problems		solver	cost
Sparse	direct	LU	$O(N^{3/2})$
Sparse, SPD (Poisson)	iterative	CG	$O(N\sqrt{\kappa} \log \frac{1}{\varepsilon})$
Sparse, indefinite (Helmholtz)	iterative	GMRES	$O(N\kappa \log \frac{1}{\varepsilon})$

→ even for simple problems, floor is $\Omega(N\sqrt{\kappa} \log \frac{1}{\varepsilon})$

Quantum algorithms for linear systems

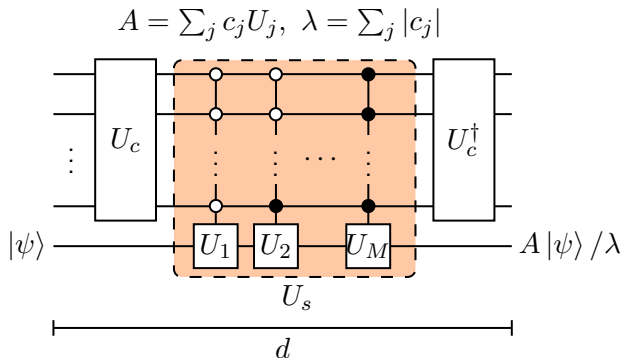
Modern quantum algorithms (QSVT)

- ▶ $O\left(\frac{\lambda}{\|A\|} \kappa \log\left(\frac{1}{\varepsilon}\right)\right)$ calls to A
- ▶ through a **block encoding**

$$U_A = \begin{pmatrix} A/\lambda & * \\ * & * \end{pmatrix}$$

- ▶ total cost:

$$O\left(\underbrace{\frac{\lambda}{\|A\|} d}_{\text{property of } U_A} \kappa \log \frac{1}{\varepsilon}\right)$$



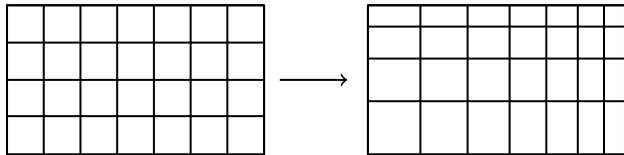
Our work

$$\text{Classical : } O\left(N\sqrt{\kappa} \log \frac{1}{\varepsilon}\right)$$

$$\text{Quantum : } O\left(\frac{\lambda}{\|A\|} d \kappa \log \frac{1}{\varepsilon}\right)$$

Find an **efficient** block encoding for finite elements operator on a **non-uniform** mesh
(graded structured)

$$\left. \begin{array}{l} \frac{\lambda}{\|A\|} = O(1) \\ d = O(\text{poly log } N) \end{array} \right\} \Rightarrow O\left(\text{poly log } N \kappa \log \frac{1}{\varepsilon}\right)$$



Block-encodings

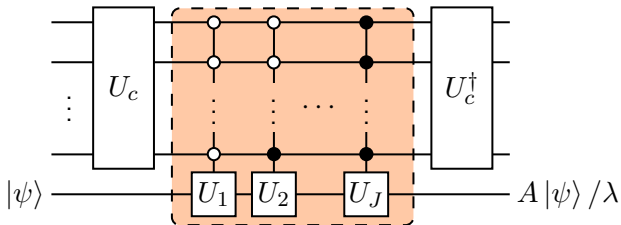
i) Decompose A into a sum of unitaries

$$A = \sum_j c_j U_j$$

ii) Define

$$U_c |0\rangle = |c\rangle, \quad \langle j| U_c |0\rangle = c_j$$

iii) Apply the Prep Sel Prep[†] circuit



Our use case

Helmholtz problem

$$\nabla^2 u + k^2 u = f$$

↓

$$K\mathbf{u} + M\mathbf{u} = \mathbf{f}$$

↓

$$K_{ij} = \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, d\Omega$$

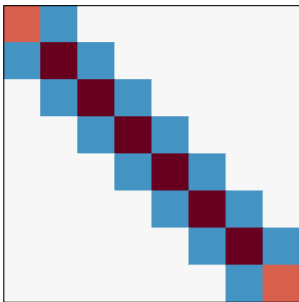
$$M_{ij} = \int_{\Omega} k^2 \varphi_i \varphi_j \, d\Omega$$

