SIGGRAPH 2021

MULTISCALE CHOLESKY PRECONDITIONING FOR ILL-CONDITIONED SYSTEMS

JIONG CHEN, ZJU/INRIA
FLORIAN SCHÄFER, CALTECH
JIN HUANG, ZJU
MATHIEU DESBRUN, INRIA/X/CALTECH

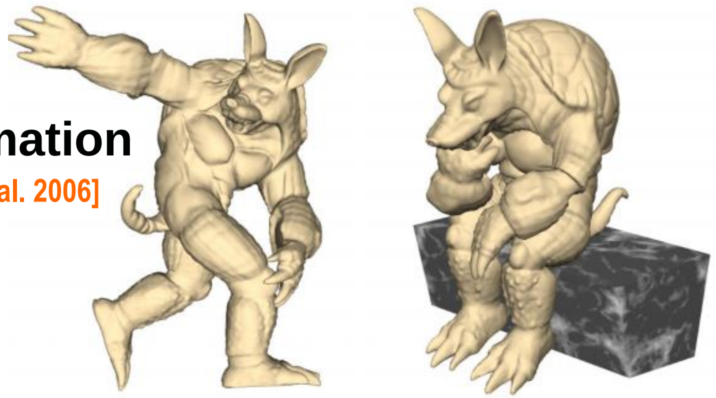
BACKGROUND

SIGGRAPH 2021

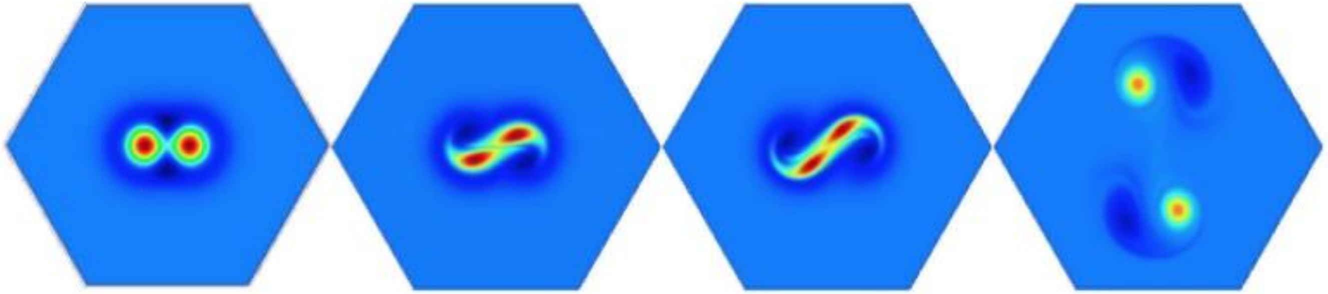
- Many applications boil down to solving linear equations

deformation

[Huang et al. 2006]

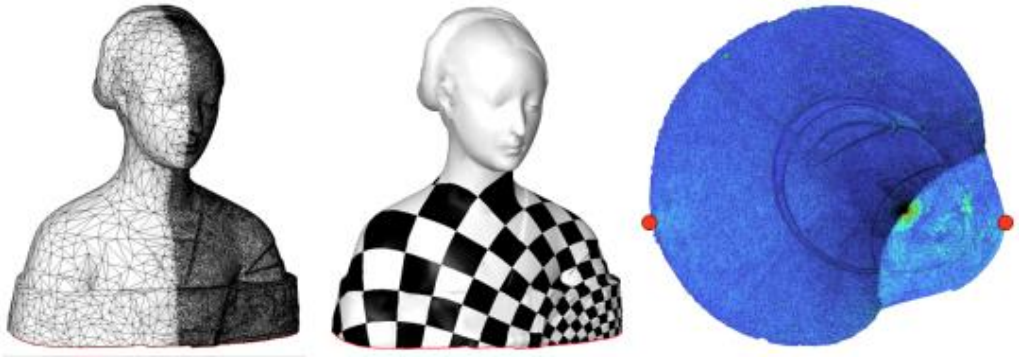


simulation



[Mullen et al. 2009]

parameterization



[Mullen et al. 2008]

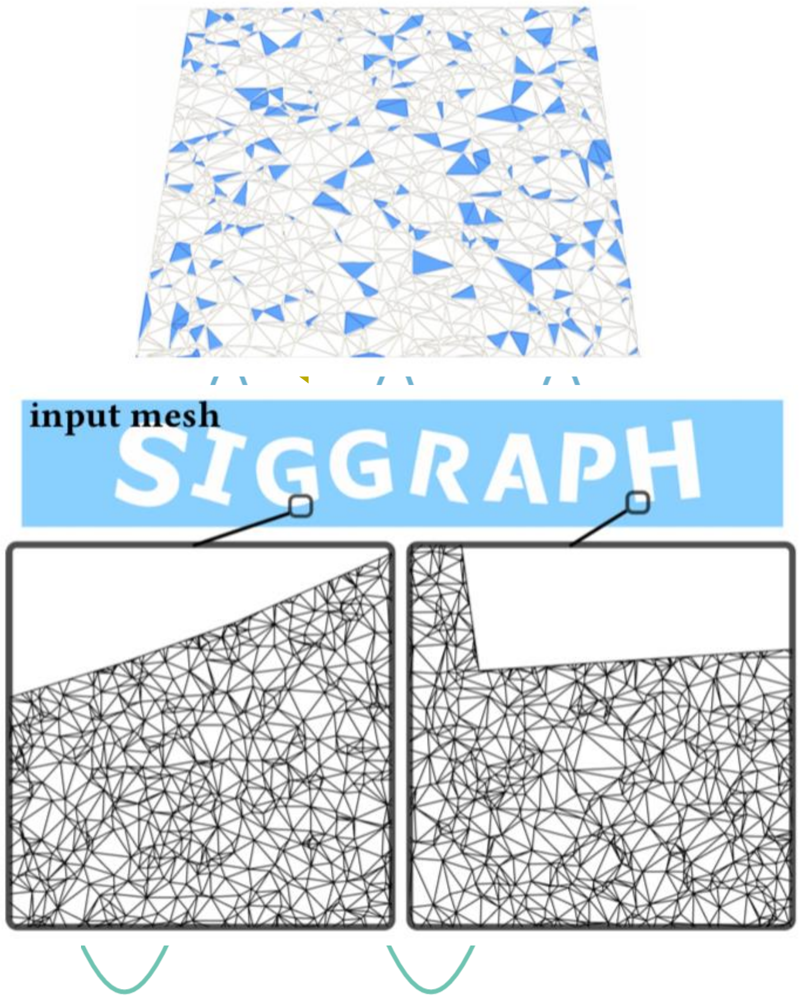
image processing

[Farbman et al. 2008]

PROBLEM & SOLUTION

SIGGRAPH 2021

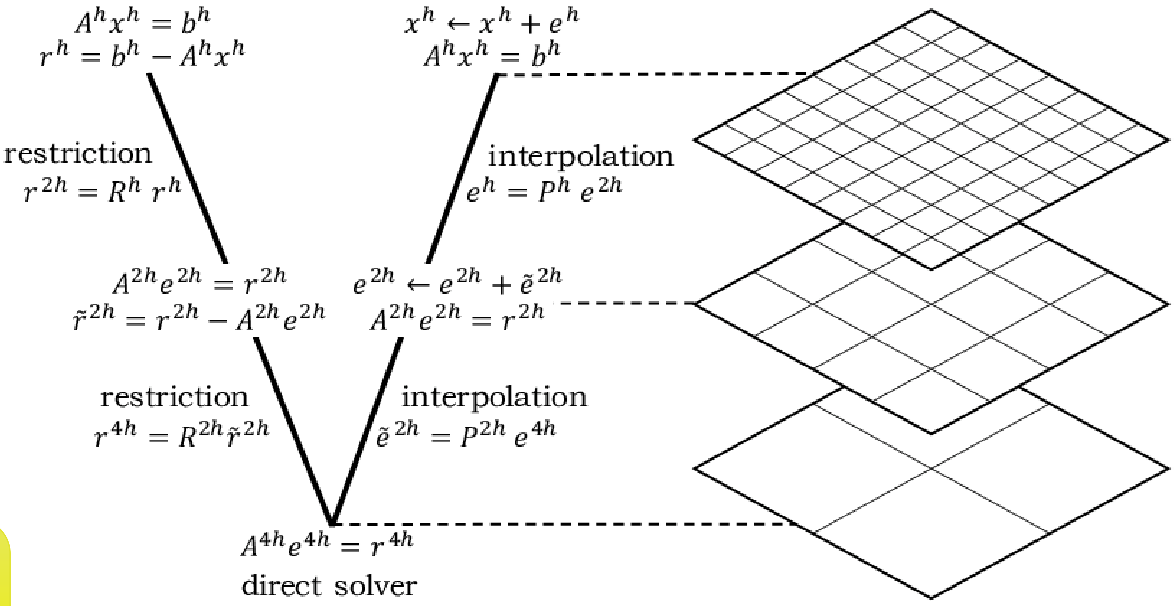
- Ill-conditioned problems
  - source: irregular sampling, bad elements, heterogeneous materials
  - result: signals containing very different scales
- Basic methodology: **multiscale decomposition**
  - **decompose** the whole problem into multiple scales
  - treat **each scale efficiently** to avoid *wasteful* & *redundant* computations
- How it works:
  - treat scales back and forth
  - to decrease the error of different frequencies



