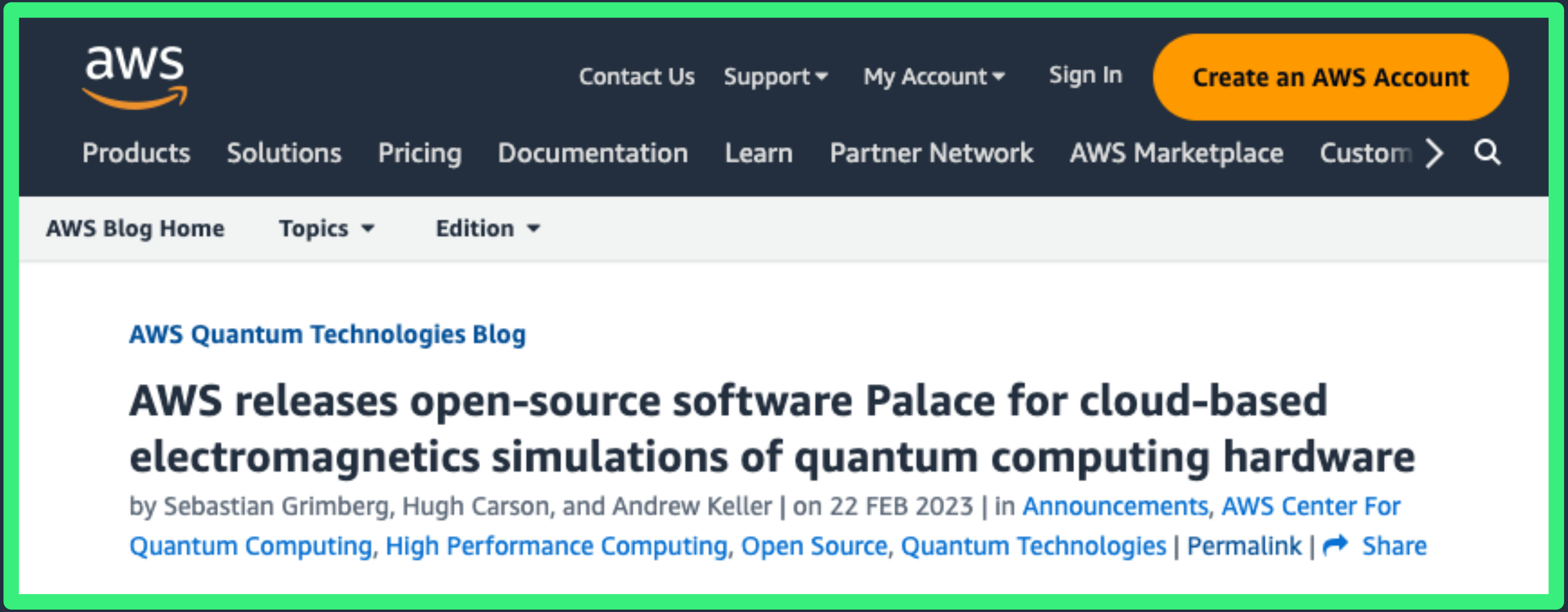
MFEM COMMUNITY WORKSHOP 2023

Palace: PArallel LArge-scale Computational Electromagnetics

Sebastian Grimberg

Sr. Applied Scientist
AWS

Palace: 3D finite element solver for computational electromagnetics



Contact Us | Support | My Account | Sign In | Create an AWS Account

Products | Solutions | Pricing | Documentation | Learn | Partner Network | AWS Marketplace | Custom