$$A_{ij} = K_{ij} + M_{ij}$$

The uniform case

$$A_{ij} = \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j d\Omega$$

*In the uniform case, A has a beautiful **translation invariance** property*



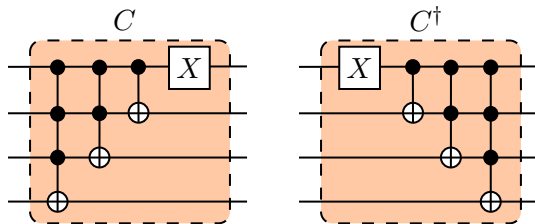
The uniform case

Translation invariance = compact decomposition

$$A = \underbrace{2I}_{\text{depth} = O(1)} + \underbrace{-C - C^\dagger + B \text{ (boundary)}}_{\text{depths} = O(\log N)}$$

The uniform case

$$C|x\rangle = |x + 1 \bmod 2^n\rangle$$



- $O(\log(N)^2)$ deep circuits
- $\exists O(\log N)$ alternatives, with ancillas

The uniform case

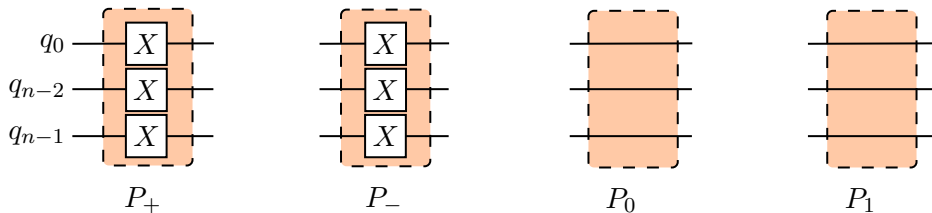
$$\begin{aligned}\sigma_0 &= |0\rangle\langle 0|, \sigma_1 = |1\rangle\langle 1| \\ \sigma_+ &= |1\rangle\langle 0|, \sigma_- = |0\rangle\langle 1|\end{aligned}$$

$$\sigma_j^{\otimes n} = \frac{1}{2}(P_j + R_j), \quad \sigma_j \in \{\sigma_0, \sigma_1, \sigma_+, \sigma_-\}$$

The uniform case

$$\begin{aligned}\sigma_0 &= |0\rangle\langle 0|, \sigma_1 = |1\rangle\langle 1| \\ \sigma_+ &= |1\rangle\langle 0|, \sigma_- = |0\rangle\langle 1|\end{aligned}$$

$$\sigma_j^{\otimes n} = \frac{1}{2}(P_j + R_j), \quad \sigma_j \in \{\sigma_0, \sigma_1, \sigma_+, \sigma_-\}$$

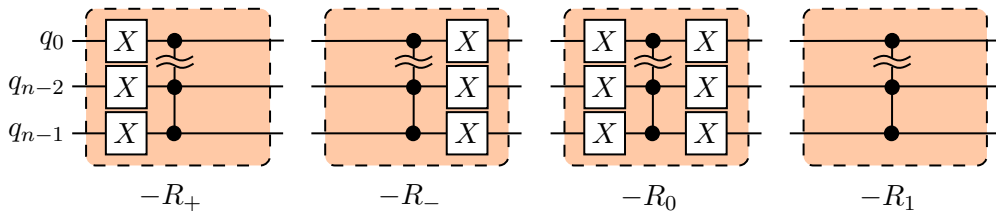


→ $O(1)$ deep circuits

The uniform case

$$\begin{aligned}\sigma_0 &= |0\rangle\langle 0|, \sigma_1 = |1\rangle\langle 1| \\ \sigma_+ &= |1\rangle\langle 0|, \sigma_- = |0\rangle\langle 1|\end{aligned}$$

$$\sigma_j^{\otimes n} = \frac{1}{2}(P_j + R_j), \quad \sigma_j \in \{\sigma_0, \sigma_1, \sigma_+, \sigma_-\}$$



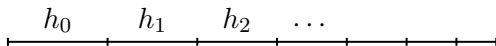
$\rightarrow O(\log N)$ deep circuits

The uniform case

At the end of the day, the block-encoding has

$$\left. \begin{array}{l} d = O(1) + O(\log N) + O(\log N) \\ \lambda/\|A\| = O(1) \end{array} \right\} \Rightarrow O\left(\log N \kappa \log \frac{1}{\varepsilon}\right)$$