EXAMPLE I: MULTIGRID APPROACH

- Coarsening and scale communications via
  - **sparse** restriction and prolongation
- Geometric & algebraic variant
  - building $\boldsymbol{P}$: mesh interpolation or graph simplification
- Efficient for particular matrices

Cheap per-iteration cost,
but slow convergence.



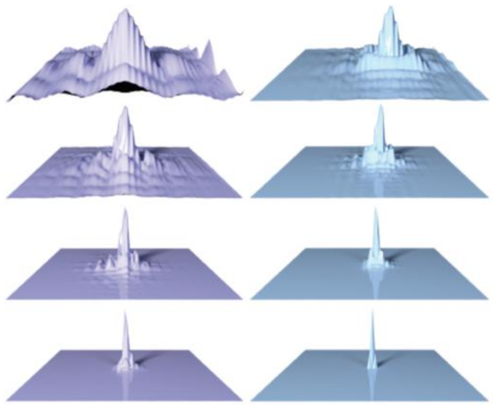
EXAMPLE II: MULTILEVEL APPROACH

- Operator-adapted wavelets [Owhadi 2017; Budninskiy 2019; Chen 2019]

Multilevel decomposition

$$\mathbf{A}\mathbf{u} = \mathbf{g}$$

$$\begin{pmatrix} \mathbb{A}^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbb{B}^1 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbb{B}^{q-1} \end{pmatrix}$$



Re-expressed with hierarchical, spatially localized, eigenfunction-like basis fcts and wavelets

Multilevel solve

$$\mathbf{A}^q u^q = g^q$$

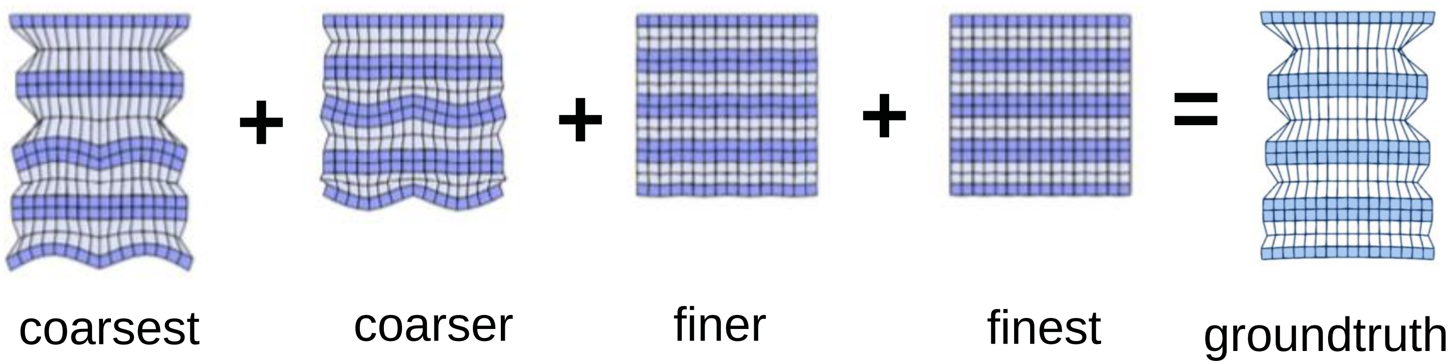
Independent solves across levels

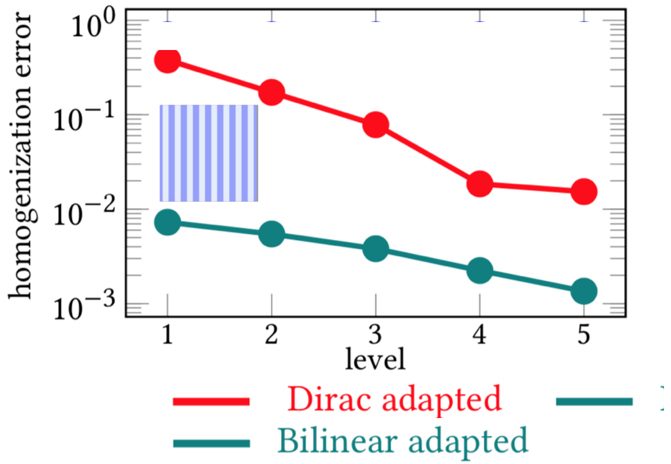
$$\mathbb{B}^k \mathbb{w}^k = \mathbf{W}^k \mathbb{g}^{k+1} \text{ for } q-1 \geq k \geq 1$$
$$\mathbb{A}^1 \mathbb{v}^1 = \mathbb{g}^1$$

Assemble all-level solutions

$$u^q = \Phi^{1,T} \mathbb{v}^1 + \sum_{k=1}^{q-1} \Psi^{k,T} \mathbb{w}^k$$

EXAMPLE II: MULTILEVEL APPROACH





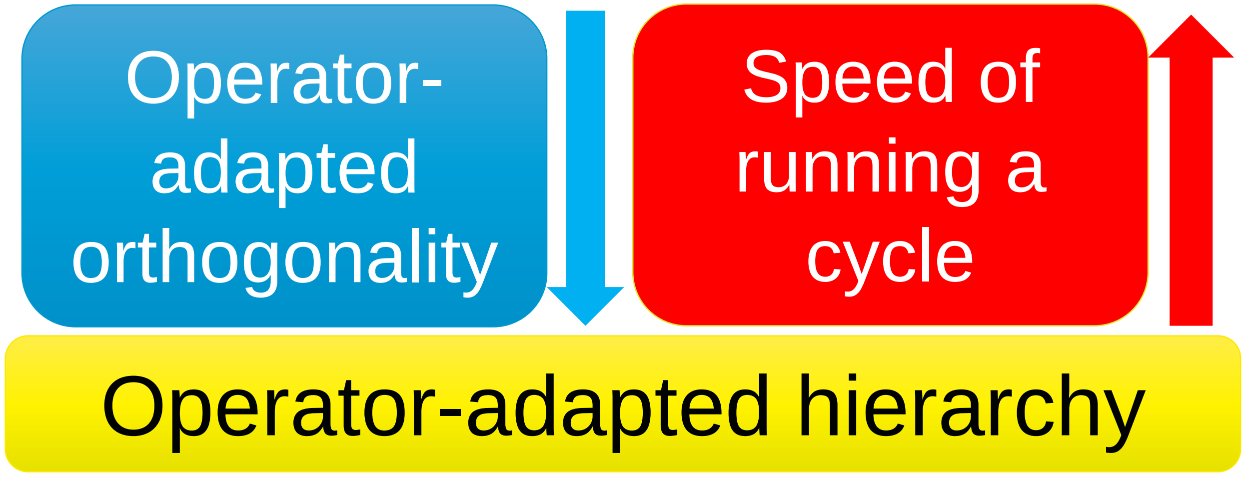
Great convergence, but very high computational cost.

EXAMPLE II: MULTILEVEL APPROACH

- Closed-form solution of restriction operator [Chen 2019]

$$\mathbb{C}^k = \mathbf{C}^{k,\dagger}\left[\mathbb{I}_{3n_{k+1}\times 3n_{k+1}} - \mathbb{A}^{k+1}\,\mathbf{W}^{k,T}\left(\mathbb{B}^k\right)^{-1}\mathbf{W}^k\right],$$

  - this operator will be denser on coarse levels
  - slow down matrix factorization & multiplication



- Our idea
  - for inhomogeneous systems, the **OA wavelet approach** is great but expensive
  - so we need to trade **convergence** for an **acceleration of each iteration** to obtain a fast solver

Need a brand new computational tool!