AWS Blog Home | Topics | Edition

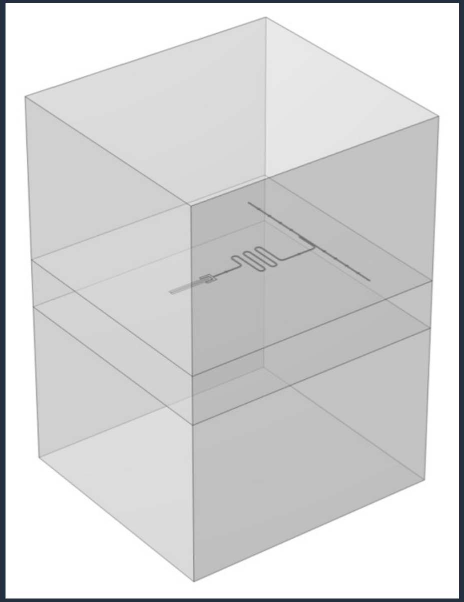
AWS Quantum Technologies Blog

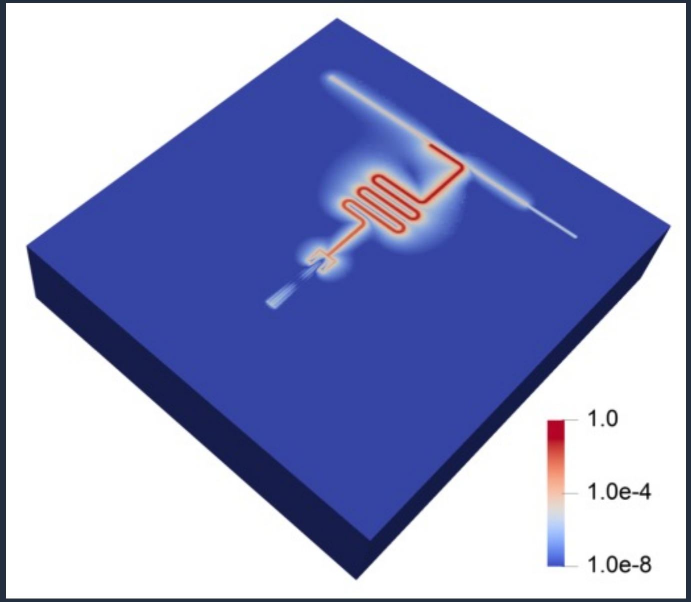
AWS releases open-source software Palace for cloud-based electromagnetics simulations of quantum computing hardware

by Sebastian Grimberg, Hugh Carson, and Andrew Keller | on 22 FEB 2023 | in Announcements, AWS Center For Quantum Computing, High Performance Computing, Open Source, Quantum Technologies | Permalink | Share

Palace is developed at the AWS Center for Quantum Computing
to enable the design of quantum computing hardware

Palace: 3D finite element solver for computational electromagnetics





Uses MFEM and libCEED to perform full-wave 3D electromagnetic simulations in the frequency and time domain

Frequency and time domain electromagnetics models

Driven

$$\nabla \times \left(\mu^{-1}\nabla \times \boldsymbol{E}\right) + i\omega\sigma\boldsymbol{E} - \omega^2\varepsilon\boldsymbol{E} = 0$$

$$\boldsymbol{n} \times \boldsymbol{E} = 0,\ \boldsymbol{x} \in \partial\Omega_D$$

$$\boldsymbol{n} \times \left(\mu^{-1}\nabla \times \boldsymbol{E}\right) + i\omega Z_s^{-1}\boldsymbol{n} \times (\boldsymbol{n} \times \boldsymbol{E}) = \boldsymbol{U}_s,\ \boldsymbol{x} \in \partial\Omega_R$$

Transient

$$\nabla \times \left(\mu^{-1}\nabla \times \boldsymbol{E}\right) + \sigma\frac{\partial \boldsymbol{E}}{\partial t} + \varepsilon\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = 0$$

$$\boldsymbol{n} \times \boldsymbol{E} = 0,\ \boldsymbol{x} \in \partial\Omega_D$$

$$\boldsymbol{n} \times \left(\mu^{-1}\nabla \times \boldsymbol{E}\right) + Z_s^{-1}\boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}}{\partial t}\right) = \boldsymbol{U}_s,\ \boldsymbol{x} \in \partial\Omega_R$$

- Lumped ports, numeric wave ports, scattering parameter calculations
- Anisotropic and lossy material properties
- Absorbing boundary conditions

