Science and Technology Facilities Council

Hartree Centre

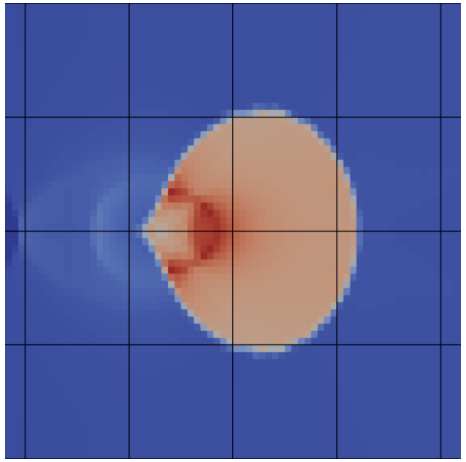
High-order FEM implementation in AMReX with PETSc

Alex Grant, Olha Ivanyshyn Yaman, Karthik Chockalingam

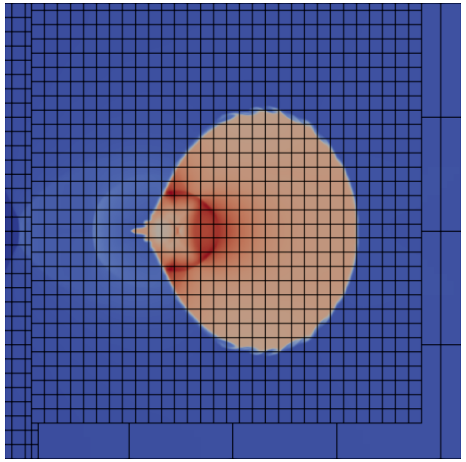
Motivation for FEM in AMReX

Many possible applications for FEM, but primary driver is Fusion Computing Lab collaboration with UKAEA

- UKAEA want to explore FEM for plasma edge modelling, however most FEM packages use unstructured grids
- Structured grids (like AMReX) should have some advantages in scaling and performance
- Want to use particles (PIC) too – difficult to keep track of on unstructured grids. AMReX has native particle support.
- One drawback is support for complex geometry – AMReX has Embedded Boundary (cut cell methods) for this.



Level 0

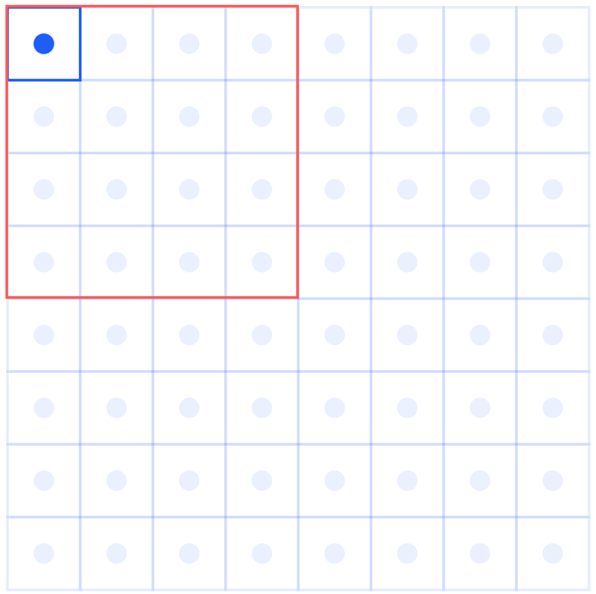


Level 1 (8x refinement)

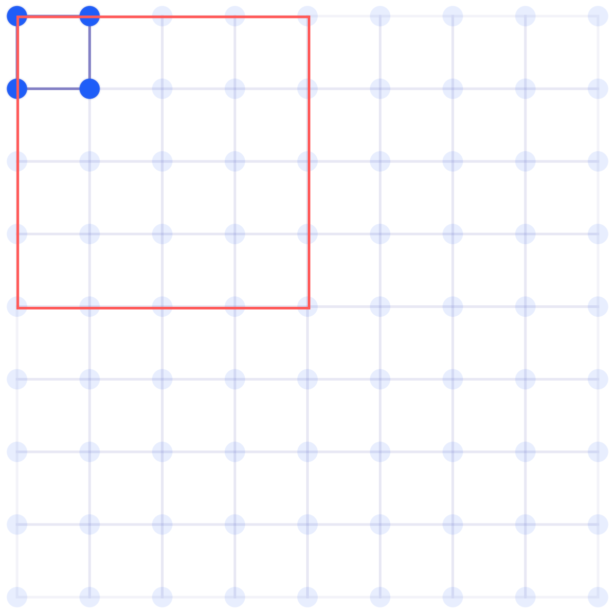
Science and Technology Facilities Council