CONTRIBUTIONS

- Re-express [Chen 2019] construction by **Cholesky decomposition** based on [Schäfer et al. 2021]
- Leverage incomplete factorization for preconditioning inhomogeneous systems
  - exploit **multiscale ordering** and **sparsity pattern** to make tradeoffs
- Introduce efficient implementation
  - parallelization, supernodes...
- Outcome
  - **a multiscale preconditioner for large scale heterogeneous linear systems.**
  - **filling a gap in the arsenal of linear solvers**

SIGGRAPH 2021

OUR METHOD

INPUT AND PIPELINE

- Input of our algorithm
  - mesh for problem discretization
  - SPD matrix discretizing a differential operator
- Pipeline of our method

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{z}_0 := \mathbf{M}^{-1}\mathbf{r}_0$
$\mathbf{p}_0 := \mathbf{z}_0$
$k := 0$
repeat
  $\alpha_k := \frac{\mathbf{r}_k^\top \mathbf{z}_k}{\mathbf{p}_k^\top \mathbf{A}\mathbf{p}_k}$
  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$
  $\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$
  if $\mathbf{r}_{k+1}$ is sufficiently small **then** exit loop **end if**
  $\mathbf{z}_{k+1} := \mathbf{M}^{-1}\mathbf{r}_{k+1}$
  $\beta_k := \frac{\mathbf{r}_{k+1}^\top \mathbf{z}_{k+1}}{\mathbf{r}_k^\top \mathbf{z}_k}$
  $\mathbf{p}_{k+1} := \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k$
  $k := k + 1$
end repeat

1. Fine-to-coarse reordering

2. Construct multiscale sparsity pattern

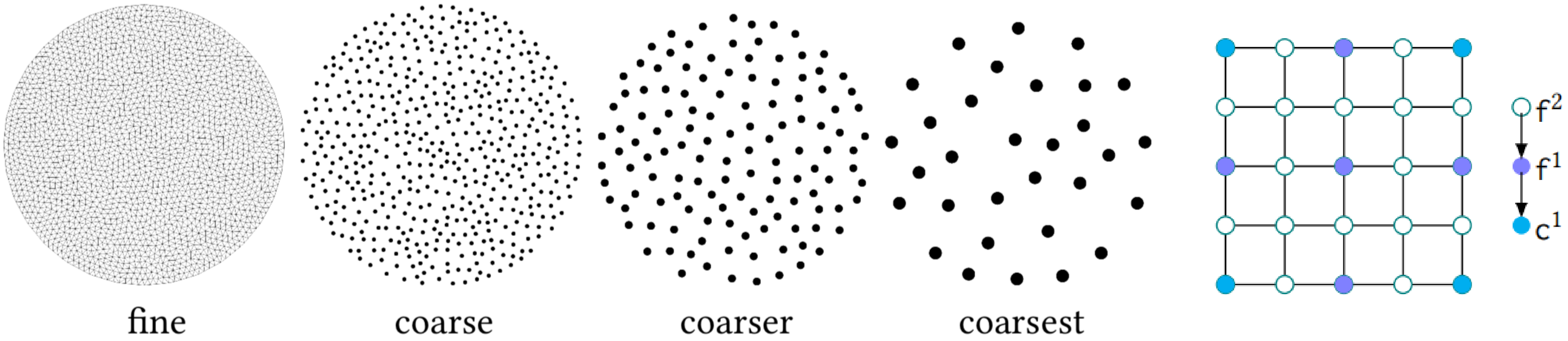
3. Perform incomplete Cholesky factorization

STEP 1: FINE-TO-COARSE REORDERING

- Max-min ordering [Guinness 2018, Schäfer 2021]

$$i_k := \underset{i \in C^q \setminus \{i_1,\ldots,i_{k-1}\}}{\arg\max} \; \min_{j \in \{i_1,\ldots,i_{k-1}\}} \operatorname{dist}(x_i, x_j),$$

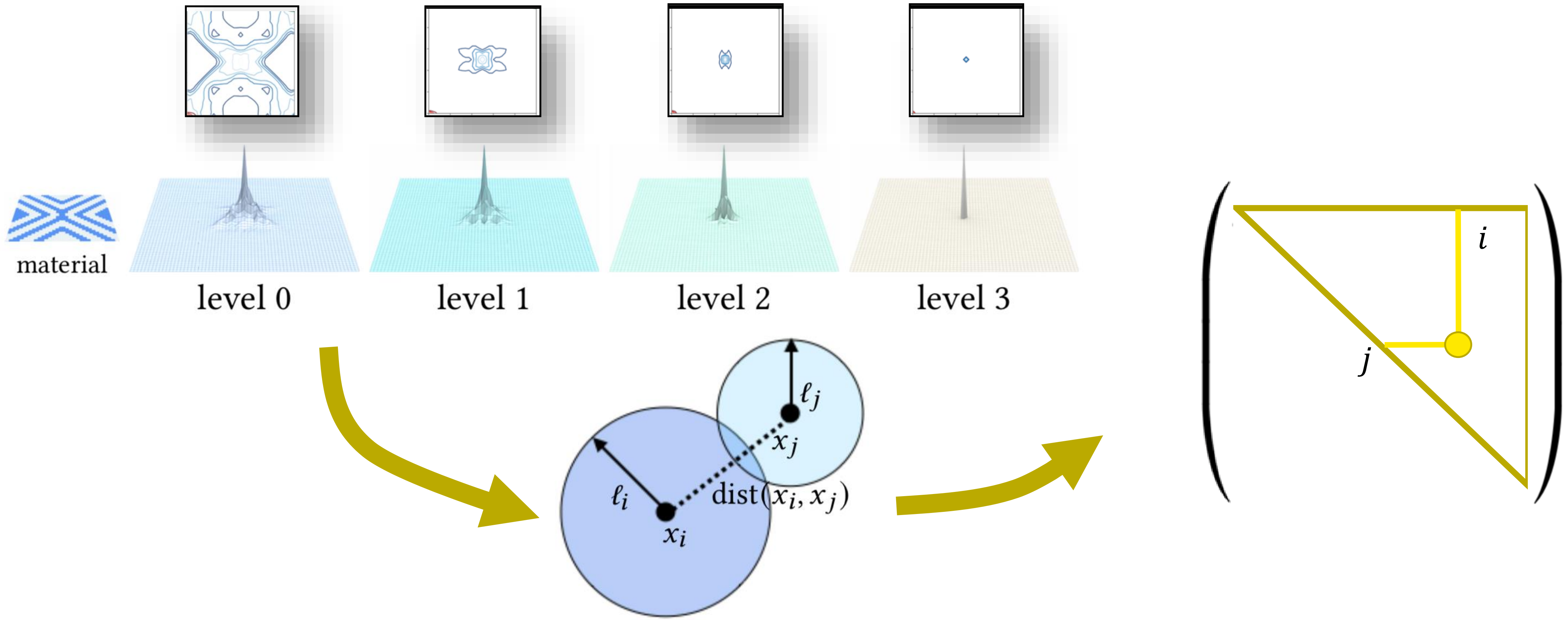
- Reverse to generate fine-to-coarse ordering
  - put the most important DoFs last



STEP 2: MULTISCALE SPARSITY PATTERN

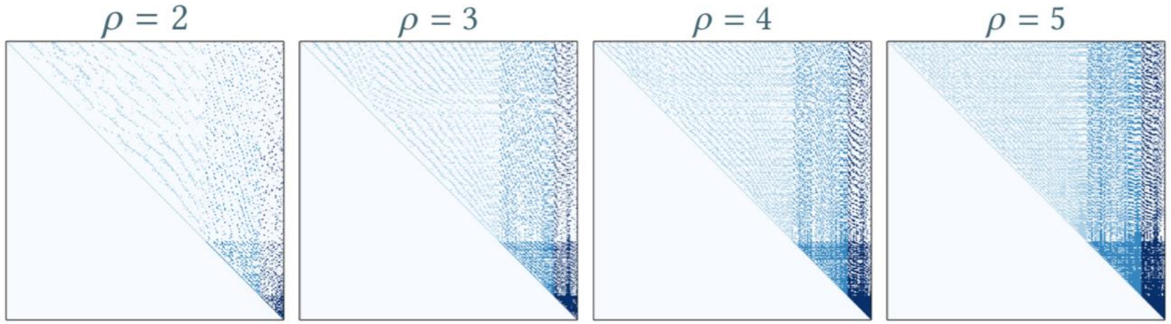
- Sparsity

$$S_\rho := \{(i, j) \in C^q \times C^q \mid \mathrm{dist}(x_i, x_j) \le \rho \min(\ell_i, \ell_j)\},$$