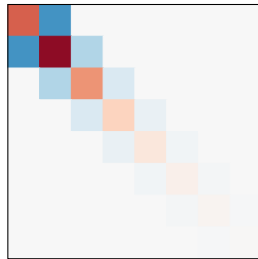
Graded meshes



$$h_j = f(j; \mathbf{q})$$

- ▶ **geometric** : $f(j; h_0, r) = h_0 r^j$
- ▶ **inverse** : $f(j; c, d) = c/(j + d)$
- ▶ **power** : $f(j; k, p) = k j^p$

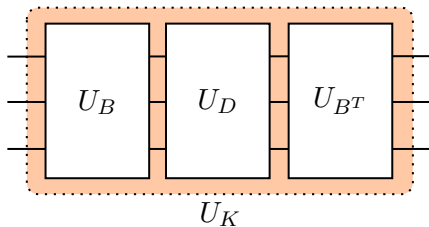
$$A_{ij} = \int_{\Omega} \nabla \varphi_i \nabla \varphi_j \, d\Omega \quad \text{depends on } (h_i, h_j)$$



Grading breaks translation invariance

Graded meshes

$$K = \underbrace{B^\top}_{C-I} \times \underbrace{D}_{\text{diag}(1/h_j)} \times \underbrace{B}_{C^T-I}$$



Encoding D

geometric grading : $h_j = h_0 r^j$

$$\begin{aligned} D &= \text{diag}(1/h_j) = \frac{1}{h_0} \text{diag}(1/r^j) \\ &= \frac{1}{h_0} \begin{bmatrix} 1 & 0 \\ 0 & 1/r^{N/2} \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1/r \end{bmatrix} \end{aligned}$$

CRY(θ)

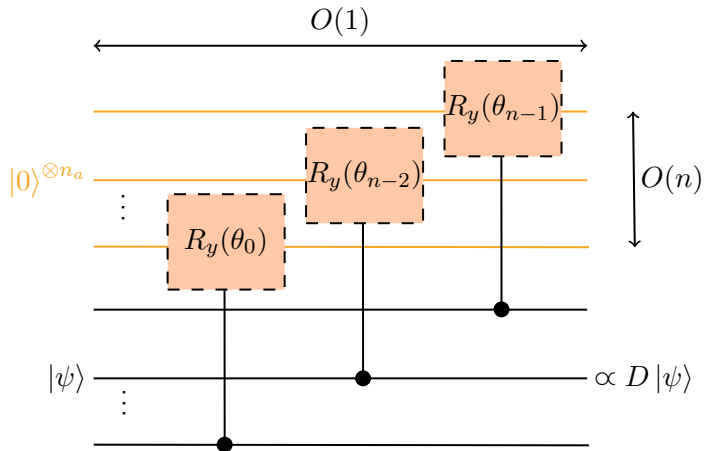
$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta/2) & 0 & -\sin(\theta/2) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\theta/2) & 0 & \cos(\theta/2) \end{bmatrix}}$$

$$\cos(\theta_j/2) = 1/r^j$$

→ CRY **natively** block encodes $\begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix}$

Encoding D

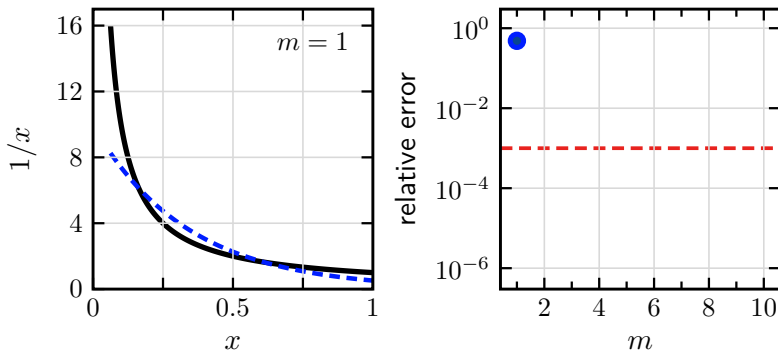
geometric grading : $h_j = h_0 r^j$



Encoding D

Inverse profile:

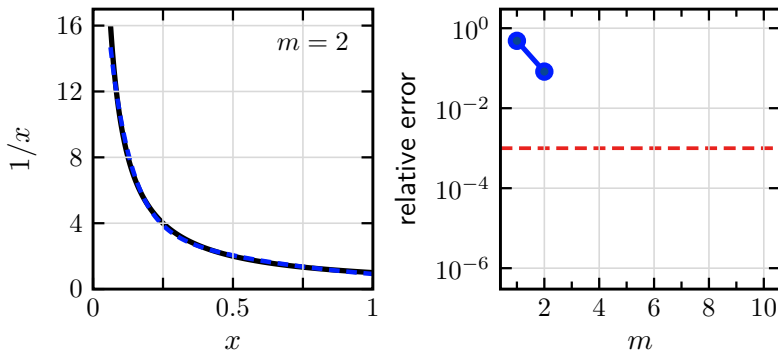
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

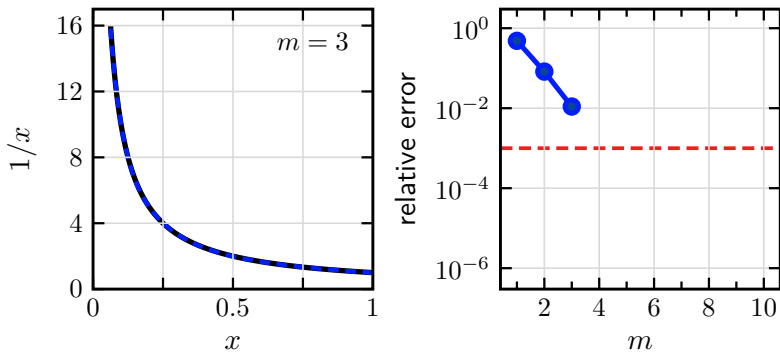
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

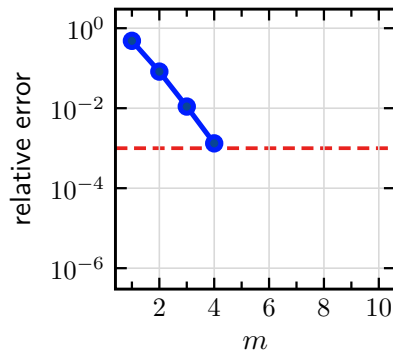
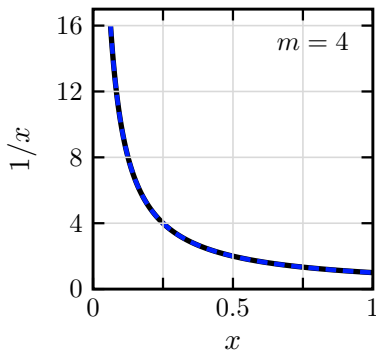
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

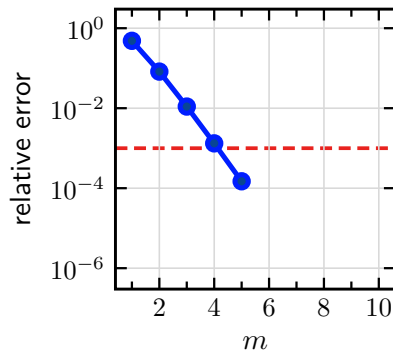
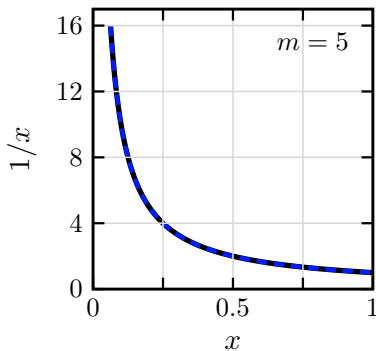
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

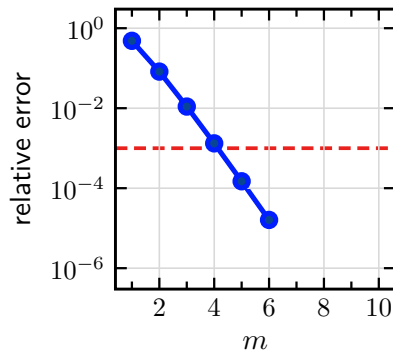
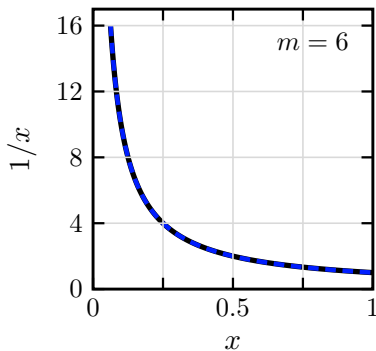
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