Frequency and time domain electromagnetics models

Eigenmode

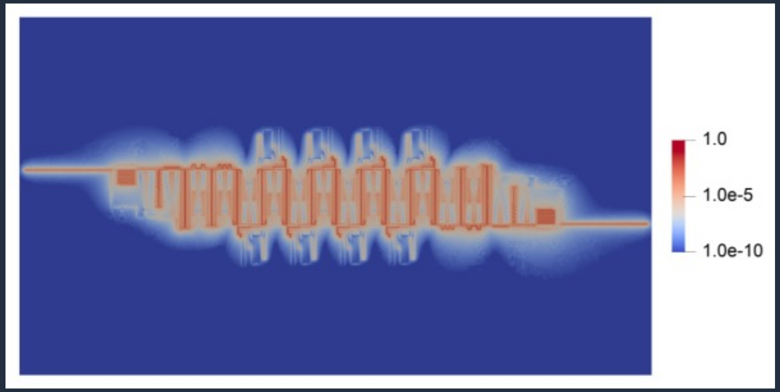
$$\left(\boldsymbol{K} + i\omega\boldsymbol{C} - \omega^2\boldsymbol{M}\right)\boldsymbol{u} = 0$$

$$\begin{bmatrix} 0 & \boldsymbol{I} \\ \boldsymbol{K} & i\boldsymbol{C} \end{bmatrix}\begin{bmatrix} \boldsymbol{u} \\ \omega\boldsymbol{u} \end{bmatrix} = \omega\begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{M} \end{bmatrix}\begin{bmatrix} \boldsymbol{u} \\ \omega\boldsymbol{u} \end{bmatrix}$$

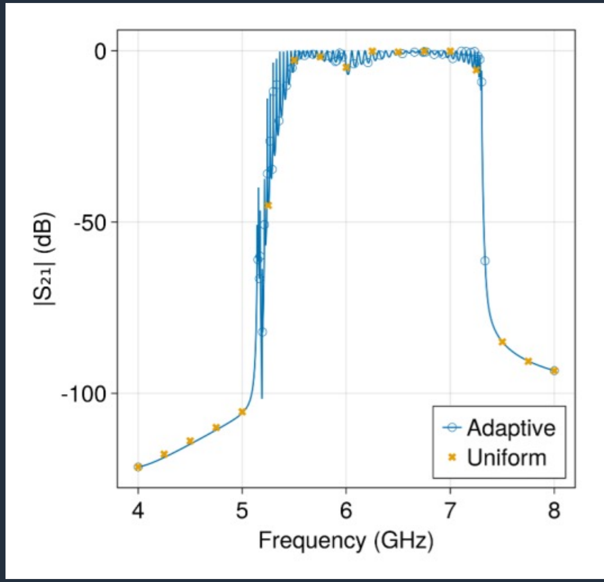
- Lumped ports, numeric wave ports, scattering parameter calculations
- Anisotropic and lossy material properties
- Absorbing boundary conditions

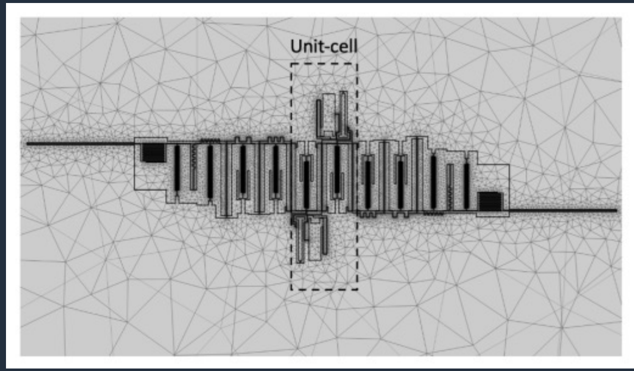
Large-scale cloud-based electromagnetics simulation

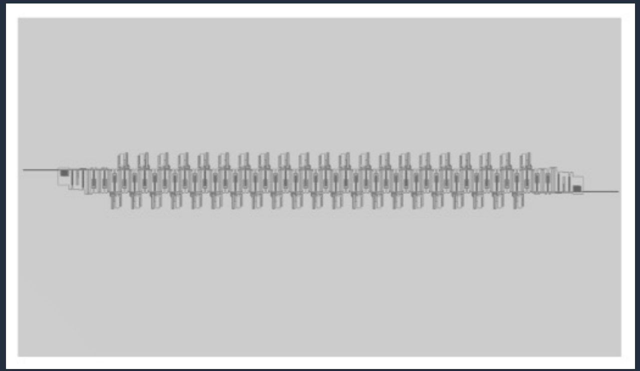
- 3D models ranging from 242.2 million to 1.4 billion degrees of freedom
- Adaptive fast frequency sweep simulations: Smallest in 45 min. (10 samples), largest in 4 hours (40 samples)
- Up to 12,800 AWS Graviton3E cores (hpc7g.16xlarge), EFA network



