Exceptional service in the national interest

Scalable methods for nonlocal models

Christian Glusa

Center for Computing Research, Sandia National Laboratories

April 16, 2024

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO. SAND2024-04215C

Motivation

- Fractional and nonlocal models allow for a more accurate description of phenomena in a wide range of applications [13, 8]:
  - anomalous diffusion, $\Delta t \sim \Delta x^{2s}$, $s \neq 1$ [6],
  - material science, peridynamics, low regularity requirements allow fractures [17, 5],
  - image processing [10, 14],
  - finance [18],
  - electromagnetic fluids [15].
- Space-fractional equations arise naturally as the limit of discrete diffusion governed by stochastic processes with long jumps [16].
- My interests in fractional order equations:
  - subsurface transport,
  - subsurface imaging,
  - sea ice dynamics,
  - fast linear solvers

Elliptic nonlocal operators

Let $\delta \in (0, \infty]$ be the *horizon*, $\Omega \subset \mathbb{R}^d$ a bounded open domain, define the *interaction domain*

$$\Omega_I := \{\mathbf{y} \in \mathbb{R}^d \setminus \Omega : |\mathbf{x} - \mathbf{y}| \le \delta, \quad \text{for } \mathbf{x} \in \Omega\}.$$

We want to numerically solve equations involving the nonlocal operator

$$\mathcal{L}u(\mathbf{x}) = \text{p.v.} \int_{\Omega \cup \Omega_I} (u(\mathbf{y}) - u(\mathbf{x}))\gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y}, \qquad \mathbf{x} \in \Omega,$$

with

$$\gamma(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}, \mathbf{y}) \left|\mathbf{x} - \mathbf{y}\right|^{-\beta(\mathbf{x},\mathbf{y})} \mathcal{X}_{|\mathbf{x}-\mathbf{y}| \le \delta}, \qquad \mathbf{x}, \mathbf{y} \in \Omega \cup \Omega_I,$$
$$\phi(\mathbf{x}, \mathbf{y}) > 0.$$

- Examples:
  - Integral fractional Laplacian: $\beta = d + 2s$, $s \in (0, 1)$, $\delta = \infty$, $\phi \sim$ const
  - Tempered fractional Laplacian: $\phi(\mathbf{x}, \mathbf{y}) \sim \exp(-\lambda|\mathbf{x} - \mathbf{y}|)$
  - Truncated fractional Laplacian: δ finite
  - Variable order fractional Laplacians with varying coefficient: $\beta(\mathbf{x}, \mathbf{y}) = d + 2s(\mathbf{x}, \mathbf{y})$ and $\beta(\mathbf{x}, \mathbf{y}) = d + 2s(\mathbf{x})$
  - Integrable kernels: constant kernel ($\beta = 0$), inverse distance kernel ($\beta = 1$)
- Normalized such that
  - $\delta \to 0$ or $s \to 1$ recovers local Laplacian $-\Delta$
  - $s \to 0$ recovers identity

- Nonlocal Poisson equation:

$$\begin{aligned} -\mathcal{L}u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{in } \Omega_I. \end{aligned}$$

- Nonlocal heat equation:

$$\begin{aligned} u_t - \mathcal{L}u &= f \quad \text{in } (0,T)\times\Omega, \\ u &= g \quad \text{in } (0,T)\times\Omega_I, \\ u &= u_0 \quad \text{on } \{0\}\times\Omega. \end{aligned}$$

- Source control:

$$\min_f \frac{1}{2}\|u-u_d\|_{L^2}^2 + \mathcal{R}(f) \quad \text{subject to nonlocal equation}$$

- Parameter learning:

$$\min_{s,\delta,\dots} \frac{1}{2}\|u-u_d\|_{L^2}^2 + \mathcal{R}(s,\delta,\dots) \quad \text{subject to nonlocal equation}$$

Goal

Assemble and solve nonlocal equations in similar complexity & memory as their local counterparts, i.e. $\mathcal{O}(n\log^\alpha n)$.

Well-posedness

$$\gamma(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x},\mathbf{y})\,|\mathbf{x}-\mathbf{y}|^{-d+s(\mathbf{x},\mathbf{y})}\,\mathcal{X}_{|\mathbf{x}-\mathbf{y}|\leq\delta}$$

Theorem

Let

$$\alpha(r) := \sup_{|\mathbf{x}-\mathbf{y}|\leq r} |\phi(\mathbf{x},\mathbf{y}) - \phi(\mathbf{y},\mathbf{x})|\,, \qquad \beta(r) := \sup_{|\mathbf{x}-\mathbf{y}|\leq r} |s(\mathbf{x},\mathbf{y}) - s(\mathbf{y},\mathbf{x})|\,.$$

Assume that

$$\int_0^1 dr\,\frac{\alpha(r)^2}{r^{1+2\overline{s}}} < \infty, \qquad \int_0^1 dr\,\frac{(\beta(r)\,|\log r|)^2}{r^{1+2\overline{s}}} < \infty$$

and that f has sufficient regularity. Then the variational formulation of

$$-\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), \qquad u(\mathbf{x}) = g(\mathbf{x})$$

is well-posed.

Interpretation

- s and ϕ need to be symmetric in the limit $\mathbf{x} \to \mathbf{y}$.
- Anti-symmetric part of coefficients has to decay fast enough.
- Particular case $s(\mathbf{x},\mathbf{y}) = s(\mathbf{x})$: need $s(\mathbf{x})$ to be Hölder continuous.

Variational formulation

- We consider

$$a(u,v) = \frac{1}{2}\int_{\Omega} d\mathbf{x} \int_{\Omega} d\mathbf{y}\ \left[(u(\mathbf{x}) - u(\mathbf{y}))(v(\mathbf{x}) - v(\mathbf{y}))\right]\gamma(\mathbf{x},\mathbf{y}) \\ + \int_{\Omega} d\mathbf{x} \int_{\Omega_I} d\mathbf{y}\, u(\mathbf{x})\, v(\mathbf{x})\, \gamma(\mathbf{x},\mathbf{y}).$$

  posed

$$V := \left\{u \in L^2\left(\mathbb{R}^d\right) \mid \|u\|_V < \infty\right\}, \qquad V_0 := \{u \in V \mid u = 0 \text{ in } \Omega_I\}$$

  and

$$\|u\|_V^2 = \iint_{(\Omega\cup\Omega_I)^2} (u(\mathbf{x}) - u(\mathbf{y}))^2\, \gamma(\mathbf{x},\mathbf{y}).$$

- Reduces to $L^2(\Omega \cup \Omega_I)$ for integrable kernels and to $H^s(\Omega \cup \Omega_I)$ for constant order fractional kernels.
- For $\delta = \infty$, if $\gamma(\mathbf{x},\mathbf{y}) = \nabla_{\mathbf{y}} \cdot \mathbf{\Gamma}(\mathbf{x},\mathbf{y})$, can reduce integral from $\Omega \times \Omega_I$ to $\Omega \times \partial\Omega$, e.g. $\Gamma(\mathbf{x},\mathbf{y}) \sim \frac{\mathbf{x}-\mathbf{y}}{|\mathbf{x}-\mathbf{y}|^{d+2s}}$ for the fractional kernels.