Hartree Centre

AMReX data structures



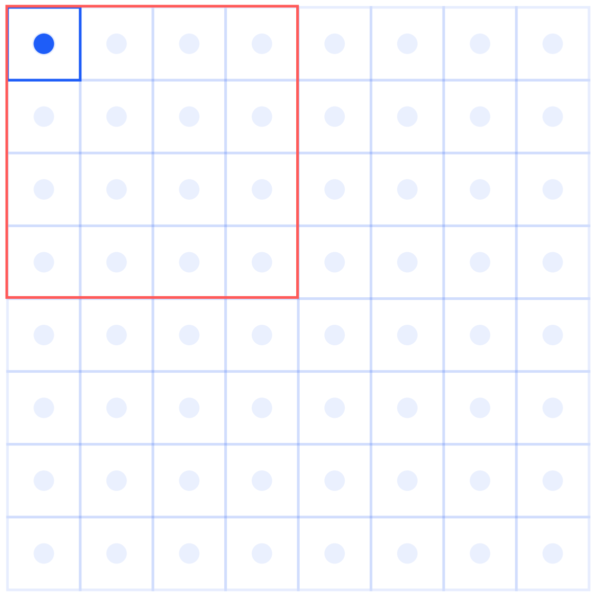
Cell-centered data



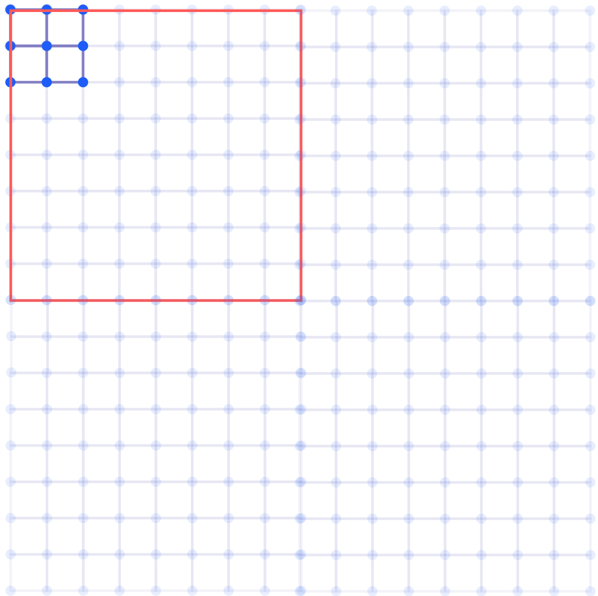
Nodal data (first order)

- For FEM, we use cell-centered boxes to iterate over elements, nodal boxes to store data, interpolate between levels, and indexing
- For higher-order elements, can map cell-centered boxes to higher resolution nodal data (up to the implementation, not AMReX)

AMReX data structures



Cell-centered data



Nodal data (second order)

