SHF: Small: Collaborative Research: Transform-to-perform: languages, algorithms, and solvers for nonlocal operators

Andreas Klöckner, Robert Kirby

Computer Science, University of Illinois, Urbana, IL; Mathematics, Baylor University, Waco, TX

SHF-1911019/1909176

Nonlocal Operators

Very broad class of mathematical operations, targeted examples:

- Layer potentials
- Volume potentials
- Fractional derivatives

$$u(\mathbf{x}) = \int_\Omega K(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\mathrm{d}\mathbf{y},$$

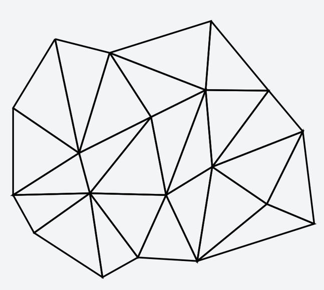
Inputs:

- Ω a (volume or surface) source geometry, given as a high-order mapped, unstructured mesh for geometric flexibility
- kernel $K(\mathbf{x},\mathbf{y})$, as a math. expression,
- density $\sigma(\mathbf{y})$ as input data,
- $\mathbf{x}$ as target var., $\mathbf{y}$ as source var.

Non-Local Operators are increasingly used in coupled simulations with conventional derivative/PDE-based modeling. Derivatives are local operators.

Background: Firedrake/UFL

- Finite Elements: PDEs/derivatives modeled via a *variational form*
  - on computational meshes representing geometry
- UFL: A DSL for expressing variational forms
- Firedrake: A framework for high-performance UFL evaluation at scale



```
V1 = FunctionSpace(mesh, 'RT', 1)
V2 = FunctionSpace(mesh, 'DG', 0)
W = V1 * V2
u, p = TrialFunctions(W)
v, q = TestFunctions(W)
f = Function(V2)
a = (p*q - q*div(u)
  + inner(v, u)
    + div(v)*p)*dx
L = f*q*dx
u = Function(W)
solve(a == L, u)
```

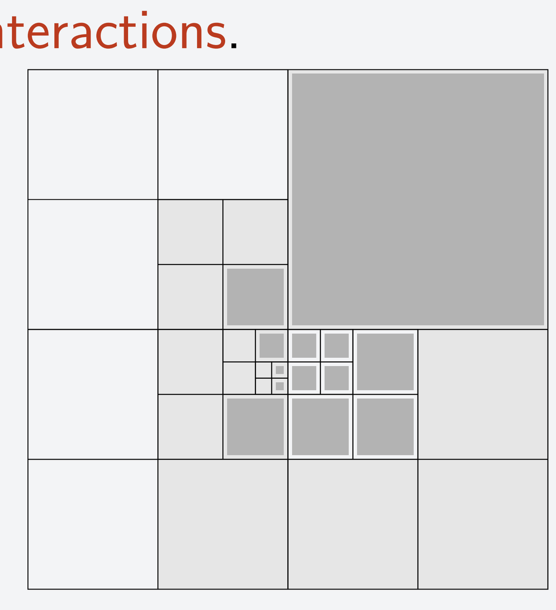
`https://firedrakeproject.org`

Background: Fast Algorithms

Naive evaluation of non-local operators requires $O(N^2)$ operations:

- Not scalable
- Not computationally feasible in 3D

Needs *fast algorithms* such as variants of Fast Multipole, making use of compressibility of far-field interactions.



NUFL

An extension of UFL to nonlocal operators:

```
k = Parameter("k")
x = TargetVariable(ambient_dim=2)
y = SourceVariable(ambient_dim=2)
kernel = ExpressionKernel(
  1/(2*pi)*hankel_1_0(
    k*sqrt((x-y)@(x-y))))

wspace = FunctionSpace(
  Surface(mesh), "Lagrange", 3)
w = Function(wspace)
v = TestFunction(wspace)
bdry_op_sym = inner(
  -0.5*w
  +  kernel.conv(w)
  - (grad(y)(kernel)
    @ FacetNormal(mesh)
   ).conv(w), v) * dx
```

- Allows seamless integration of local/nonlocal modeling
- Layer potentials, fractional derivatives, and conventional FEM all "under one roof"

Computational realization:

- Pytential with GIGAQBX TS fast algorithm [Wala, Klöckner '20] for surface potentials
- Volumential for volume potentials