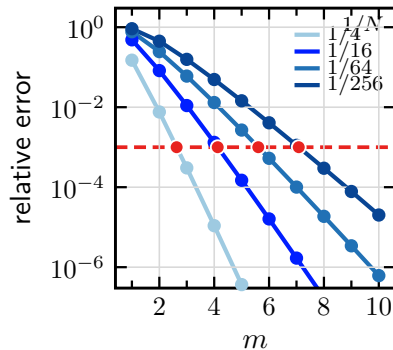
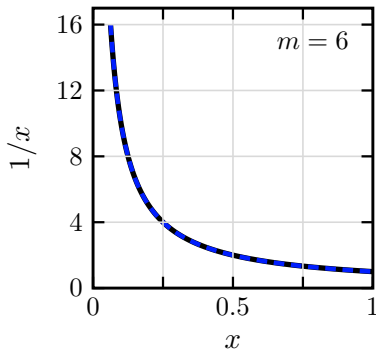
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

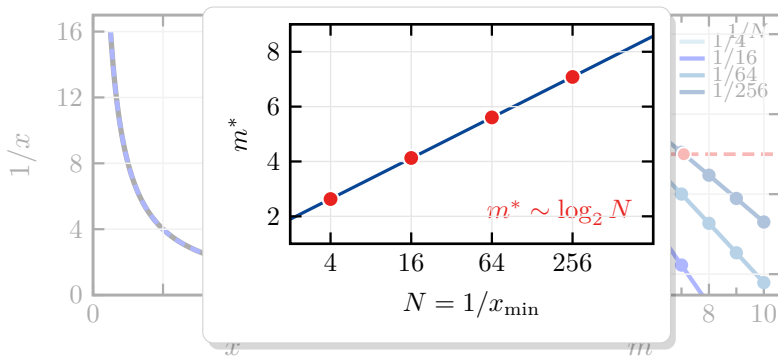
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Inverse profile:

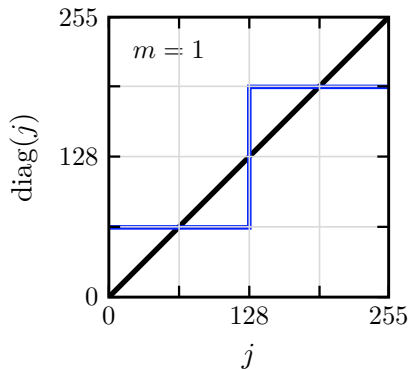
$$D = \text{diag}(1/j) \approx \sum_{k=1}^m c_k \exp(d_k j)$$



Encoding D

Linear profile:

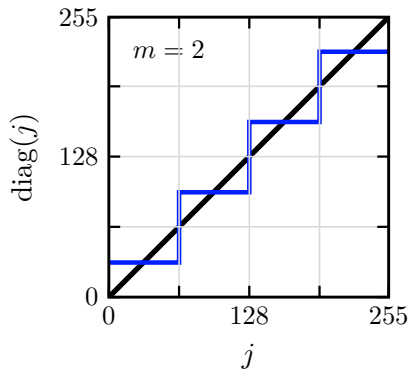
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

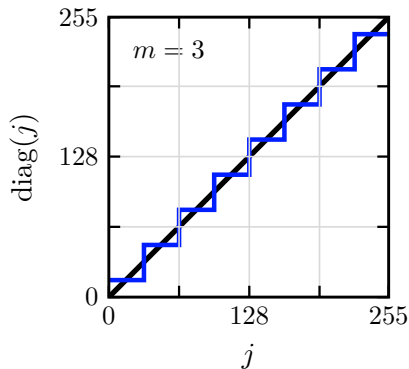
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

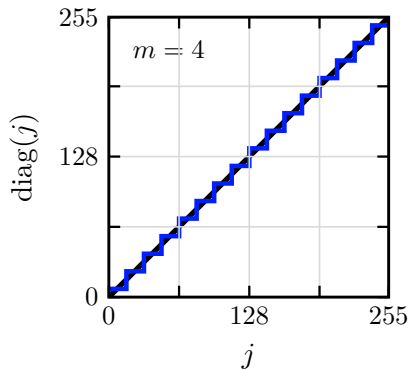
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