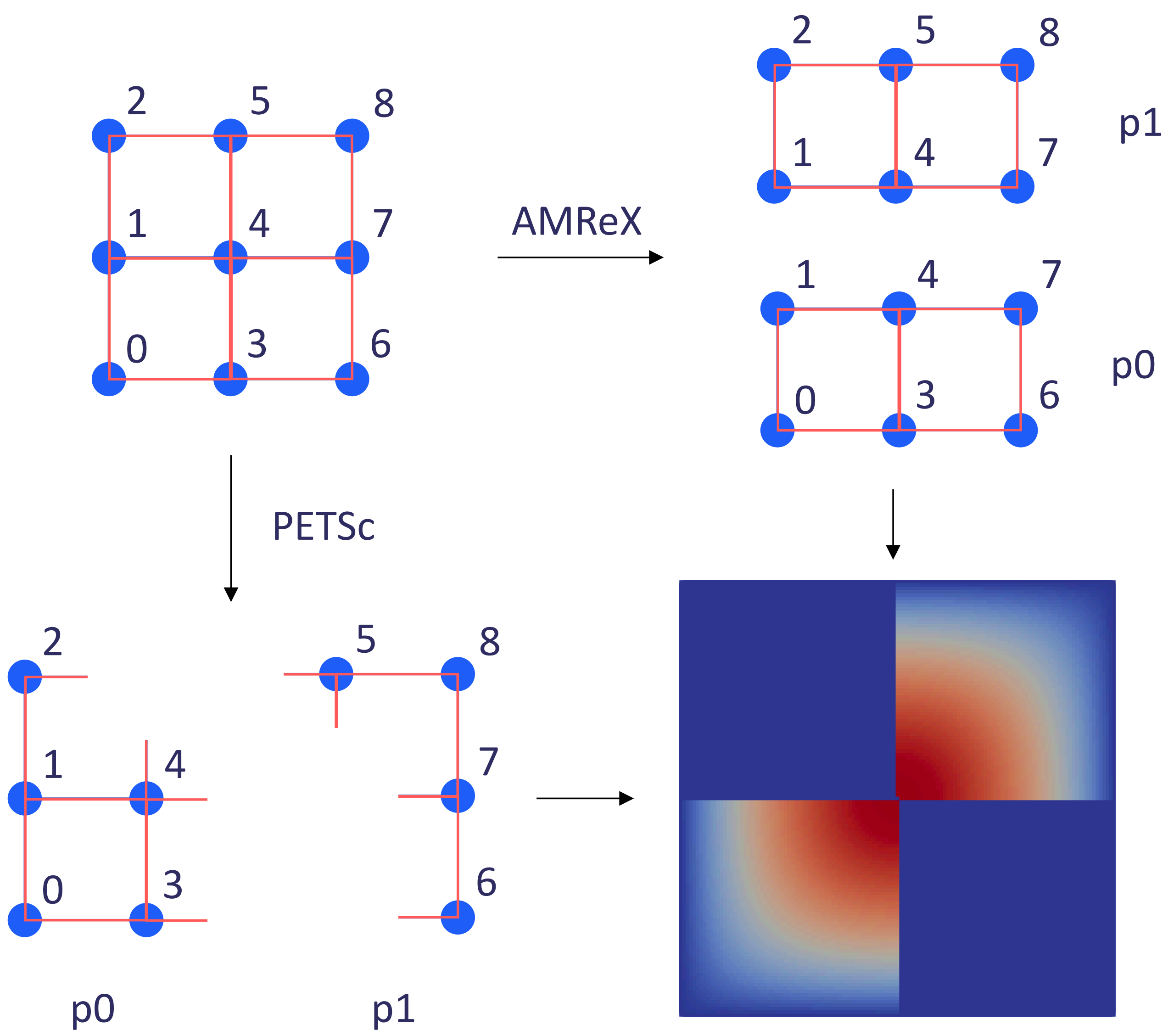
- For FEM, we use cell-centered boxes to iterate over elements, nodal boxes to store data, interpolate between levels, and indexing
- For higher-order elements, can map cell-centered boxes to higher resolution nodal data (up to the implementation, not AMReX)