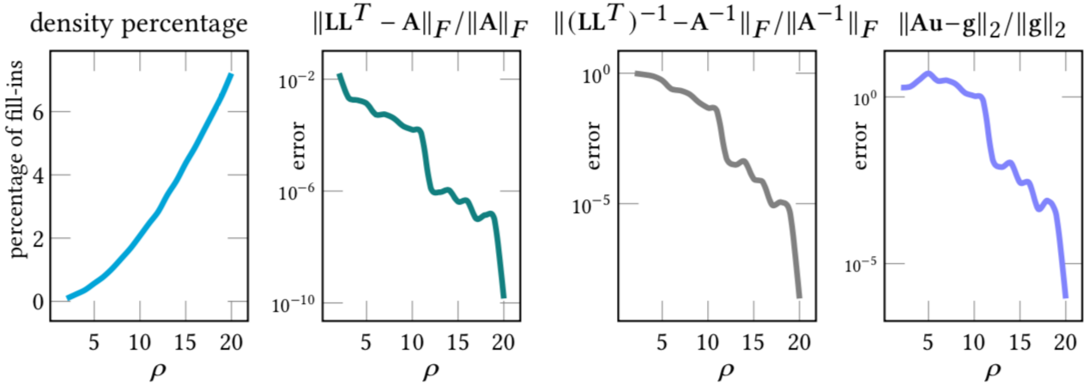


DISCUSSION

- **Single scale *vs.* multiscale sparsity**
  - with comparable nnz, multiscale sparsity pattern is more accurate

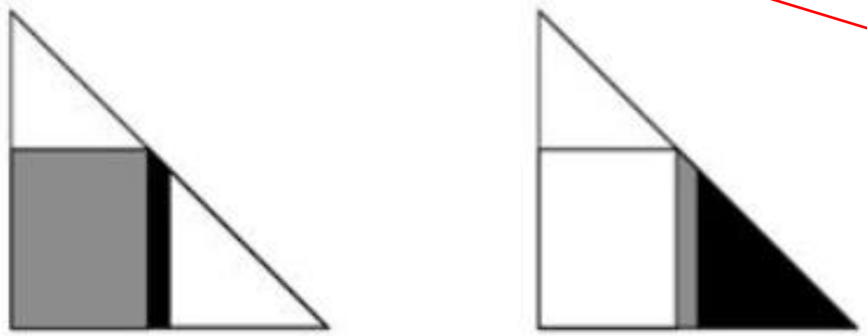


- **Small ρ *vs.* large ρ**
  - the error decays rapidly as ρ increases
    - complexity of $O(\log n \exp(-C\rho))$
  - convergence *vs*. update speed



STEP 3: ZERO FILL-IN INCOMPLETE FACTORIZATION

- Ignore entries outside of the given sparsity pattern



Left–looking Right–looking

Input: SPD matrix $\mathbf{A}$ and sparsity pattern S_ρ.

Output: Incomplete Cholesky factor $\mathbf{L}$ such that $\mathbf{A} \approx \mathbf{L}\mathbf{L}^T$

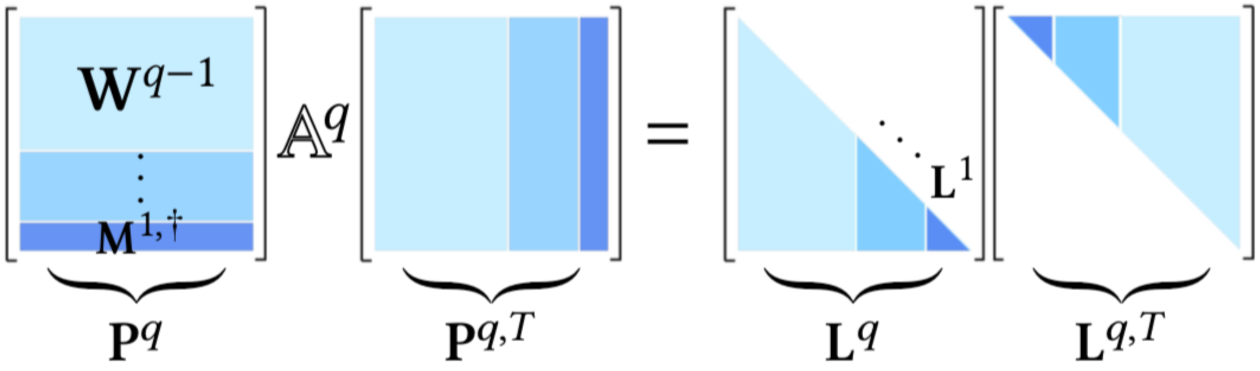
place non-zeros of $\mathbf{A}$ into $\mathbf{L}$ according to S_ρ;
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow i$ **to** n **do**
        for $k \leftarrow 1$ **to** $i-1$ **do**
            if $(j,i),(j,k),(i,k) \in S_\rho$ **then**
                $\mathbf{L}(j,i) \leftarrow \mathbf{L}(j,i) - \mathbf{L}(j,k)\mathbf{L}(i,k)$;
            end
        end
    end
    $\mathbf{L}(i,i) \leftarrow \sqrt{\mathbf{L}(i,i)}$;
    for $j \leftarrow i+1$ **to** n **do**
        $\mathbf{L}(j,i) \leftarrow \mathbf{L}(j,i)/\mathbf{L}(i,i)$;
    end
end

Algorithm 1: Left-looking incomplete Cholesky factorization.

Operator-adapted wavelets ≡ Schur complement ≡ Cholesky

(once expressed in the right basis)

[Schäfer et al. 2021]



Material adapted-operators: $\mathbb{A}^k = \widehat{\mathrm{L}}_{\mathsf{c}}\,\widehat{\mathrm{L}}_{\mathsf{c}}^T, \quad \mathbb{B}^k = \widehat{\mathrm{L}}_{\mathsf{f}}\,\widehat{\mathrm{L}}_{\mathsf{f}}^T.$

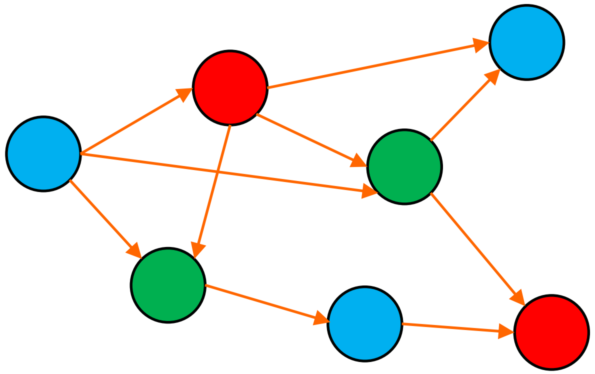
SIGGRAPH 2021

IMPLEMENTATION

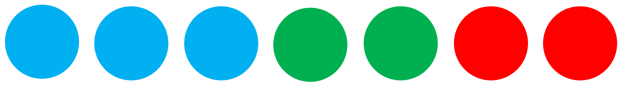
MULTITHREADING

- Level scheduling
  - eliminate independent columns in parallel
  - topological ordering based on DAG