AWS Quantum Technologies Blog

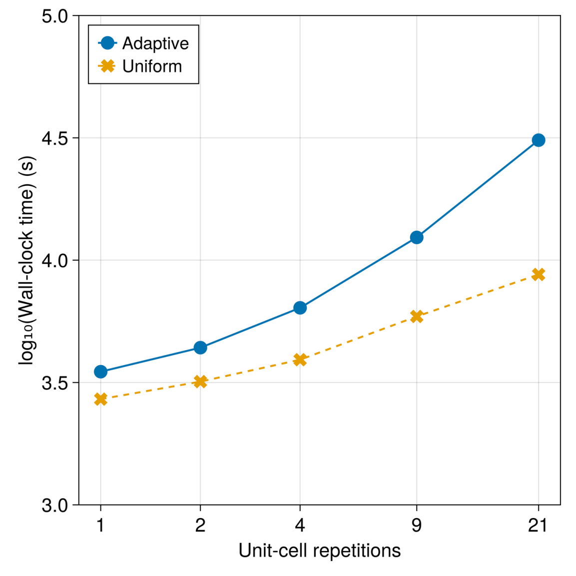




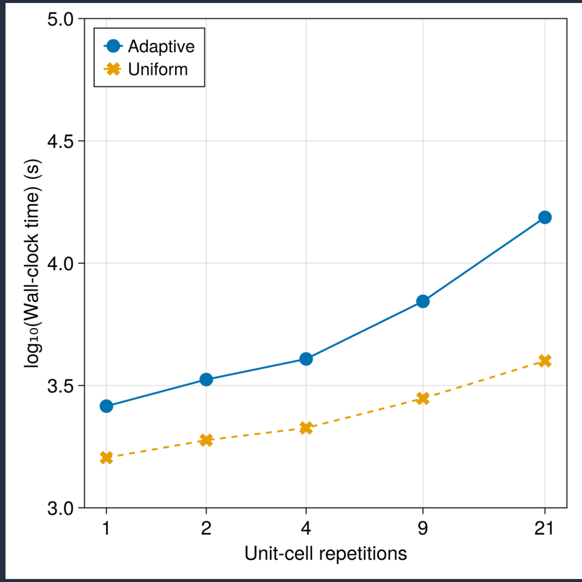


Large-scale cloud-based electromagnetics simulation

- 3D models ranging from 242.2 million to 1.4 billion degrees of freedom
- Adaptive fast frequency sweep simulations: Smallest in 45 min. (10 samples), largest in 4 hours (40 samples)
- Up to 12,800 AWS Graviton3E cores (hpc7g.16xlarge), EFA network



Feb. 2023
c6gn.16xlarge



Oct. 2023
1.69x – 2.19x improvement

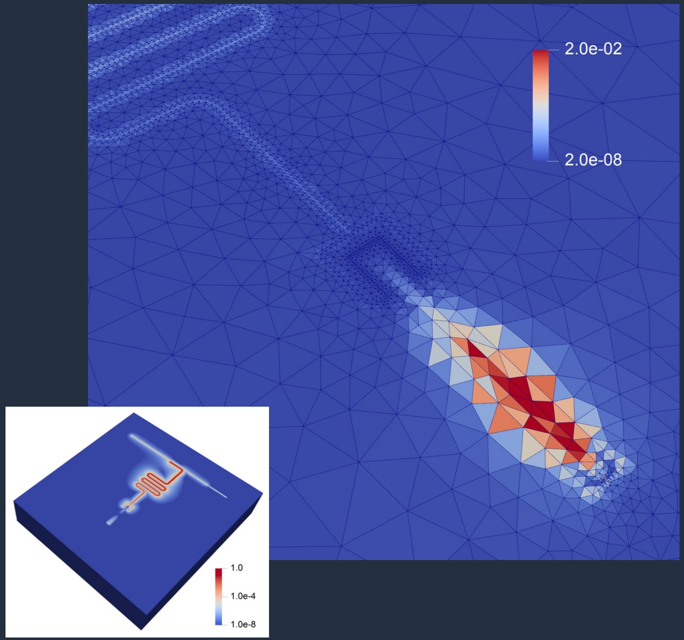
Adaptive mesh refinement

- Conforming refinement + rebalancing on tetrahedral meshes, general non-conforming refinement + coarsening + rebalancing for mixed meshes
- Non-conforming AMR for high-order Nédélec elements with triangular faces