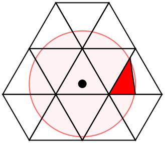
Finite element approximation

- Partition domain Ω into shape-regular mesh $\mathcal{P}_h = \{K\}$ with mesh size h and with edges e on the boundary $\partial\Omega$.
- Set $V_h \subset V$ the space of continuous, piecewise linear functions.

$$a(u,v) = \frac{1}{2}\sum_K \sum_{\tilde{K}} \int_K d\mathbf{x} \int_{\tilde{K}} d\mathbf{y} \left(u\left(\mathbf{x}\right) - u\left(\mathbf{y}\right)\right)\left(v\left(\mathbf{x}\right) - v\left(\mathbf{y}\right)\right)\gamma(\mathbf{x},\mathbf{y})$$
$$+ \sum_K \sum_e \int_K d\mathbf{x}\, u\left(\mathbf{x}\right) v\left(\mathbf{x}\right) \int_e d\mathbf{y}\, \mathbf{n}_e \cdot \Gamma(\mathbf{x},\mathbf{y}).$$

$\dim V_h =: n$

- No closed form for local stiffness matrix → need to use numerical quadrature
- Finite horizon δ or jumps in kernels: approximate with sub-simplices, $\mathcal{O}(h_K^2)$ error[1]



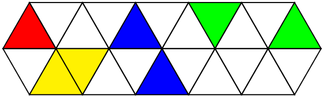
[1] Marta D'Elia, Max Gunzburger, and Christian Vollmann. "A cookbook for approximating Euclidean balls and for quadrature rules in finite element methods for nonlocal problems". In: *Mathematical Models and Methods in Applied Sciences* 31.08 (2021), pp. 1505–1567.

Quadrature

- In subassembly procedure, use quadrature to evaluate element pair contributions:

$$a^{K\times\tilde{K}}(\phi_i,\phi_j)=\frac{1}{2}\int_K d\mathbf{x}\int_{\tilde{K}} d\mathbf{y}\,(\phi_i(\mathbf{x})-\phi_i(\mathbf{y}))\,(\phi_j(\mathbf{x})-\phi_j(\mathbf{y}))\,\gamma(\mathbf{x},\mathbf{y})$$

- Treatment for element pairs $K\cap\tilde{K}\neq\emptyset$, containing the singularity at $\mathbf{x}=\mathbf{y}$:



  - split $K\times\tilde{K}$ into sub-simplices,
  - Duffy transform onto a hypercube, with Jacobian canceling the singularity.
  - Choose quadrature order so that quadrature error $\leq$ discretization error[2]:
    - $|\log h_K|$ if the elements coincide (*red*),
    - $|\log h_K|^2$ if the elements share only an edge (*yellow*),
    - $|\log h_K|^3$ if the elements share only a vertex (*blue*),
    - $|\log h_K|^4$ if the elements are “near neighbours” (*green*), and
    - C if the elements are well separated.

[2] Mark Ainsworth and Christian Glusa. “Aspects of an adaptive finite element method for the fractional Laplacian: A priori and a posteriori error estimates, efficient implementation and multigrid solver”. In: *Computer Methods in Applied Mechanics and Engineering* (2017).

Regularity for the fractional Poisson problem

Theorem (Grubb [11] and Acosta and Borthagaray [1])

Take $\delta = \infty$ and let $\partial\Omega \in C^\infty$, $f \in H^r(\Omega)$ for $r \geq -s$ and $u \in \widetilde{H}^s(\Omega)$ be the solution of the fractional Poisson problem. Then the following regularity estimate holds:

$$u \in H^{s+\vartheta}(\Omega), \qquad \vartheta = \min\{s+r, 1/2-\varepsilon\}\ \forall\varepsilon > 0$$

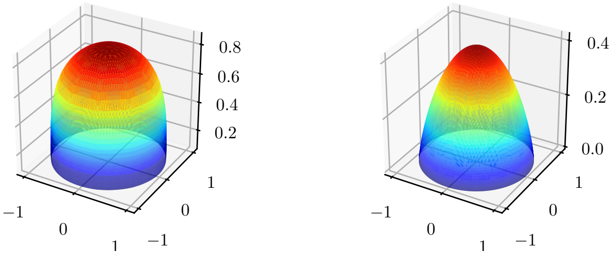


Figure: Solutions $u = C(1-|\boldsymbol{x}|^2)^s \sim \text{dist}\,(\boldsymbol{x}, \partial\Omega)^s$ corresponding to the constant right-hand side $f = 1$ for $s = 0.25$ and for $s = 0.75$, $\delta = \infty$.

Finite element approximation: a priori error estimates

Lemma (Quasi-uniform mesh (Acosta and Borthagaray [1]))

If $u \in H^{s+\vartheta}(\Omega)$, for $s+\vartheta \in (1/2, 2]$ and $0 < \vartheta$, then

$$\|u - u_h\|_{\widetilde{H}^s(\Omega)} \leq Ch^{\vartheta} |u|_{H^{s+\vartheta}(\Omega)}.$$

In practice, $\vartheta \leq 1/2 - \varepsilon$, so $h^{1/2-\varepsilon}$ convergence in $\widetilde{H}^s(\Omega)$ at best.

Lemma (Non-uniform meshes)

If $u \in H^{s+\vartheta}(\Omega) \cap H^{s+\ell}_{loc}(\Omega)$, for $s+\vartheta, s+\ell \in (1/2, 2]$ and $0 < \vartheta \leq \ell$,
i.e. if u has Sobolev regularity $s+\vartheta$ and interior regularity $s+\ell$, then

$$\|u - u_h\|_{\widetilde{H}^s(\Omega)} \leq C\left(h^{\ell} |u|_{H^{s+\ell}_{loc}(\Omega)} + h_{\partial}^{\vartheta} |u|_{H^{s+\vartheta}(\Omega)}\right),$$

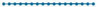
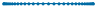
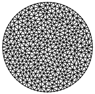
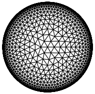
where h_{∂} is the maximum size of all elements K whose patch touches the boundary.

Picking $h_{\partial} \sim h^{\ell/\vartheta}$, the optimal rate of convergence is $h^{2-s} \sim n^{(s-2)/d}$ due to better interior regularity. Unfortunately, shape regularity constrains us to $n^{-1/d}$ in dimensions $d > 1$.