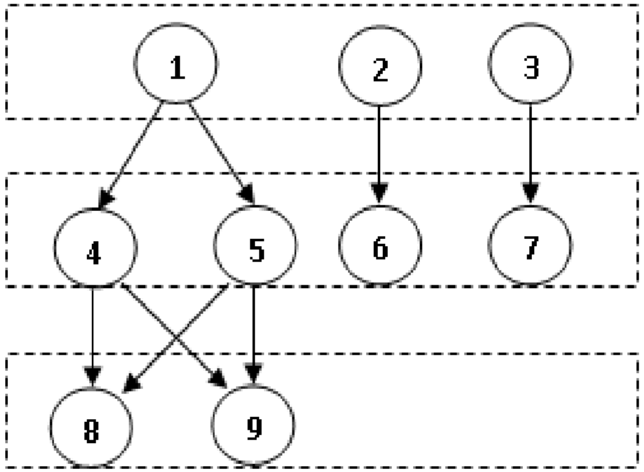
- Multicoloring



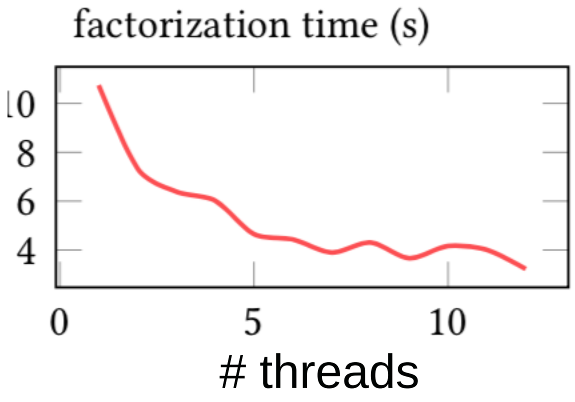
Multicolor reordering



$$\begin{pmatrix} a_{11} & * & * & * & * & * & * & * & * \\ & a_{22} & * & * & * & * & * & * & * \\ & & a_{33} & * & * & * & * & * & * \\ a_{41} & & & a_{44} & * & * & * & * & * \\ a_{51} & & & & a_{55} & * & * & * & * \\ & a_{62} & & & & a_{66} & * & * & * \\ & & a_{73} & & & & a_{77} & * & * \\ & & & a_{84} & a_{85} & & & a_{88} & * \\ & & & a_{94} & a_{95} & & & & a_{99} \end{pmatrix}$$



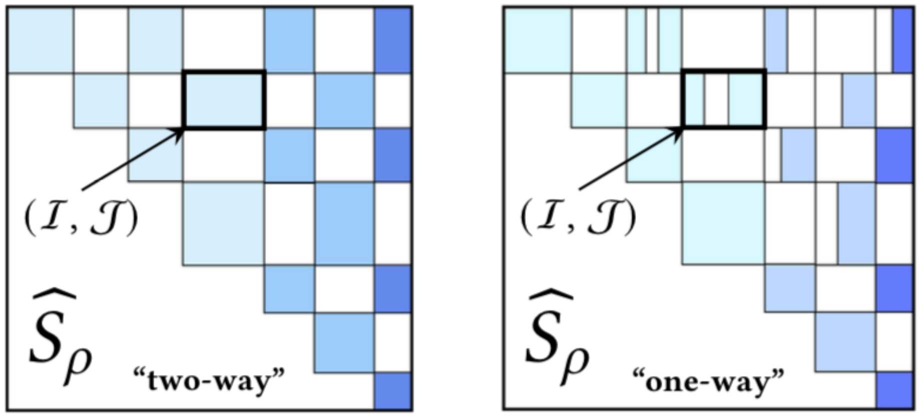
[Naumov 2012]



SUPERNODES

- Supernodal approach: a block version of factorization
  - **scalar product → matrix product**
  - **square root → dense Cholesky**
  - **division → triangular solves**
- Introducing extra fill-ins : two-way *vs.* one-way supernodes

$$\widehat{S}_\rho := \{(\mathcal{I},\mathcal{J}) | \exists i \in \mathcal{I}, \exists j \in \mathcal{J} \text{ and } (i,j) \in S_\rho\}$$



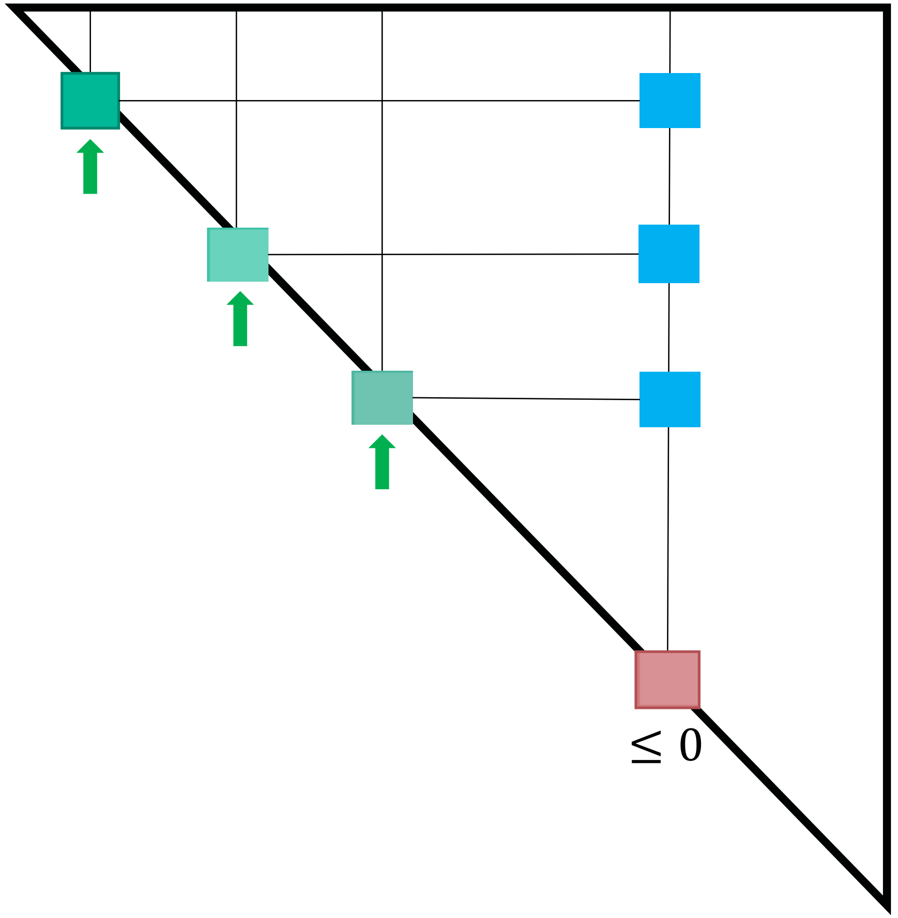
Input: SPD matrix $\mathbf{A}$ and sparsity pattern S_ρ.
Output: Incomplete Cholesky factor $\mathbf{L}$ such that $\mathbf{A} \approx \mathbf{L}\mathbf{L}^T$

place non-zeros of $\mathbf{A}$ into $\mathbf{L}$ according to S_ρ;
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow i$ **to** n **do**
        for $k \leftarrow 1$ **to** $i-1$ **do**
            if $(j,i),(j,k),(i,k) \in S_\rho$ **then**
                $\mathbf{L}(j,i) \leftarrow \mathbf{L}(j,i) - \mathbf{L}(j,k)\mathbf{L}(i,k);$
            end
        end
    end
    $\mathbf{L}(i,i) \leftarrow \sqrt{\mathbf{L}(i,i)};$
    for $j \leftarrow i+1$ **to** n **do**
        $\mathbf{L}(j,i) \leftarrow \mathbf{L}(j,i)/\mathbf{L}(i,i);$
    end
end

Algorithm 1: Left-looking incomplete Cholesky factorization.