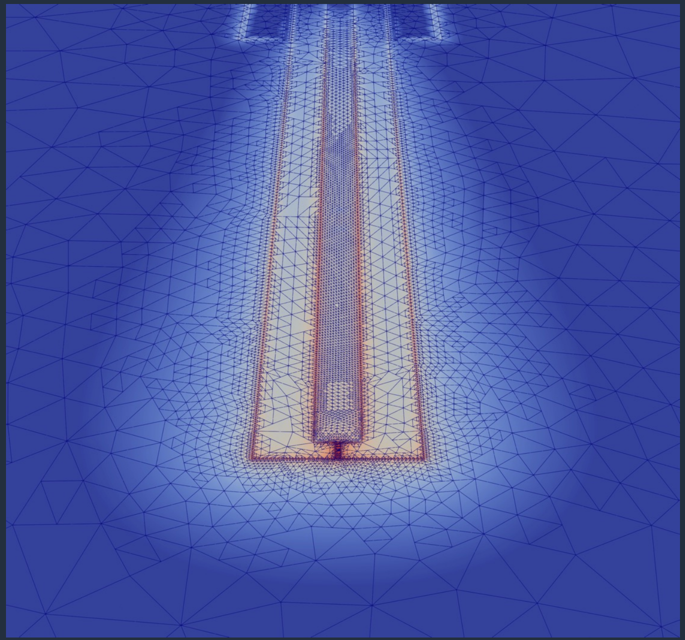
Adaptive mesh refinement

- Conforming refinement + rebalancing on tetrahedral meshes, general non-conforming refinement + coarsening + rebalancing for mixed meshes
- Non-conforming AMR for high-order Nédélec elements with triangular faces