Lower convergence rates than for the PDE Laplacian

FEM convergence rates

(Assuming maximum RHS regularity)

			$\widetilde{H}^s(\Omega)$		$L^2(\Omega)$	
$d=1$		quasi-uniform	$h^{1/2}$	$\frac{1}{n^{1/2}}$	$h^{(1/2+s)\wedge 1}$	$\frac{1}{n^{(1/2+s)\wedge 1}}$
		graded / adaptive		$\frac{1}{n^{2-s}}$		$\frac{1}{n^{2}}$
$d\geq 2$		quasi-uniform	$h^{1/2}$	$\frac{1}{n^{1/(2d)}}$	$h^{(1/2+s)\wedge 1}$	$\frac{1}{n^{(1/2+s)/d)\wedge(1/d)}}$
		graded / adaptive		$\frac{1}{n^{1/d}}$		$\frac{1}{n^{(1+s)/d}}$

After FEM discretization:

$$Au = b, \qquad A \in \mathbb{R}^{n\times n}$$

Depending on δ and h:

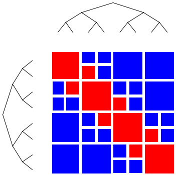
- Straightforward discretization can lead to a fully dense matrix.
- Assembly and solve would have complexity and memory
  - $\mathcal{O}\left(n(\frac{\delta}{h})^d\right)$ for $\delta < \infty$ and
  - $\mathcal{O}(n^2)$ for $\delta = \infty$.

Better approach

Panel clustering / Fast Multipole Method / hierarchical matrix approximation

- Split operator into near and far interactions
- Directly assembly near interactions
- Low-rank approximation of far interactions
- Keep approximation error below discretization error to preserve FE convergence.

Hierarchical matrices: Admissible sub-blocks



Tasks:

1. Choose sub-blocks to be compressed.
2. Construct low-rank approximations.

Build tree of clusters of DoFs.

- root contains all unknowns
- subdivision based on DoF coordinates
- distributed computations: first level given by MPI distribution of unknowns

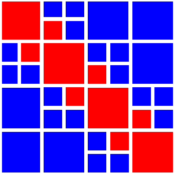
Admissibility criterion:

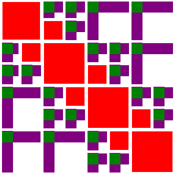
- Cluster pairs (P, Q) that are sufficiently separated compared to their sizes are *admissible* for compression:

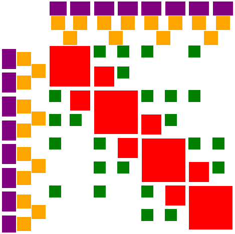
$$\eta \, \mathrm{dist}(P, Q) \leq \max\{\mathrm{diam}(P), \mathrm{diam}(Q)\}, \qquad \eta > 0 \text{ fixed parameter}$$

- Matrix entries that are not admissible are assembled directly into a sparse near-field matrix A_{near}.

Hierarchical matrices: low-rank approximation







- Splitting of operator into sub-blocks based on admissibility

$$A = A_{\text{near}} + A_{\text{far}} = A_{\text{near}} + \sum_{\text{blocks}(P,Q)} A_{P,Q}$$

- $\mathcal{H}$-matrix approximation

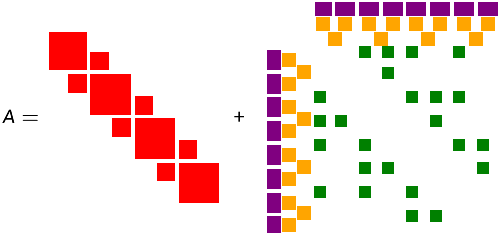
$$A_{P,Q} \approx U_P \Gamma_{P,Q} U_Q^T \qquad \text{(low-rank approximation)}$$

  I use Chebyshev interpolation, but other techniques are possible, e.g. Adaptive Cross Approximation (ACA).

- $\mathcal{H}^2$-matrices
  Using hierarchical nestedness of clusters, can express

$$U_P = \sum_{Q \text{ child of } P} U_Q T_{Q,P}$$

Matrix-vector product with an $\mathcal{H}^2$-matrix



Steps:

- Matvec with sparse near-field matrix
- Upward recursion
- Cluster-cluster interaction
- Downward recursion

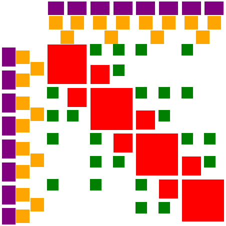
$\mathcal{H}^2$-matrix approximation

Matrix-vector product (and FE assembly) in $\mathcal{O}\left(n \log^{2d} n\right)$ operations & memory.
Approximation error same order as discretization error.

Implementation detail: representation using sparse matrices

Recast hierarchical matrix in terms of sparse matrices

- No special purpose code
- Leverage well-optimized distributed sparse linear algebra



Reindexing of far-field leads to

$$A \approx {\color{red}A_{\text{near}}} + {\color{purple}B}\left[(I+{\color{orange}T_K})\cdots(I+{\color{orange}T_1})\right]{\color{green}\Gamma}\left[(I+{\color{orange}T_1})^T\cdots(I+{\color{orange}T_K})^T\right]{\color{purple}B^T},$$

A_{near} and Γ involve MPI communication, all other matrices are block diagonal

Potential to leverage block structure, but requires support for variable block sizes.
→ I am using point CRS matrices at the moment.

Conditioning and scalable solvers

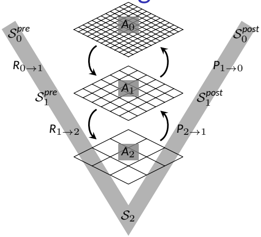
- $\mathcal{O}(n \log n)$ matrix-vector product in all cases → can explore iterative solvers
- Steady-state:
  - Fractional kernel, $\delta = \infty$[3]: $\kappa(\mathbf{A}) \sim h^{-2s} \sim n^{2s/d}$
  - Fractional kernel, $\delta \leq \delta_0$[4]: $\kappa(\mathbf{A}) \sim \delta^{2s-2} h^{-2s} \sim \delta^{2s-2} n^{2s/d}$
  - Integrable kernel, δ finite[8]: $\kappa(\mathbf{A}) \sim \delta^{-2}$
- Time-dependent:
  - $\kappa(\mathbf{M} + \Delta t \mathbf{A}) \sim 1 + \Delta t \, \kappa(\mathbf{A})$
  - Depending on time-stepper and CFL condition, this is well-conditioned for small s, large δ.