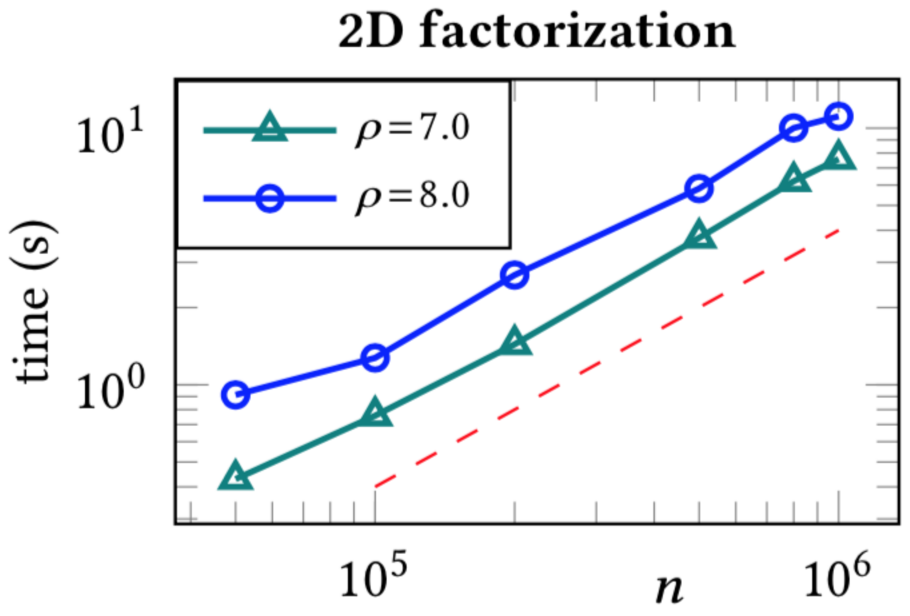
BREAKDOWNS FIXING

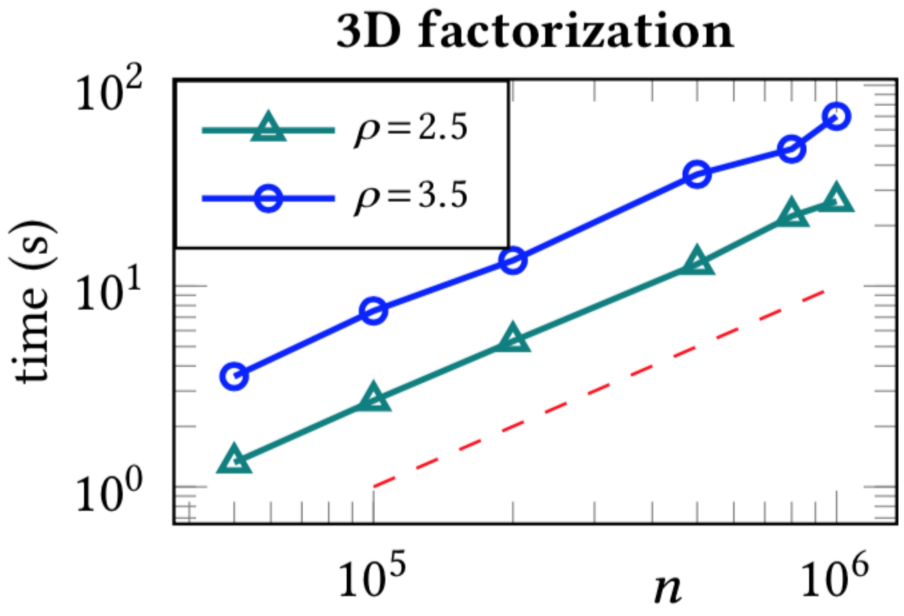
- Global fixing strategy **[Scott and Tuma 2014]**
  - M-matrix will always success
  - $A \leftarrow A + \alpha\, \mathrm{Id}$ (trial and error)
  - slowed-down convergence
- Partial fixing strategy
  - negative pivot appears because of error accumulation along elimination path
  - change only relevant diagonal elements



SCALABILITY OF FACTORIZATION

- Theoretical complexity of numerical factorization $\boldsymbol{O}(\boldsymbol{n}\,\boldsymbol{\rho}^{2d})$ [Schäfer et al. 2021]