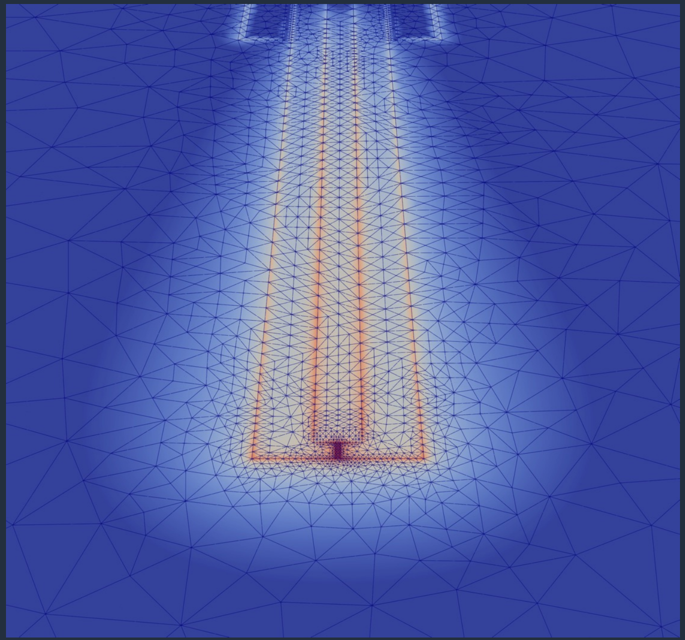
Indicator



Non-conforming refinement

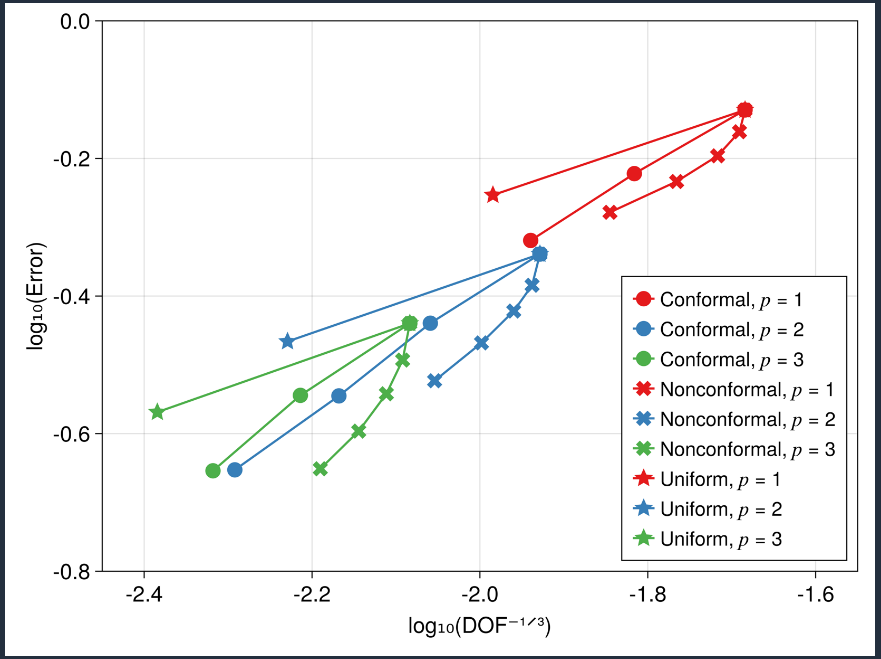


Conforming refinement



Adaptive mesh refinement

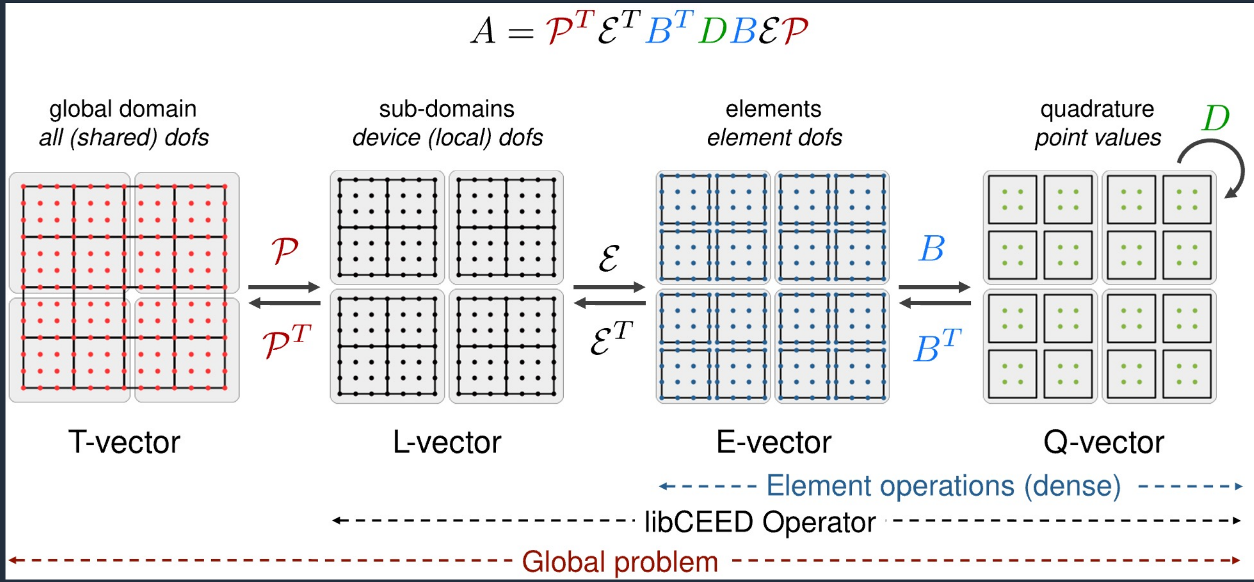
- Conforming refinement + rebalancing on tetrahedral meshes, general non-conforming refinement + coarsening + rebalancing for mixed meshes
- Non-conforming AMR for high-order Nédélec elements with triangular faces