- Scalable solver options:
  - Multigrid
    - Geometric (GMG)
    - Algebraic (AMG)
  - Domain decomposition
    - Substructuring
    - Schwarz methods
  - Krylov methods
    The matrix is well-conditioned in the certain parameter regimes, e.g.
    - integrable kernel, δ large, or
    - or fractional kernel, s small, δ large.

[3] Mark Ainsworth, William McLean, and Thanh Tran. "The conditioning of boundary element equations on locally refined meshes and preconditioning by diagonal scaling". In: *SIAM Journal on Numerical Analysis* 36.6 (1999), pp. 1901–1932.

[4] Burak Aksoylu and Zuhal Unlu. "Conditioning analysis of nonlocal integral operators in fractional Sobolev spaces". In: *SIAM Journal on Numerical Analysis* 52.2 (2014), pp. 653–677.

Geometric multigrid in a nutshell



User specifies:

- Operators A_ℓ, assembled on hierarchy of nested meshes
- Transfer operators: prolongations $P_{\ell+1\to\ell}$, restrictions $R_{\ell\to\ell+1} = P_{\ell+1\to\ell}^T$,
- Smoothers $\mathcal{S}_\ell^{\text{pre/post}}$ (e.g. Jacobi)
- Coarse solver $\mathcal{S}_L$

How does multigrid work?

- On each level: smoother reduces high frequency error, low frequency error is transferred to coarser levels
- High/low frequency splitting depends on mesh

Drawbacks:

- Need hierarchy of nested meshes, complications for locally refined meshes
- Assembly on every level, tight coupling between assembly and solve

Geometric multigrid (GMG) for nonlocal equations

- Hierarchy of meshes from uniform or adaptive refinement
- Restriction / prolongation given by nesting of FE spaces
- Assembly into hierarchical or CSR matrix format on every level
- Smoothers:
  - Jacob, Chebyshev for $\mathcal{H}$-matrices
  - no constraints for CSR matrices.
- Coarse solve: convert to dense or CSR matrix

Test problems:

$$\begin{aligned} -\mathcal{L}u &= f \quad \text{in } \Omega = B_1(\mathbf{0}) \subset \mathbb{R}^2, \\ u &= 0 \quad \text{in } \Omega_I. \end{aligned}$$

1. $s \equiv 0.25$,
2. $s \equiv 0.75$,
3. $s = \begin{cases} 0.25 & x_0 \leq 0.1, \\ \text{cubic transition} & 0.1 \leq x_0 \leq 0.1, \\ 0.75 & 0.1 \leq x_0 \end{cases}$

Numerical Examples – errors

Single core on Intel Broadwell (SNL/Solo)

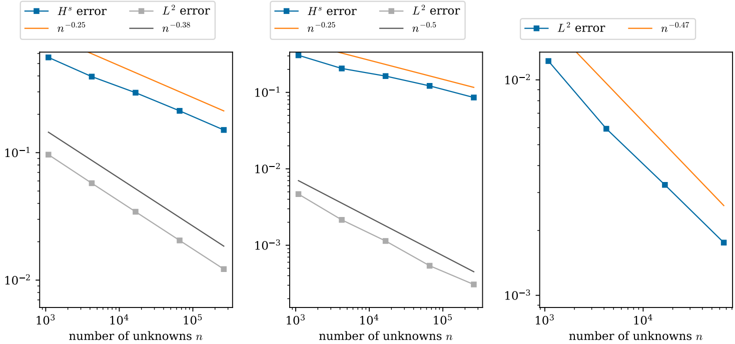


Figure: Timings and memory usage for assembly of the stiffness matrix for fractional kernels, $\delta = \infty$ and solution of linear system using Geometric Multigrid for the two-dimensional problem.
$s = 0.25$ on the left, $s = 0.75$ in the middle, $s(\boldsymbol{x})$ on the right.

Numerical Examples – assembly and solve

Single core on Intel Broadwell (SNL/Solo)

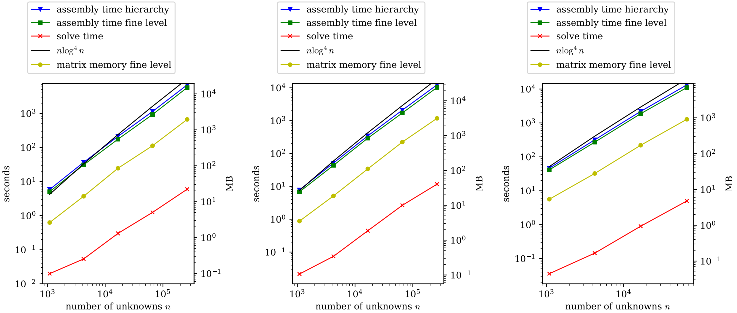


Figure: Timings and memory usage for assembly of the stiffness matrix for fractional kernels, $\delta = \infty$ and solution of linear system using Geometric Multigrid for the two-dimensional problem.
$s = 0.25$ on the left, $s = 0.75$ in the middle, $s(\boldsymbol{x})$ on the right.

Numerical Examples – weak scaling

Intel Broadwell (SNL/Solo)

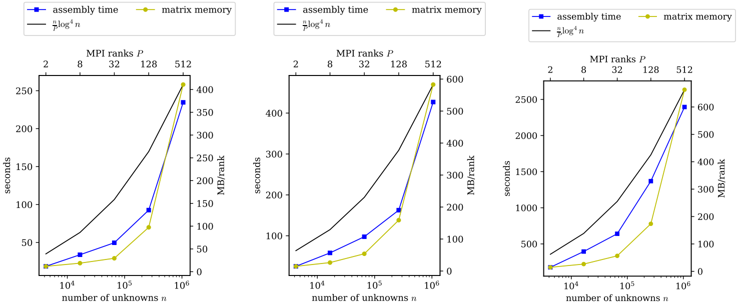


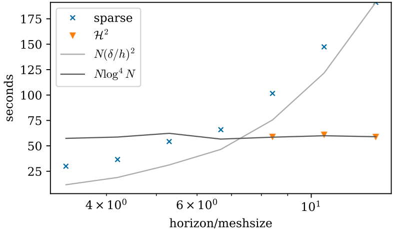
Figure: Timings and memory usage for assembly of the stiffness matrix for fractional kernels, $\delta = \infty$ for the two-dimensional problem.
$s = 0.25$ on the left, $s = 0.75$ in the middle, $s(\boldsymbol{x})$ on the right.

- Bad load balancing for $s(\boldsymbol{x})$

Hierarchical matrices for finite horizon kernels

- Regular CSR sparse matrices are efficient (complexity, memory) for small horizon $\delta \sim h$.
- $\mathcal{H}^2$-matrices are efficient for $\delta = \infty$.
- Low-rank approximation relies on smoothness of the kernel
  $\rightarrow$ need to fully assemble entries near $\partial B_\delta(\boldsymbol{x})$.