Science and Technology Facilities Council

Hartree Centre

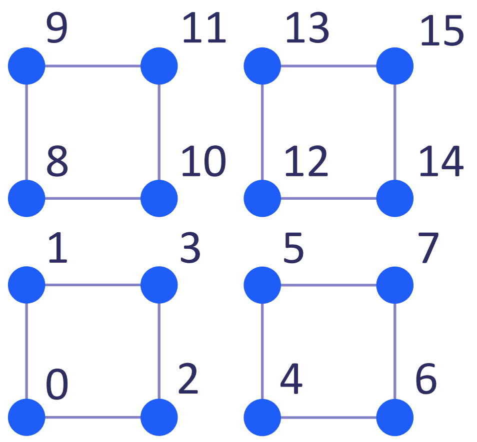
PETSc - indexing

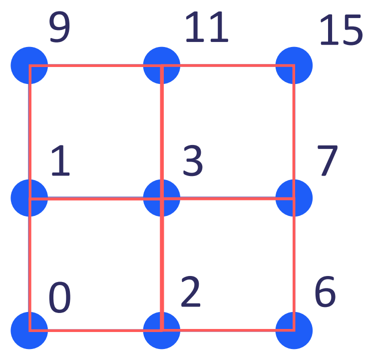
- Simplest way to generate a global node index for PETSc is to use the nodal position, i.e. global_id = i * n_cells_i + j
  - Problematic at higher levels
  - PETSc domain decomposition wants ID’s in order within each MPI process, leading to a mismatch
- Box-based indexing scheme solves both these issues
  - Global ID based on node position within a box, with duplicates pinned
  - Can use AMReX for MPI communication



PETSc - indexing

- Simplest way to generate a global node index for PETSc is to use the nodal position, i.e. global_id = i * n_cells_i + j
  - Problematic at higher levels
  - PETSc domain decomposition wants ID's in order within each MPI process, leading to a mismatch
- Box-based indexing scheme solves both these issues
  - Global ID based on node position within a box, with duplicates pinned
  - Can use AMReX for MPI communication





0, 1, 2, 3

~~4, 5~~, 6, 7

~~8~~, 9,~~10~~,11

~~12,13,14~~,15

```
auto mask = amrex::OwnerMask(mf_node[level],
model_geom[level].periodicity());

mf_node[level].OverrideSync(*mask,
model_geom[level].periodicity());
```

Science and Technology Facilities Council

Hartree Centre

PETSc – block matricies

- Moving from simple poisson problem to multi-component cases, expected to have to rewrite code to avoid copy-paste
- Used PETSc block matrix type instead
  - Each element of the block matrix is an n x n submatrix, where n is the blocksize
- Note – not all solvers support this

- Non-linear Newton solve

$$
\begin{aligned}
\mathbf{J}\left(\vec{u}_n\right)\delta\vec{u}_{n+1} &= -\vec{R}\left(\vec{u}_n\right), \\
\vec{u}_{n+1} &= \vec{u}_n + \delta\vec{u}_{n+1},
\end{aligned}
$$

$$
\begin{bmatrix}
\frac{\partial R^I_{u_1}}{\partial u^I_1} & \cdots & \frac{\partial R^I_{u_1}}{\partial u^I_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial R^I_{u_N}}{\partial u^I_1} & \cdots & \frac{\partial R^I_{u_N}}{\partial u^I_N}
\end{bmatrix}
\begin{bmatrix}
\delta u^J_1 \\ \vdots \\ \delta u^J_N
\end{bmatrix}
= -
\begin{bmatrix}
R^I_{u_1} \\ \vdots \\ R^I_{u_N}
\end{bmatrix},
$$

```
MatCreate(PETSC_COMM_WORLD, &A[level]);
MatSetType(A[level], MATMPIBAIJ);
MatSetSizes(A[level], locN*blocksize, …
MatSetBlockSize(A[level], blocksize);
```

Blob2D – fully coupled, fully implicit

- Implemented the 'Blob2D' fusion test case

$$\frac{\partial n_e}{\partial t} = -\nabla\cdot(n_e \mathbf{v}_{E\times B}) + \nabla\cdot\frac{1}{e}\mathbf{j}_{sh}$$

$$p_e = e n_e T_e$$

$$\frac{\partial \omega}{\partial t} = -\nabla\cdot(\omega \mathbf{v}_{E\times B}) + \nabla\left(p_e \nabla\times\frac{\mathbf{b}}{B}\right) + \nabla\cdot\mathbf{j}_{sh}$$