Operator assembly and linear solvers

Operator assembly and linear solvers

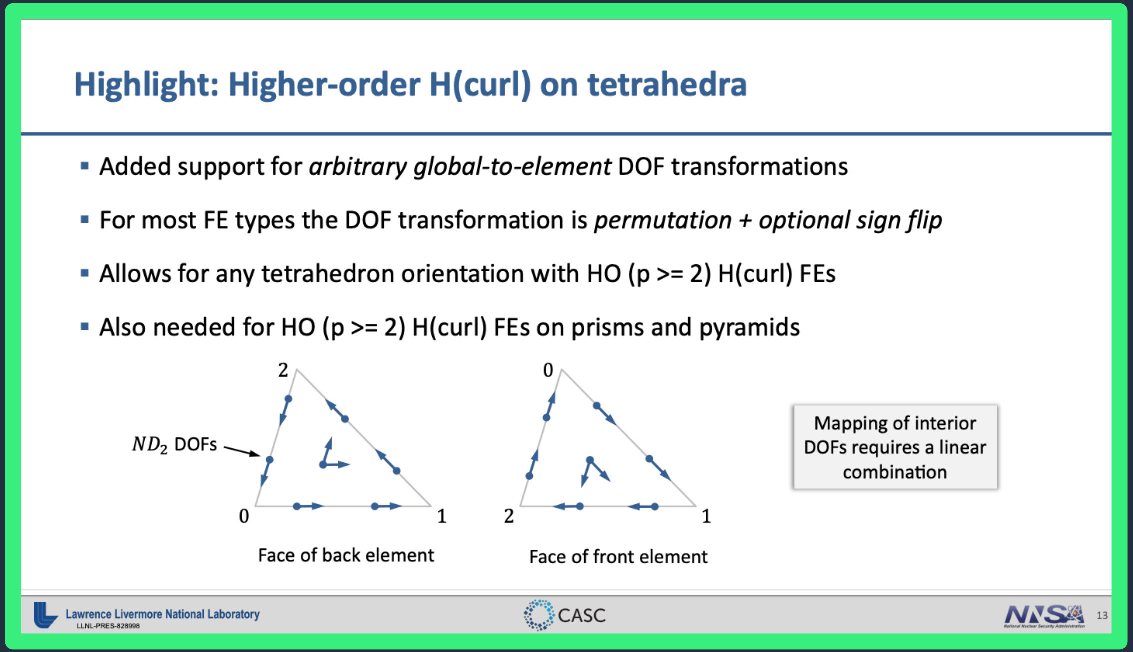
- Matrix-free *p*-multigrid preconditioning with partial assembly of high-order operators
- Partial assembly support: Simplex elements and mixed meshes, (mixed) operators on H^1-, *H(curl)*-, and *H(div)*-conforming spaces, boundary integrators, matrix-valued coefficients, all using libCEED



[libCEED Documentation]

Operator assembly and linear solvers

- Special element restriction operator $\mathcal{E}_{ND} = \mathcal{T}\mathcal{E}_{H^1}$ for high-order Nédélec elements with triangular faces



V. Dobrev's presentation at the [2021 MFEM Community Workshop]

Operator assembly and linear solvers

- Diagonally-scaled Chebyshev polynomial smoothers on each multigrid level only require assembly of operator diagonal

Operator assembly and linear solvers

- Diagonally-scaled Chebyshev polynomial smoothers on each multigrid level only require assembly of operator diagonal
- Auxiliary-space smoothing for non-static Maxwell problems (time and frequency domain):

$$\boldsymbol{B} = \boldsymbol{B}_{ND} + \boldsymbol{G}\boldsymbol{B}_{H^1}\boldsymbol{G}^T$$

(construct the operator $\boldsymbol{A}_{H^1} = \boldsymbol{G}^T\boldsymbol{A}_{ND}\boldsymbol{G}$ for $\boldsymbol{B}_{H^1}$ directly without a sparse matrix triple product)

Operator assembly and linear solvers

- Diagonally-scaled Chebyshev polynomial smoothers on each multigrid level only require assembly of operator diagonal
- Auxiliary-space smoothing for non-static Maxwell problems (time and frequency domain):

$$\boldsymbol{B} = \boldsymbol{B}_{ND} + \boldsymbol{G}\boldsymbol{B}_{H^1}\boldsymbol{G}^T$$

  (construct the operator $\boldsymbol{A}_{H^1} = \boldsymbol{G}^T\boldsymbol{A}_{ND}\boldsymbol{G}$ for $\boldsymbol{B}_{H^1}$ directly without a sparse matrix triple product)
- Coarse solve (p = 1, real-valued $\boldsymbol{A}_0 = \mathrm{Re}\{\boldsymbol{A}_0\} + \mathrm{Im}\{\boldsymbol{A}_0\}$):