SIGGRAPH 2021

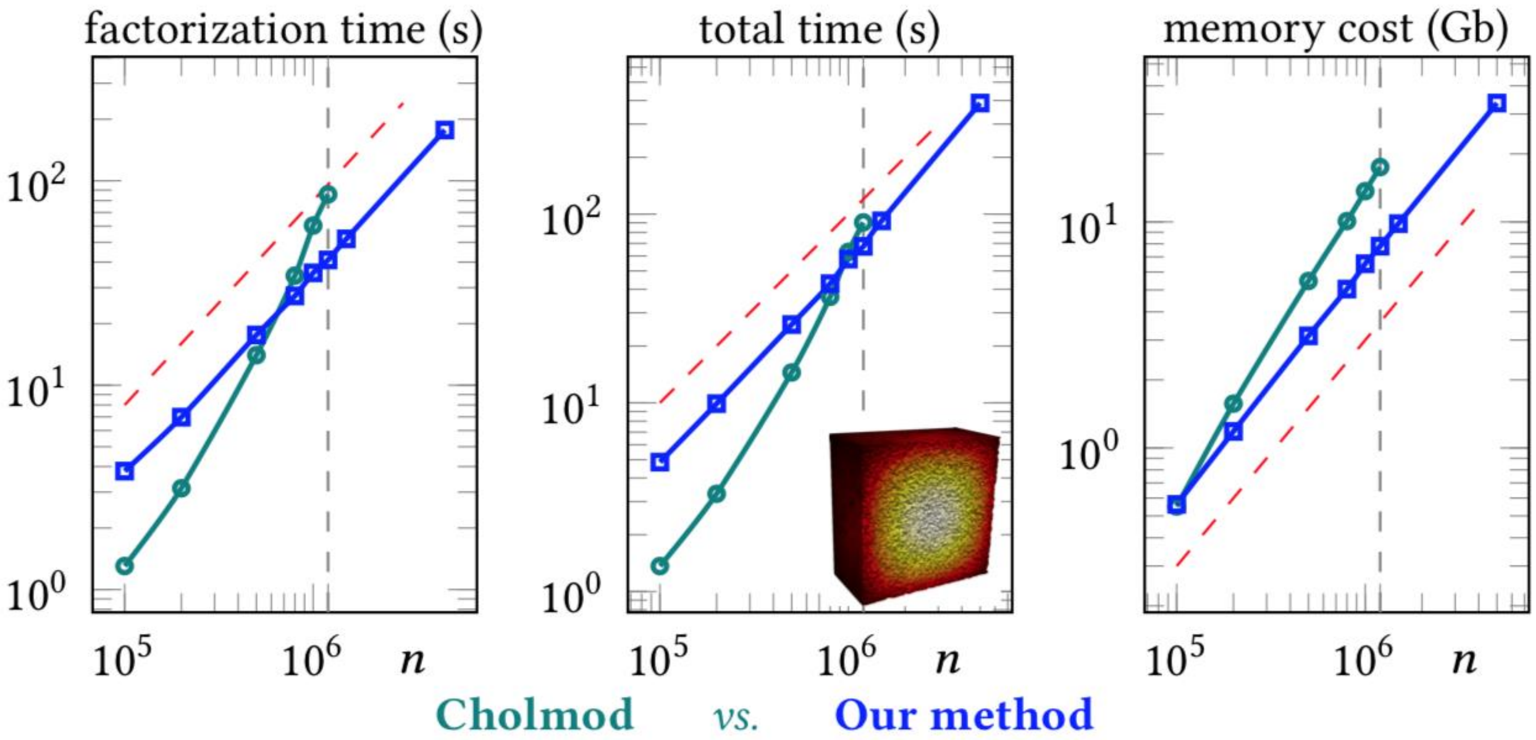
RESULTS

COMPARISONS WITH CHOLMOD

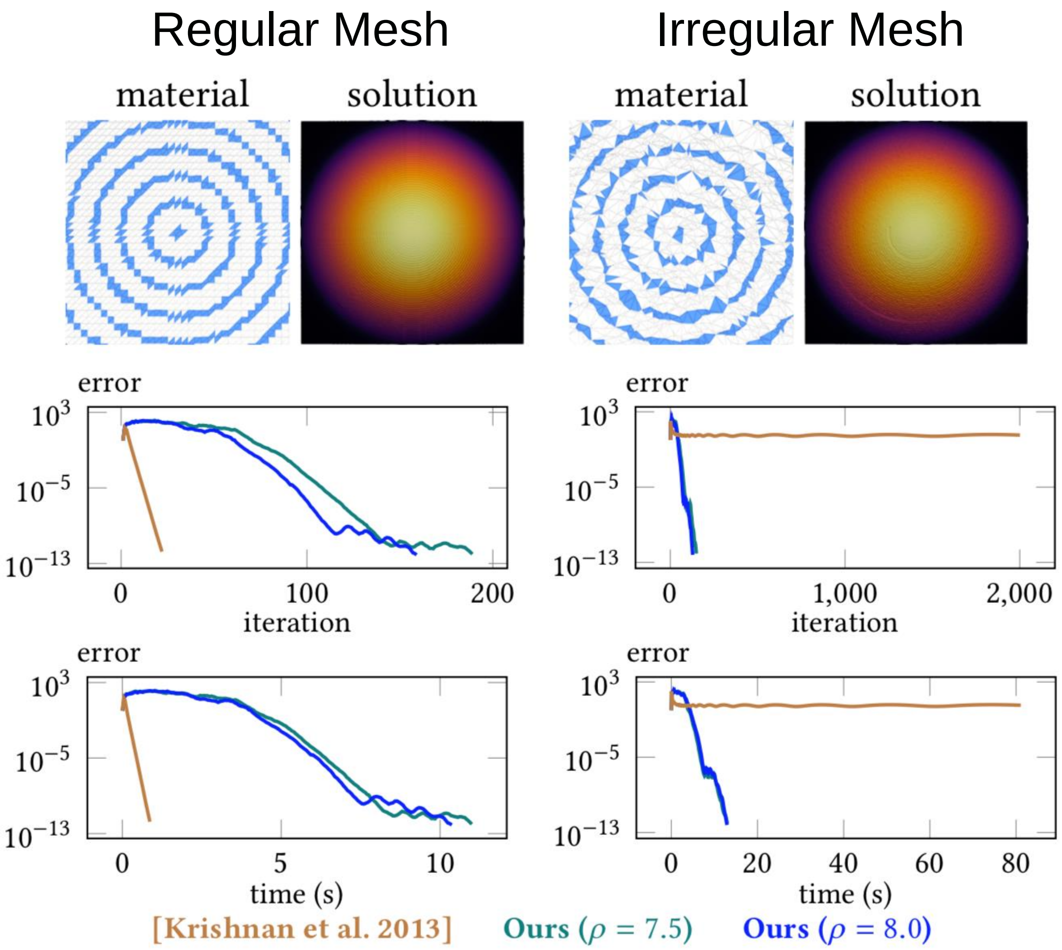
- Sparse complete Cholesky

	time	memory
2D	$O(n^{3/2})$	$O(n \log n)$
3D	$O(n^2)$	$O(n^{4/3})$

- Our incomplete Cholesky
  - **time:** $\boldsymbol{O\left(n\,\rho^{2d}\right)}$
  - **storage:** $\boldsymbol{O\left(n\,\rho^{d}\right)}$



COMPARISONS WITH [KRISHNAN 2013]



Multigrid preconditioner

- For matrices with non-positive off diagonal entries
  - $\kappa(P^{-1}A) \leq 3$
  - efficient for Laplacian matrices on regular grids
- For other matrices
  - drops positive off-diagonal entries
  - fails to converge

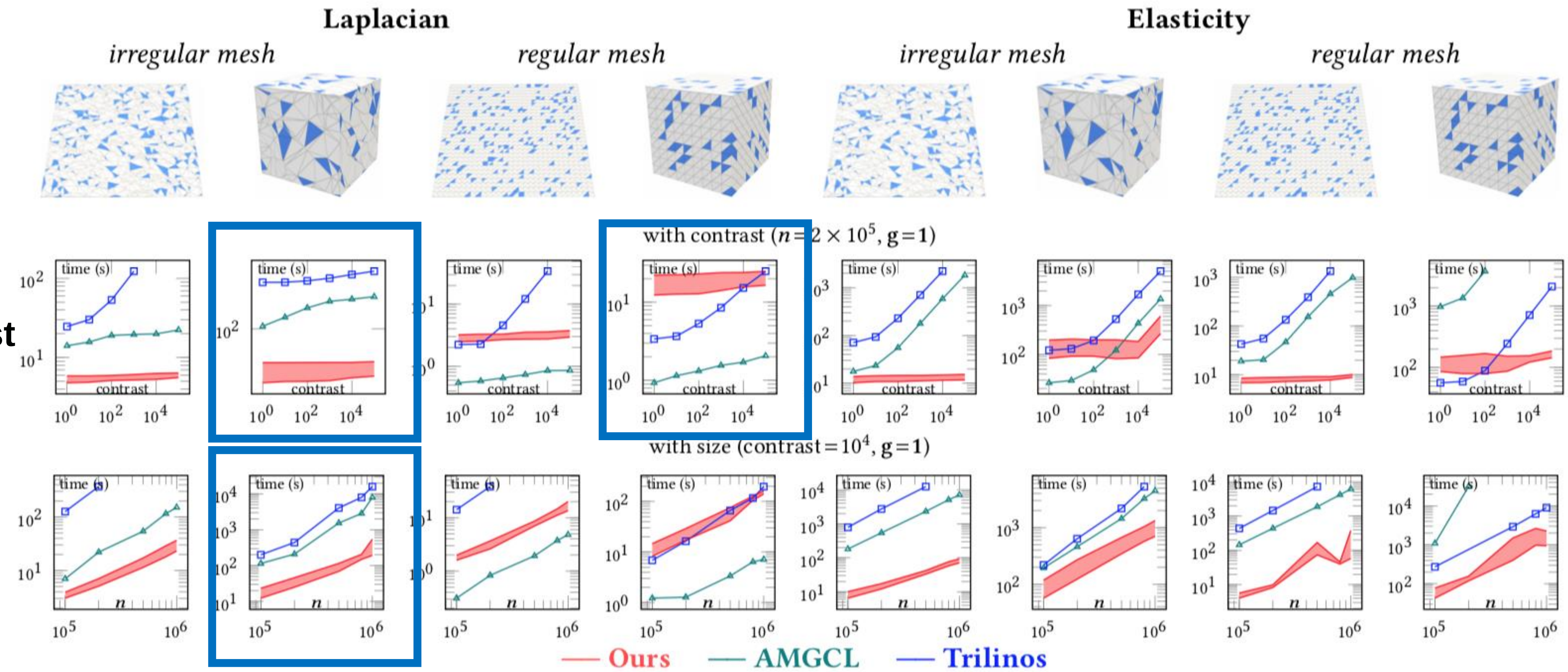
COMPARISONS WITH AMG LIBRARIES

- Settings
  - Trilinos [Trilinos 2020]
    - smoothed aggregation + symmetric Gauss-Seidel for relaxation
  - AMGCL [Demidov 2019]
    - Ruge-Stuben + sparse inverse approximation for relaxation
  - # Pre- and post-relaxation = 1
    - more relaxations do not pay off
  - Tolerance
    - we ask for $\frac{||Ax-b||}{||b||} < 10^{-12}$