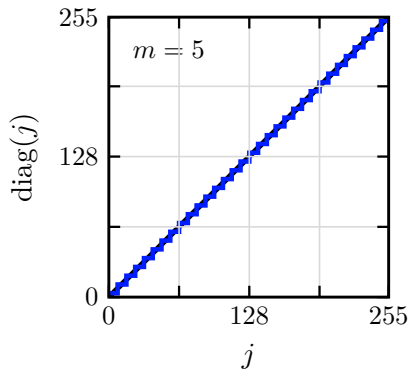
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

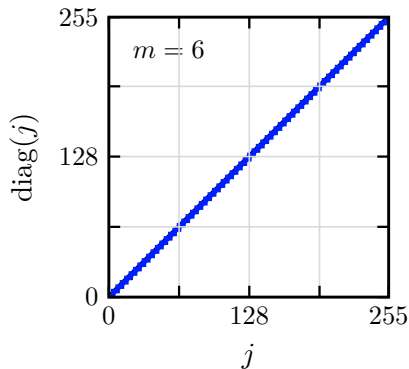
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

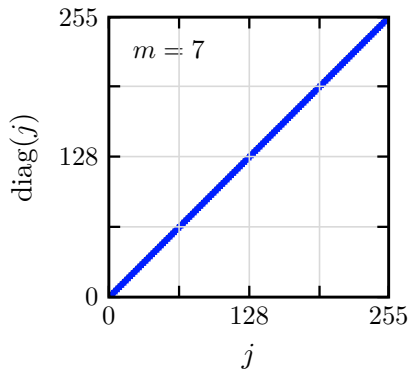
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

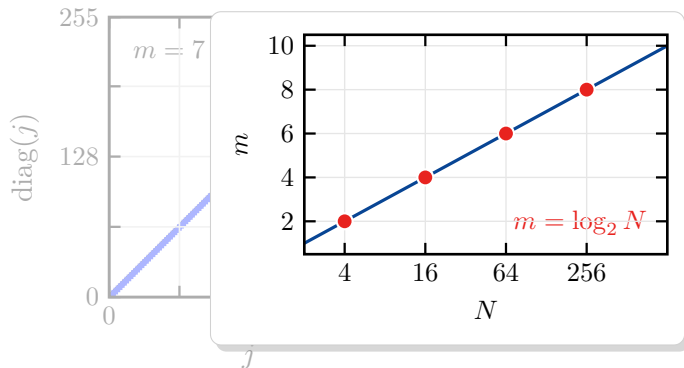
$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$



Encoding D

Linear profile:

$$D = \text{diag}(j) = \frac{N-1}{2}I - \frac{1}{2} \sum_{k=0}^{m-1} 2^k Z^{(k)}, \quad Z^{(k)} = I^{\otimes k} \otimes Z \otimes I^{\otimes m-k-1}$$

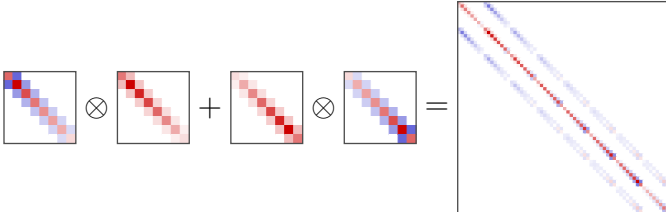


2D meshes

For **tensorial** meshes,

$$K^{2D} = K_x \otimes M_y + M_x \otimes K_y$$

$$M^{2D} = M_x \otimes M_y$$

$$K^{(2D)} = \begin{matrix} \begin{matrix} \text{[Matrix 1]} \end{matrix} \otimes \begin{matrix} \text{[Matrix 2]} \end{matrix} + \begin{matrix} \text{[Matrix 3]} \end{matrix} \otimes \begin{matrix} \text{[Matrix 4]} \end{matrix} = \begin{matrix} \text{[Result Matrix]} \end{matrix}$$


2D meshes

For **tensorial** meshes,

$$K^{2D} = K_x \otimes M_y + M_x \otimes K_y$$

$$M^{2D} = M_x \otimes M_y$$

$$M^{(2D)} = \begin{array}{|c|} \hline \text{[1D Mass Matrix]} \\ \hline \end{array} \otimes \begin{array}{|c|} \hline \text{[1D Mass Matrix]} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{[2D Mass Matrix]} \\ \hline \end{array}$$