At what ratio δ/h do $\mathcal{H}^2$-matrices become more efficient than sparse matrices?



- $\Omega \subset \mathbb{R}^2, \gamma(\boldsymbol{x},\boldsymbol{y}) = c_{d,\delta}\mathcal{X}_{|\boldsymbol{x}-\boldsymbol{y}|\leq\delta}$
- Break-even:
  - 1D: $\delta/h \sim 100 - 200$
  - 2D: $\delta/h \sim 5 - 10$
- Break-even depends on:
  - Cost of quadrature
  - Lots of implementation details

Smoothed Aggregation Algebraic multigrid (SA-AMG)

Drawbacks of Geometric Multigrid

- Good (nested) coarse meshes might be hard to come by for adaptively refined or graded meshes
- FE assembly of operators on all levels is not cheap

Algebraic multigrid mimics geometric multigrid while only using algebraic information.

Inputs:

- A_0, DoF coordinates $\boldsymbol{c}$, near-nullspace (constant function, rigid body modes, ...)

AMG setup

- construction of transfer operators $P_{\ell+1\to\ell}$ using algebraic information (e.g. matrix graph, strength of connection)
- Galerkin projection $A_{\ell+1} = P^T_{\ell+1\to\ell} A_\ell P_{\ell+1\to\ell}$

Issues for nonlocal problems:

- Many graph algorithms used in AMG cannot be applied to $\mathcal{H}$-matrices as they rely access to matrix entries.
- Inefficient for operators that are too dense.
- Hierarchical information contained in $\mathcal{H}$-matrix does not translate directly to a multigrid hierarchy.

Auxiliary operator multigrid

Idea

- Construct multigrid transfer operators $\left\{P_{\ell+1,\ell}\right\}$ wrt an auxiliary matrix $\widetilde{A}_0$.
- Then construct preconditioner via Galerkin projections $A_{\ell+1} = P^T_{\ell+1\to\ell} A_\ell P_{\ell+1\to\ell}$.

Requirements for auxiliary operator $\widetilde{A}_0$:

- sparse
- contains sufficient information about nonlocal problem (mesh, coefficients, kernel, ...)

Possible auxiliary operators:

- PDE Laplacian on the same mesh
- distance Laplacian on graph G of filtered near-field matrix

$$L_{ij} = \begin{cases} -1/\left|\boldsymbol{c}_i - \boldsymbol{c}_j\right| & \text{if } (i,j) \in G, i \neq j, \\ -\sum_{k\neq i} L_{ik} & \text{if } i = j, \end{cases} \qquad (\boldsymbol{c}_i \text{ are DoF coordinates})$$

- lumped and re-scaled near-field matrix

Galerkin projection with $\mathcal{H}^2$-matrices

Galerkin projection:
If

$$A = A_{\text{near}} + B\left[(I+T_K)\cdots(I+T_1)\right]\Gamma\left[(I+T_1)^T\cdots(I+T_K)^T\right]B^T,$$

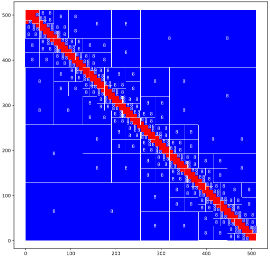
then

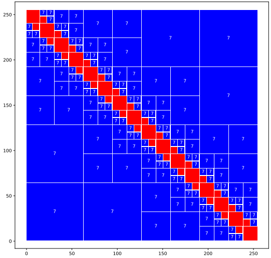
$$P^TAP = \underbrace{P^TA_{\text{near}}P}_{\text{multiplied out}} + \underbrace{(P^TB)}_{\text{multiplied out}}\left[(I+T_K)\cdots(I+T_1)\right]\Gamma\left[(I+T_1)^T\cdots(I+T_K)^T\right](P^TB)^T.$$

- Galerkin projection of $\mathcal{H}^2$-matrix is an $\mathcal{H}^2$-matrix.
- Uses the same compression of the off-rank matrix blocks.
  $\rightarrow$ Will become inefficient, especially after several multigrid coarsenings.
  $\rightarrow$ Need to recompress coarse operators

Recompression: What would geometric MG do?

1D example, mesh sizes h and $2h$





- <span>red</span> = near field cluster pairs
- blue = far field cluster pairs, rank of approximation in white
- cluster pairs that are admissible on coarse mesh are admissible on fine mesh, but approximation rank differs
- some cluster pairs on fine mesh are not admissible on coarse mesh, since represented block becomes to small to be represented by low-rank matrices

Recompression strategy

- Keep ranks of approximations.
- Multiply out small low-rank clusters and add to near field.

Recompression in terms of matrix operations

In order to drop the last level K of the operator

$$A = A_{\text{near}} + B\left[(I+T_K)\cdots(I+T_1)\right]\Gamma\left[(I+T_1)^T\cdots(I+T_K)^T\right]B^T,$$

we split off the last level of cluster interactions

$$\Gamma = \tilde{\Gamma} + \Gamma_K$$

and reorder terms

$$\begin{aligned} A = &\left[A_{\text{near}} + B(I+T_K)\Gamma_K(I+T_K)^T B^T\right] \\ &+ \left[B(I+T_K)\right]\left[(I+T_{K-1})\cdots(I+T_1)\right]\tilde{\Gamma}\left[(I+T_1)^T\cdots(I+T_{K-1})^T\right]\left[B(I+T_K)\right]^T \end{aligned}$$

leading to

$$A = \tilde{A}_{\text{near}} + \tilde{B}\left[(I+T_{K-1})\cdots(I+T_1)\right]\tilde{\Gamma}\left[(I+T_1)^T\cdots(I+T_{K-1})^T\right]\tilde{B}^T$$

where

$$\tilde{B} = B(I+T_K), \qquad \tilde{A}_{\text{near}} = A_{\text{near}} + \tilde{B}\Gamma_K\tilde{B}^T$$

are explicitly formed.

- All operations required for Galerkin projection and recompression use sparse matrix-matrix addition & multiplication.
- Since the ranks for low-rank blocks and transfers do not change, only $\tilde{A}_{\text{near}}$ and $\tilde{B}$ need to be explicitly saved on coarse levels.

Implementation details

Components:

- PyNucleus[5] for assembly of nonlocal operators
- Trilinos/Tpetra[6] for distributed sparse linear algebra
- Trilinos/Belos[7] for Krylov solvers
- Trilinos/MueLu[7] for Algebraic Multigrid
- Kokkos[7] programming model for performance portability

Features:

- $\mathcal{H}$- and $\mathcal{H}^2$-matrices, reader for hierarchical operators
- Krylov solvers, AMG preconditioner
- MPI distributed
- Compute architectures supported by Kokkos:
  CPU (Serial, OpenMP), GPU (Cuda, HIP, ...), ...