COMPARISONS WITH AMG LIBRARIES

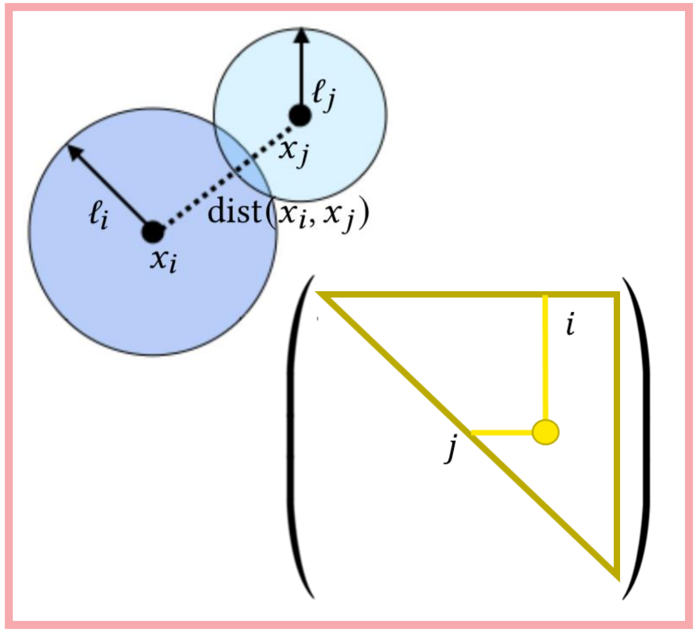
contrast

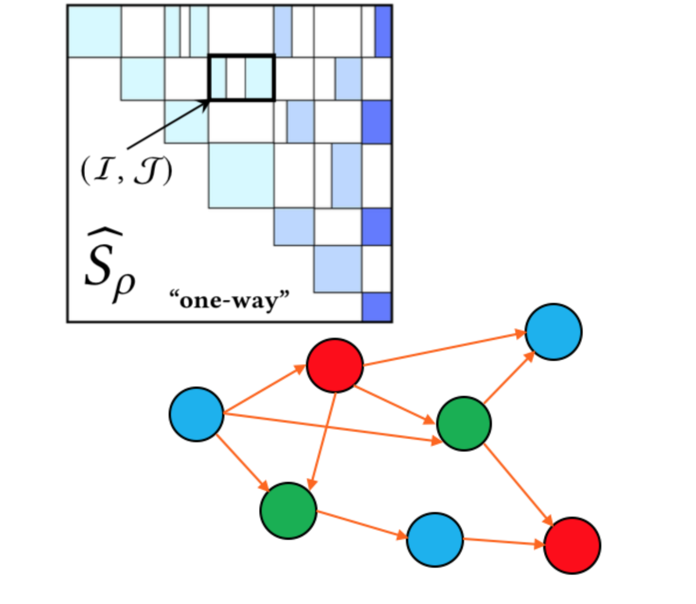
size

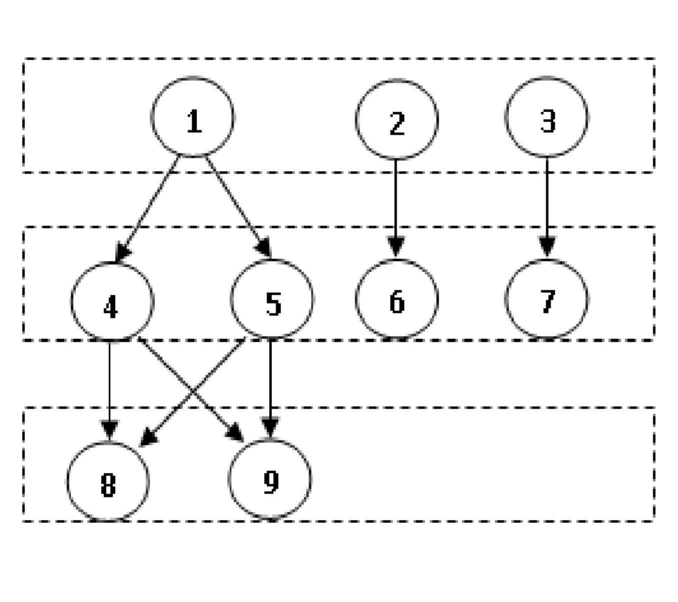


PRECOMPUTATION

- Precomputations involve
  - construction of sparsity pattern
  - aggregation and coloring of the supernodes
  - level scheduling

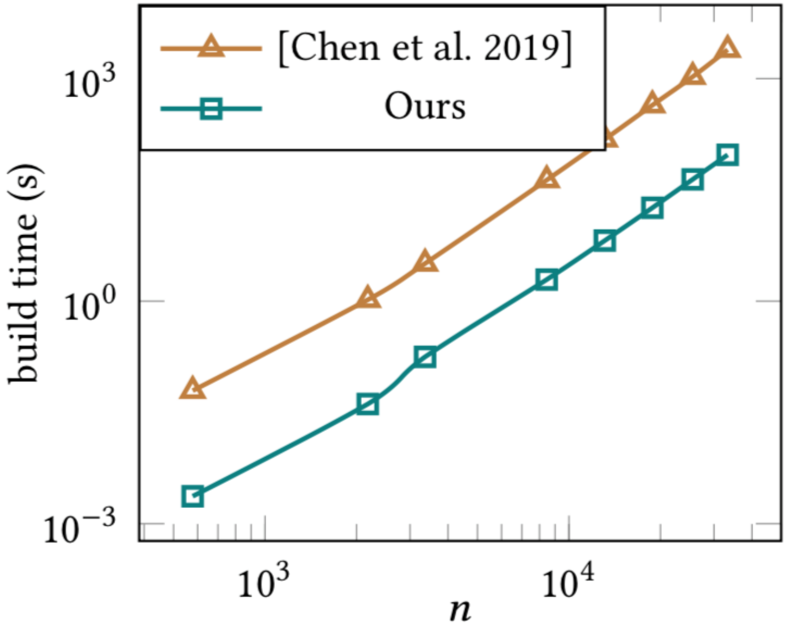


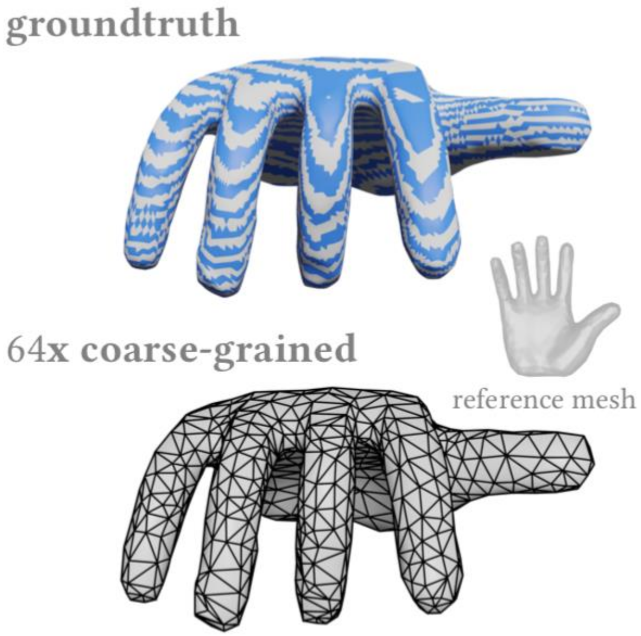




SIGGRAPH 2021

- Fast construction of material-adapted basis functions and wavelets

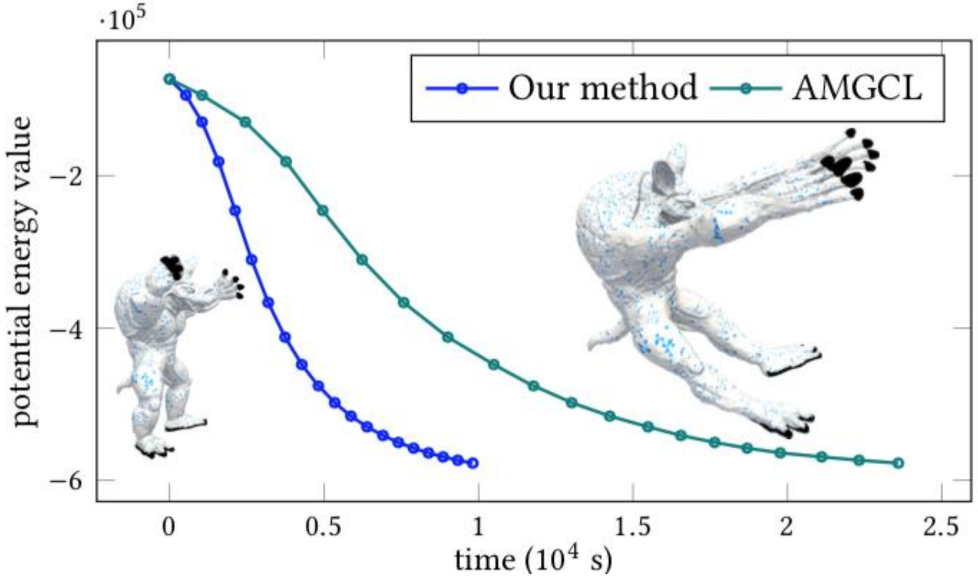




Useful for modal reduction

APPLICATIONS

SIGGRAPH 2021



Nonlinear elastostatics

Nonlinear elastic dynamics

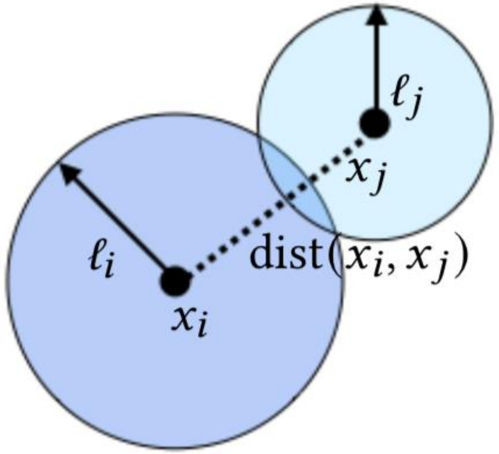
Edge-preserving multiscale decomposition

LIMITATIONS AND FUTURE WORKS

- Problem-adapted sparsity pattern
  - e.g., specify l_i according to Green's function
- High-order wavelet transformation

ρ	#iter (bilinear)	#iter (lazy)
2.0	184	433
2.5	135	278
3.0	85	134
3.5	80	117
4.0	67	72

- Breakdowns
  - avoid all breakdowns?
- Non-symmetric problems



SIGGRAPH 2021

THANKS!