At the end of the day, the block-encoding has

$$\left. \begin{array}{l} d = O(\log N) \\ \lambda/\|A\| = O(1) \end{array} \right\} \Rightarrow O\left(\log N \kappa \log \frac{1}{\varepsilon}\right)^{(*)}$$

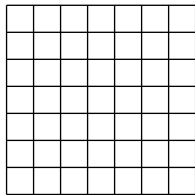
(*) with higher constants than the uniform case

Numerical results

$$\nabla^2 u(x, y) = \begin{cases} 1, & 0 \leq y \leq 1/4 \\ 0, & 1/4 < y \leq 1 \end{cases}$$

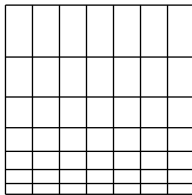
$$u(x, y) = 0, \quad x, y \in \partial\Omega$$

uniform



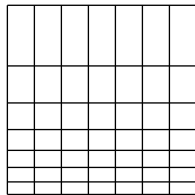
$$h_j = h_0$$

geometric



$$h_j = h_0 r^j$$

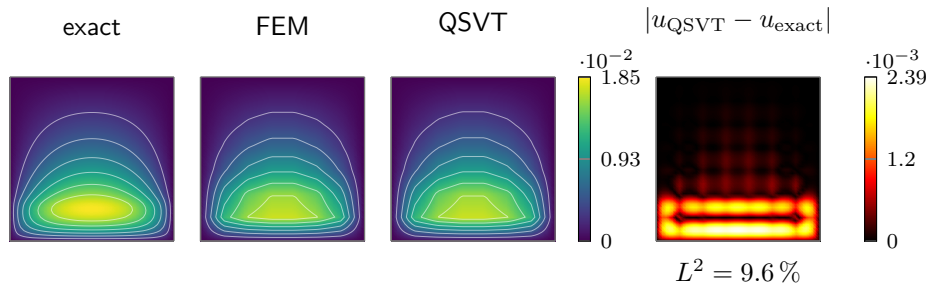
Inverse



$$h_j = 1/(aj + b)$$

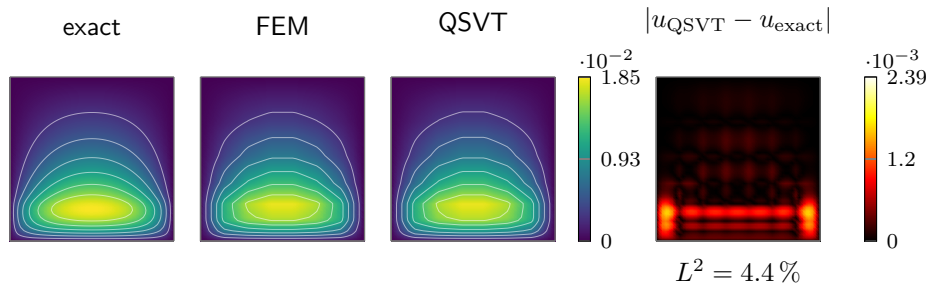
Numerical result

Uniform mesh



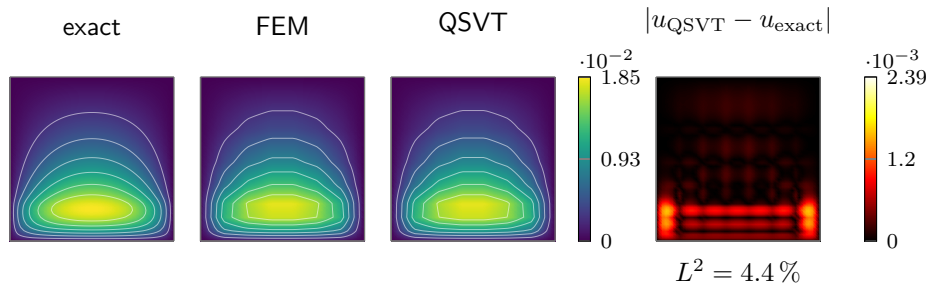
Numerical result

Geometric mesh



Numerical result

Inverse mesh



Concluding remarks

Btw, this generalizes to

- ▶ high-order FE
- ▶ triangle elements

Some future works :

- ▶ Other differential operators;
- ▶ Space dependent material parameters;
- ▶ Non-tensorial meshes.