Fractional Derivatives

Example: Fractional Laplacian

$$(-\triangle)^s u(x) = \frac{2^{2s}\Gamma(s+n/2)}{\pi^{n/2}\Gamma(1-s)} \mathrm{PV}\int_\Omega \frac{u(x)-u(y)}{\|x-y\|_2^{n+2s}}\mathrm{d}y$$

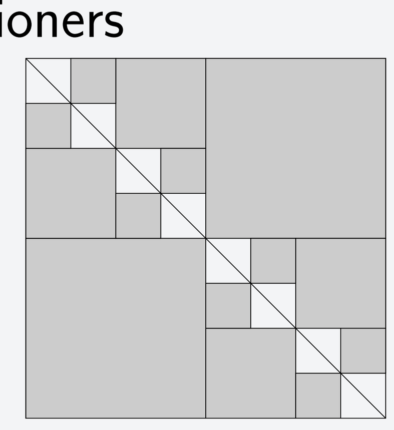
Shortcoming in many computational experiments so far: no fast algorithm, $O(N^2)$ scaling.

Idea: "a volume potential with odd kernel"
Reality: More complex, but not too far

Solvers, Symbolic/HSS Matrix Integration

Successful solution of a coupled FEM/non-local system of equations requires sophisticated solver technology.
Example: Nonloccal operators can precondition themselves and FEM systems. Important to represent nonlocal operators in matrix form. For example as Hierarchically Semiseparable Matrices.
Idea: Extend UFL to represent solver infrastructure:

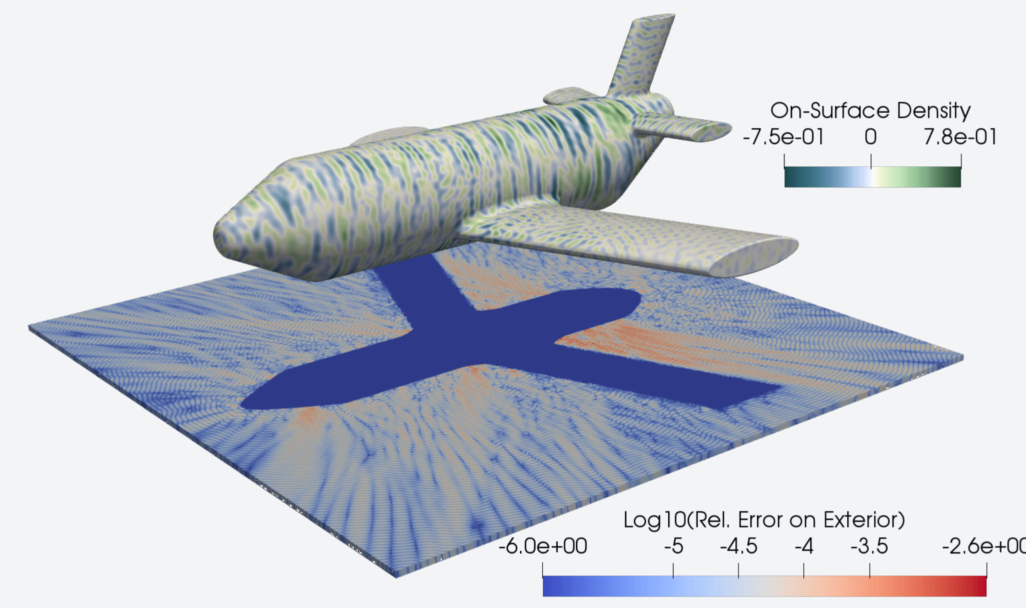
- matrix-free/HSS/sparse
- Schur complements
- Preconditioners



(shaded blocks have low rank)