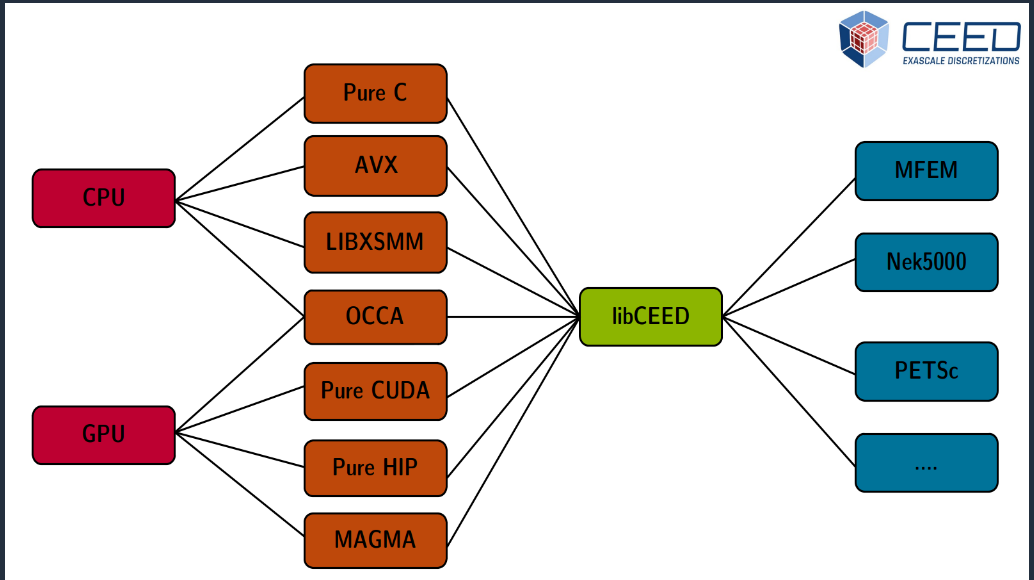
  Electrostatics: Standard AMG (Hypre)

  Magnetostatic and transient: Auxiliary-space Maxwell Solver (AMS)

  Driven and eigenmode (complex symmetric, indefinite): AMS with SPD preconditioner matrix, or parallel sparse-direct solve (SuperLU_DIST, STRUMPACK, MUMPS)

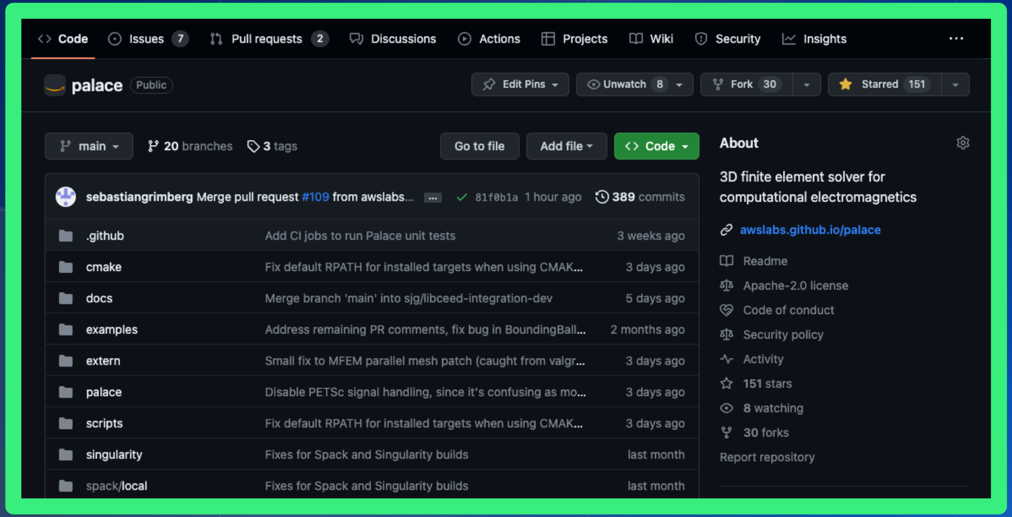
Backends and GPU support

- Runtime backend selection for performance portability
- Support for NVIDIA and AMD GPUs using MFEM and libCEED's CUDA and HIP backends
- *Work in progress, but expected to release in coming weeks*



[libCEED Documentation]

Thank you!



https://github.com/awslabs/palace

Sebastian Grimberg
sjg@amazon.com

Hugh Carson
hughcars@amazon.com

Andrew Keller
kllrak@amazon.com