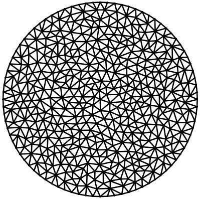
[5] `https://github.com/sandialabs/PyNucleus`

[6] `https://github.com/trilinos/Trilinos`

[7] `https://github.com/kokkos/kokkos`

Numerical results - CPU

Solo, SNL, Broadwell CPUs



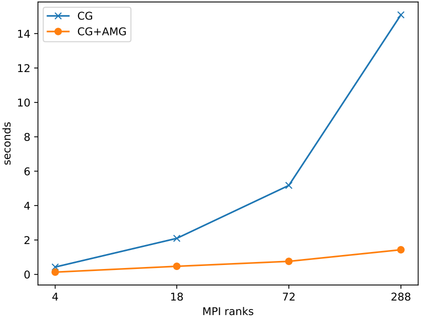
- Quasi-uniform mesh, P1 elements
- 2 Jacobi sweeps of pre-/post-smoothing
- LAPACK coarse solve

		memory (finest level)		iterations (time)	
DoFs	ranks	dense	$\mathcal{H}^2$	PDE Δ	distance Δ
12,173	4	1.1 GB	0.1 GB	8 (0.15s)	8 (0.14s)
49,139	18	18 GB	0.55 GB	8 (0.47s)	9 (0.54s)
197,565	72	291 GB	3 GB	9 (0.73s)	10 (0.84s)
792,548	288	4,680 GB	19.7 GB	9 (1.43s)	10 (1.56s)
n	n	n^2	$n \log^4 n$	constant ($\log^4 n$)	

Table: 2D fractional Poisson problem on unit disk, $s = 0.75$, $\delta = \infty$

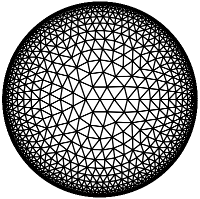
- Dense matrices only for comparison.
- Only the first two dense problems would actually fit in memory on their respective job allocations.

Numerical results - Comparison with unpreconditioned CG



- Both solvers use a $\mathcal{H}^2$-matrix.
- AMG preconditioned solve is scalable, Krylov by itself is not.

Numerical results - graded meshes



- Motivation: resolution of low regularity near domain boundary improves convergence of discretization error
- Weak scaling of solve time needs work (load balancing).

			memory (finest level)		iterations (time)
DoFs	$h_{\max}/h_{\min}$	ranks	dense	$\mathcal{H}^2$	CG+SA-AMG
15,852	105	4	1.87 GB	0.33 GB	7 (0.37s)
78,674	218	18	46.1 GB	2.4 GB	7 (1.74s)
363,472	439	72	984.3 GB	16.6 GB	8 (3.73s)

Table: 2D fractional Poisson problem on graded unit disk, $s = 0.75$, $\delta = \infty$

Numerical results - GPU

Lassen, LLNL, V100 GPUs

		memory (finest level)		iterations (time)
DoFs	ranks	dense	$\mathcal{H}^2$	CG+SA-AMG
49,139	4	18 GB	0.6 GB	9 (0.12s)
197,565	16	291 GB	2.9 GB	11 (0.29s)
792,548	64	4,680 GB	14.7 GB	12 (0.62s)
3,175,042	256	75,109 GB	61.9 GB	12 (1.79s)

Table: 2D fractional Poisson problem on unit disk, $s = 0.75$, $\delta = \infty$

- 1000x reduction in memory
- Weak scaling behavior can be improved
  (no AMG parameter tuning for GPU so far)

Domain decomposition: Schwarz methods (WIP, with Pierre Marchand (INRIA))

- Schwarz method
  - overlapping subdomain restrictions $\{\boldsymbol{R}_p\}$, local matrices $\boldsymbol{A}_p = \boldsymbol{R}_p \boldsymbol{A} \boldsymbol{R}_p^T$
  - partition of unity $\sum_{p=1}^P \boldsymbol{R}_p^T \boldsymbol{D}_p \boldsymbol{R}_p = \boldsymbol{I}$, with $\{\boldsymbol{D}_p\}$ diagonal
  - additive Schwarz preconditioner: $\boldsymbol{Q}_1 := \sum_{p=1}^P \boldsymbol{R}_p^T \boldsymbol{A}_p^{-1} \boldsymbol{R}_p$, or restricted additive Schwarz
- No global information exchange → need a coarse grid
- GenEO approach:
  Span coarse space using solutions of subdomain eigenvalue problems $\boldsymbol{D}_p \boldsymbol{A}_p \boldsymbol{D}_p \boldsymbol{v}_{p,k} = \lambda_{p,k} \boldsymbol{B}_p \boldsymbol{v}_{p,k}$, where $\boldsymbol{B}_p$ is similar to $\boldsymbol{A}_p$, but assembled over a modified local mesh.
- Distributed $\mathcal{H}$-matrix is built using Pierre Marchand's Htool library `https://github.com/htool-ddm/htool`
- HPDDM library for Schwarz DD and GenEO `https://github.com/hpddm/hpddm`
- 2D fractional Poisson problem, $s = 0.75$, $\delta = \infty$ on Sandia's Solo machine

		memory (finest level)		iterations (time)
unknowns	# MPI ranks	dense	$\mathcal{H}$	GMRES+DD
65,025	72	31.5 GB	5.4 GB	21 (1.34s)
261,121	288	508 GB	12.6 GB	23 (0.96s)
1,046,529	1152	8,160 GB	86 GB	24 (2.4s)

- Caveats:
  - solver setup needs improvement, working on alternative low-rank approximations
  - direct solves (subdomain, coarse) and eigenvalue problems in dense format

Advertisement: PyNucleus, a FEM code for nonlocal problems

- Interface in Python, computational kernels via Cython, C, C++.
- Compatible with NumPy/SciPy
- Simplical meshes in 1D, 2D, 3D; refinement with boundary snapping options
- MPI distributed computations via mpi4py
- Partitioning using (Par)METIS
- Finite Element discretizations: discontinuous P_0, continuous P_1, P_2, P_3
- Assembly of local differential operators
- Lots of solvers (direct, Krylov, simple preconditioners),
  and in particular geometric multigrid
  WIP: AMG (Trilinos/MueLu), DD (Htool&HPDDM)
- Assembly of the nonlocal operators in weak form into
  - CSR sparse matrix ($\delta \sim h$),
  - dense matrix ($\delta \gg h$),
  - $\mathcal{H}^2$ hierarchical matrix ($\delta \gg h$)
- Code: `github.com/sandialabs/PyNucleus`
- Documentation and examples: `sandialabs.github.io/PyNucleus`
- Available via cloud-hosted Jupyter notebook, container image, Spack package `py-pynucleus`