Results: Surface Potentials

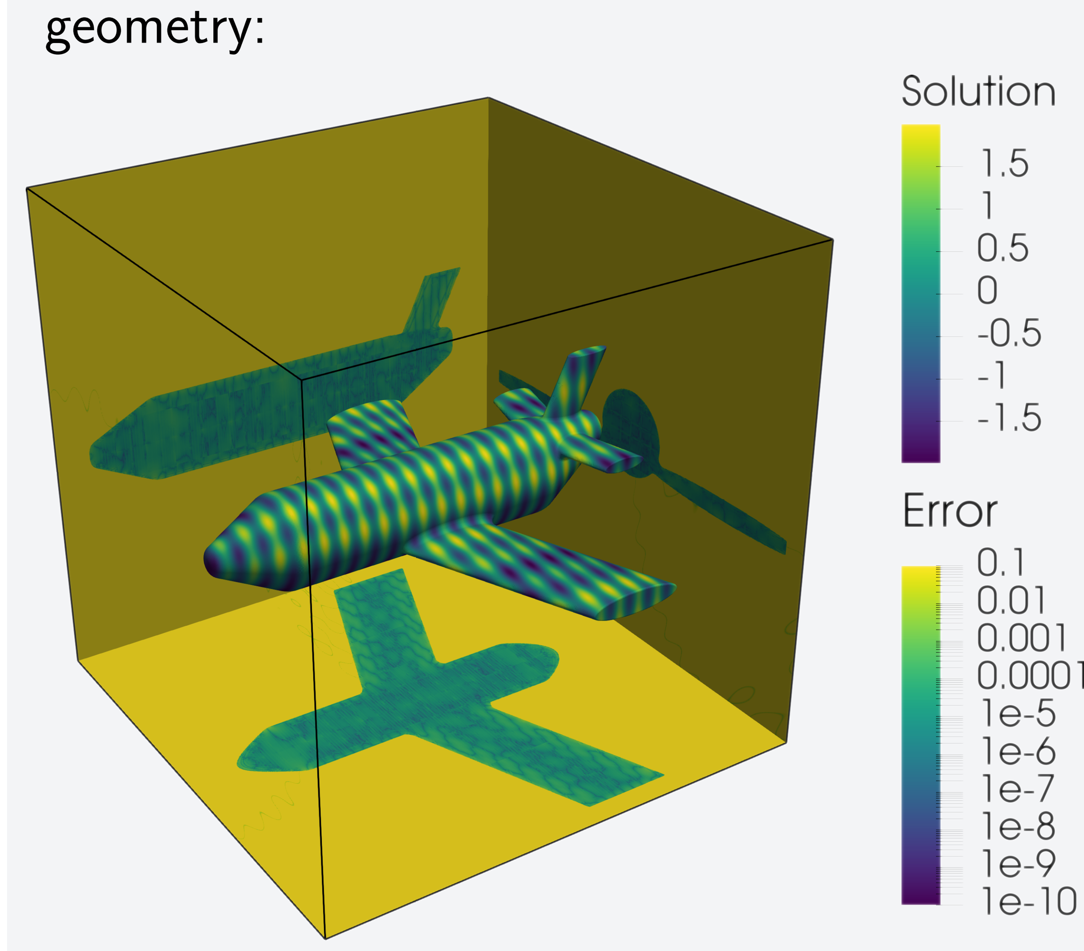
An exterior, moderate-frequency Helmholtz problem in complex geometry:



- Linearly scaling fast algorithm
- Predictable error behavior

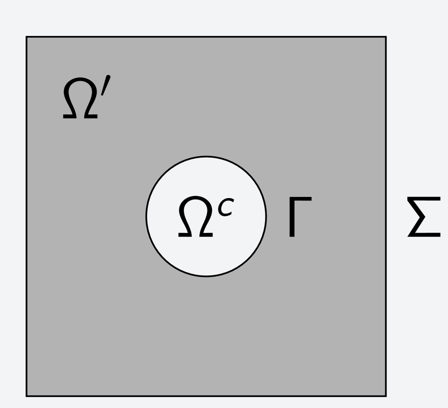
Results: Volume Potentials

An interior Poisson problem in complex geometry:



- Linearly scaling fast algorithm
- Predictable error behavior

Results: Domain Truncation



$$\begin{aligned}(\nabla u,\nabla v) - \kappa^2(u,v) - i\kappa\langle u,v\rangle_\Sigma \\ + \left\langle\left(i\kappa - \tfrac{\partial}{\partial n}\right) D(u), v\right\rangle_\Sigma \\ = \langle f,v\rangle_\Gamma + \left\langle\left(i\kappa - \tfrac{\partial}{\partial n}\right) S(f), v\right\rangle_\Sigma\end{aligned}$$

- PML complicates solvers
- Retain ellipticity, optimal error estimates + preconditioners
- Impl.: Firedrake + Pytential
  - (NUFL: future work)

Summary

- Local+nonlocal modeling: attractive but technically complex
  - Moreso in complex geometry
- Computational tools will help make this field more accessible
- Fast algorithms for matvecs and solvers will ensure computational scalability

Baylor University ILLINOIS