Code example

```python
from PyNucleus import (kernelFactory, nonlocalMeshFactory, dofmapFactory,
                       functionFactory, HOMOGENEOUS_DIRICHLET, solverFactory)

# Infinite horizon fractional kernel
kernel = kernelFactory('fractional', dim=2, s=0.75, horizon=inf)

# Mesh for unit disc, no interaction domain for homogeneous Dirichlet
mesh, _ = nonlocalMeshFactory('disc', kernel=kernel,
                              boundaryCondition=HOMOGENEOUS_DIRICHLET,
                              hTarget=0.15)

dm = dofmapFactory('P1', mesh)                        # P1 finite elements
f = functionFactory('constant', 1.)                   # constant forcing
b = dm.assembleRHS(f)                                 # $\int_\Omega f\phi_i$
A = dm.assembleNonlocal(kernel, matrixFormat='h2')   # $a(\phi_i, \phi_j)$, $\mathcal{H}^2$-matrix
u = dm.zeros()                                        # solution vector

# solve with diagonally preconditioned CG
solver = solverFactory('cg-jacobi', A=A, setup=True)
solver(b, u)
u.plot()
```

Conclusion:

- Discretized nonlocal equations are often dense, but not structurally dense.
  $\rightarrow$ Assembly and matrix-vector product in $\mathcal{O}(n \log^{2d} n)$ operations and memory
- Multigrid (and domain decomposition) are also optimal for elliptic nonlocal problems.
- Auxiliary operator approach allows to apply algebraic multigrid to hierarchical matrices.
- Sparse matrix representation of hierarchical matrices allows to leverage a lot of existing code.

Outlook:

- Coefficients variations (AMG should be good for that!)
- AMG for nonlocal operators in sparse format (δ/h small but denser than PDEs)
- AMG for boundary integral equations
- Inverse problems with variable order fractional Laplacians, Bayesian inference

Thanks for listening!

Funded by LDRD projects at Sandia:

- *MATNIP: MAThematical foundations for Nonlocal Interface Problems: multiscale simulations for heterogeneous materials (FY20-22)*
- *FOMSI: Fractional-Order Models for Sea Ice (FY23-25)*

References I

[1] Gabriel Acosta and Juan Pablo Borthagaray. “A fractional Laplace equation: regularity of solutions and Finite Element approximations”. In: *SIAM Journal on Numerical Analysis* 55.2 (2017).

[2] Mark Ainsworth and Christian Glusa. “Aspects of an adaptive finite element method for the fractional Laplacian: A priori and a posteriori error estimates, efficient implementation and multigrid solver”. In: *Computer Methods in Applied Mechanics and Engineering* (2017).

[3] Mark Ainsworth, William McLean, and Thanh Tran. “The conditioning of boundary element equations on locally refined meshes and preconditioning by diagonal scaling”. In: *SIAM Journal on Numerical Analysis* 36.6 (1999).

[4] Burak Aksoylu and Zuhal Unlu. “Conditioning analysis of nonlocal integral operators in fractional Sobolev spaces”. In: *SIAM Journal on Numerical Analysis* 52.2 (2014).

[5] Peter W. Bates. “On some nonlocal evolution equations arising in materials science”. In: *Nonlinear dynamics and evolution equations* 48 (2006).

[6] David A. Benson, Stephen W. Wheatcraft, and Mark M. Meerschaert. “Application of a fractional advection-dispersion equation”. In: *Water Resources Research* 36.6 (2000).

[7] Giacomo Capodaglio et al. “An energy-based coupling approach to nonlocal interface problems”. In: *Computers & Fluids* 207 (2020).

[8] Marta D’Elia et al. “Numerical methods for nonlocal and fractional models”. In: *Acta Numerica* 29 (2020).

References II

[9] Marta D'Elia, Max Gunzburger, and Christian Vollmann. "A cookbook for approximating Euclidean balls and for quadrature rules in finite element methods for nonlocal problems". In: *Mathematical Models and Methods in Applied Sciences* 31.08 (2021).

[10] Paolo Gatto and Jan S Hesthaven. "Numerical approximation of the fractional laplacian via hp-finite elements, with an application to image denoising". In: *Journal of Scientific Computing* 65.1 (2015).

[11] Gerd Grubb. "Fractional Laplacians on domains, a development of Hörmander's theory of μ-transmission pseudodifferential operators". In: *Advances in Mathematics* 268 (2015).

[12] Manuel Klar et al. "A scalable domain decomposition method for FEM discretizations of nonlocal equations of integrable and fractional type". In: *arXiv preprint arXiv:2306.00094* (2023).

[13] Anna Lischke et al. "What is the fractional Laplacian? A comparative review with new results". In: *Journal of Computational Physics* 404 (2020).

[14] Yifei Lou et al. "Image recovery via nonlocal operators". In: *Journal of Scientific Computing* 42.2 (2010).

[15] B. M. McCay and M. N. L. Narasimhan. "Theory of nonlocal electromagnetic fluids". In: *Archives of Mechanics* 33.3 (1981).

[16] Mark M. Meerschaert and Alla Sikorskii. *Stochastic models for fractional calculus*. Vol. 43. de Gruyter Studies in Mathematics. Walter de Gruyter & Co., Berlin, 2012.

References III

[17] Stewart A. Silling. “Reformulation of elasticity theory for discontinuities and long-range forces”. In: *Journal of the Mechanics and Physics of Solids* 48.1 (2000).

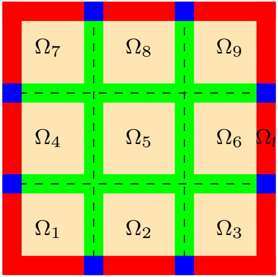
[18] Peter Tankov. *Financial modelling with jump processes*. Vol. 2. CRC press, 2003.

[19] Xiao Xu et al. “A FETI approach to domain decomposition for meshfree discretizations of nonlocal problems”. In: *Computer Methods in Applied Mechanics and Engineering* 387 (2021).

Domain decomposition: FETI[8,9,10]

Substructuring - FETI

- Break up global system into subdomains.
- Couple via Lagrange multipliers on interfaces.



For nonlocal operators with horizon $\delta = \mathcal{O}(h)$:

- Cover with **overlapping** subdomains $\Omega \cup \Omega_I = \bigcup \Omega_i$, diam $(\Omega_i \cap \Omega_j) \sim \delta$ for adjacent subdomains.
- Duplicate unknowns in overlaps:

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} \Leftrightarrow \begin{pmatrix} \boldsymbol{A}_{\epsilon\epsilon} & \boldsymbol{M}^T \\ \boldsymbol{M} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_\epsilon \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_\epsilon \\ 0 \end{pmatrix}$$

- $\boldsymbol{A}_{\epsilon\epsilon}$ is block diagonal.
- For floating subdomains, local matrix $\boldsymbol{A}_p$ is singular.
- Binary matrix $\boldsymbol{M}$ encodes constraints on the overlaps.
- Use projected CG solver, “Dirichlet” preconditioner.

[8] Giacomo Capodaglio, Marta D’Elia, Pavel Bochev, and Max Gunzburger. “An energy-based coupling approach to nonlocal interface problems”. In: *Computers & Fluids* 207 (2020), p. 104593.

[9] Xiao Xu, Christian Glusa, Marta D’Elia, and John T. Foster. “A FETI approach to domain decomposition for meshfree discretizations of nonlocal problems”. In: *Computer Methods in Applied Mechanics and Engineering* 387 (2021), p. 114148.

[10] Manuel Klar, Giacomo Capodaglio, Marta D’Elia, Christian Glusa, Max Gunzburger, and Christian Vollmann. “A scalable domain decomposition method for FEM discretizations of nonlocal equations of integrable and fractional type”. In: *arXiv preprint arXiv:2306.00094* (2023).

Weak scaling – 2D, constant kernel

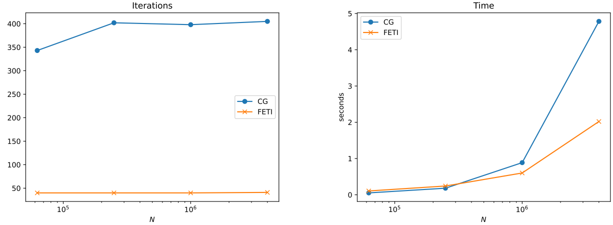


Figure: $\delta = 8e - 3 \to \kappa \sim$ const

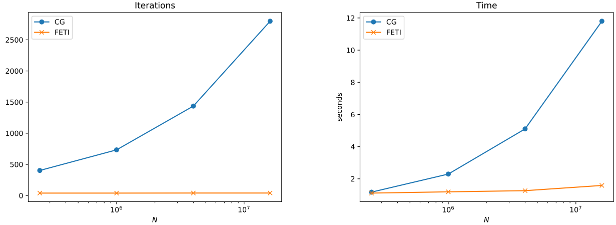


Figure: $\delta = 4h \to \kappa \sim n$

Weak scaling – 2D, fractional kernel, $s = 0.4$

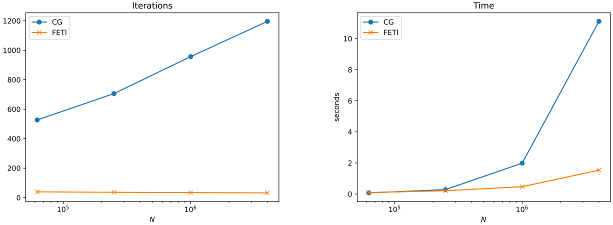


Figure: $\delta = 8e - 3 \rightarrow \kappa \sim n^s$

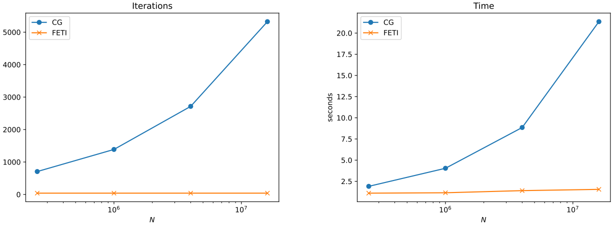


Figure: $\delta = 4h \rightarrow \kappa \sim n$

Strong scaling, 2D

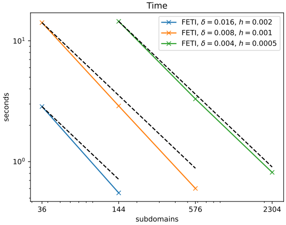


Figure: constant kernel, $\delta = 8h$.

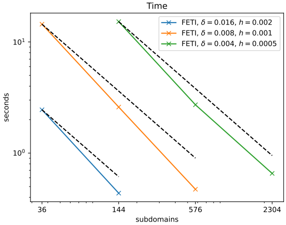


Figure: fractional kernel, $s = 0.4$, $\delta = 8h$.

Substructuring: Reduced system and Dirichlet preconditioner

- Let nullspace of $\boldsymbol{A}_{\epsilon\epsilon}$ be given by $\boldsymbol{Z}$.
- Eliminate primal variables from

$$\begin{pmatrix} \boldsymbol{A}_{\epsilon\epsilon} & \boldsymbol{M}^T \\ \boldsymbol{M} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_\epsilon \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_\epsilon \\ 0 \end{pmatrix}$$

  and obtain

$$\begin{aligned} \boldsymbol{P}_0 K\lambda &= \boldsymbol{P}_0(\boldsymbol{M}\boldsymbol{A}_{\epsilon\epsilon}^{\dagger}\boldsymbol{f}_\epsilon) \\ \boldsymbol{G}^T\boldsymbol{\lambda} &= \boldsymbol{Z}^T\boldsymbol{f}_\epsilon, \end{aligned}$$

  where $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{A}_{\epsilon\epsilon}^{\dagger}\boldsymbol{M}^T$, $\boldsymbol{G} = \boldsymbol{M}\boldsymbol{Z}$, $\boldsymbol{P}_0 = \boldsymbol{I} - \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^{\dagger}\boldsymbol{G}^T$.
- Use projected CG to solve system.
- $\boldsymbol{P}_0$ acts as a “coarse grid”.
- Preconditioner for $\boldsymbol{K}$:
  - Let $\boldsymbol{A}_p$, $\boldsymbol{M}_p$ be local parts of $\boldsymbol{A}_{\epsilon\epsilon}$ and $\boldsymbol{M}$.
  - Write $\boldsymbol{K} = \sum_{p=1}^{P} \boldsymbol{M}_p\boldsymbol{A}_p^{\dagger}\boldsymbol{M}_p^T = \sum_{p=1}^{P} \widetilde{\boldsymbol{M}}_p\boldsymbol{S}_p^{\dagger}\widetilde{\boldsymbol{M}}_p^T$.
  - Dirichlet preconditioner: $\boldsymbol{Q} = \sum_{p=1}^{P} \widetilde{\boldsymbol{M}}_p\boldsymbol{S}_p\widetilde{\boldsymbol{M}}_p^T$.
- Results shown use Manuel Klar’s (U of Trier) assembly code `https://gitlab.uni-trier.de/klar/nonlocal-assembly`