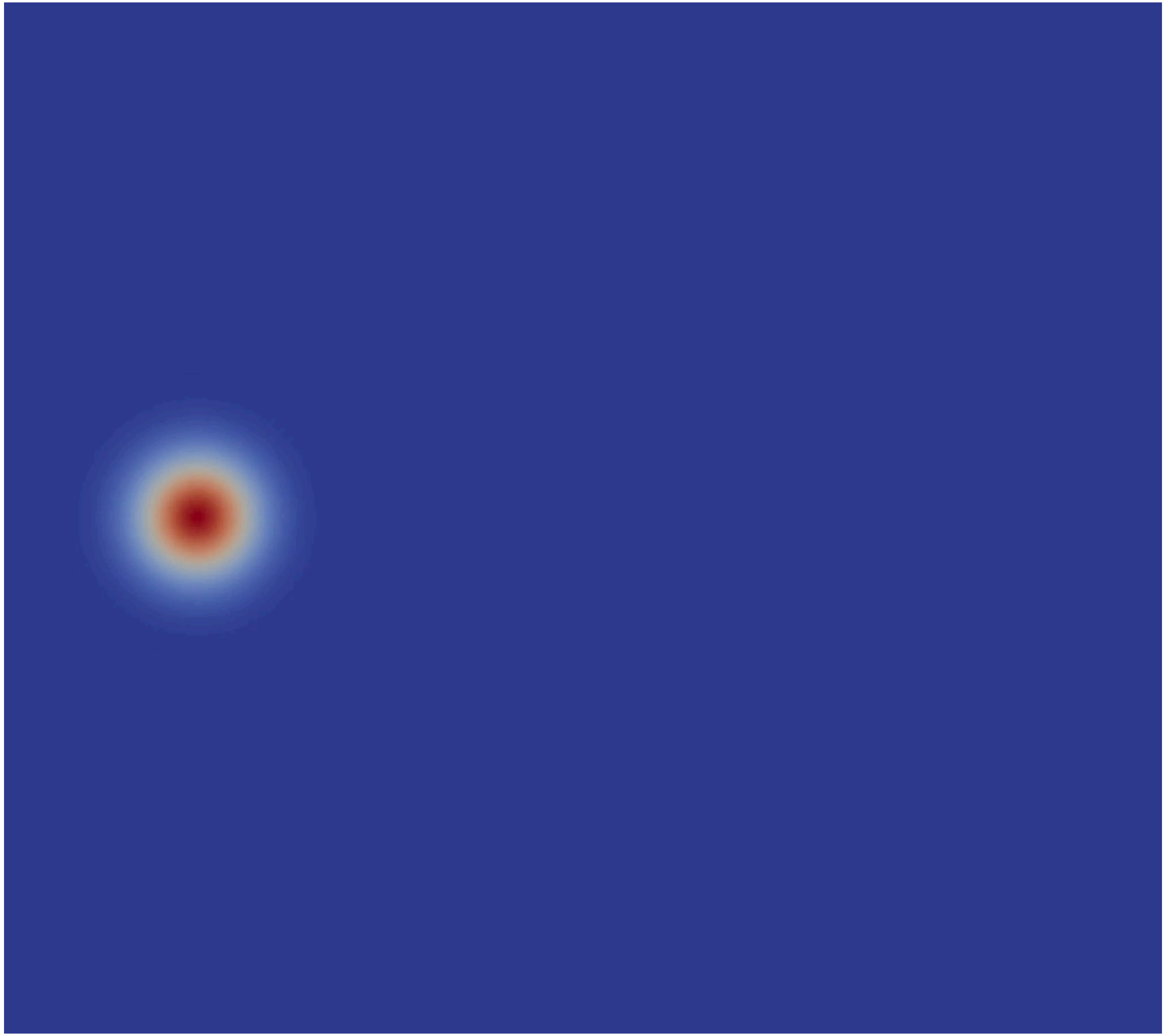
$$\nabla\cdot\left(\frac{1}{B^2}\nabla_\perp\phi\right) = \omega$$

$$\nabla\cdot\mathbf{j}_{sh} = \frac{n_e\phi}{L_{||}}$$

Solve for the electron density, vorticity, connection length

- Currently, only implemented on the base level, with no refinement

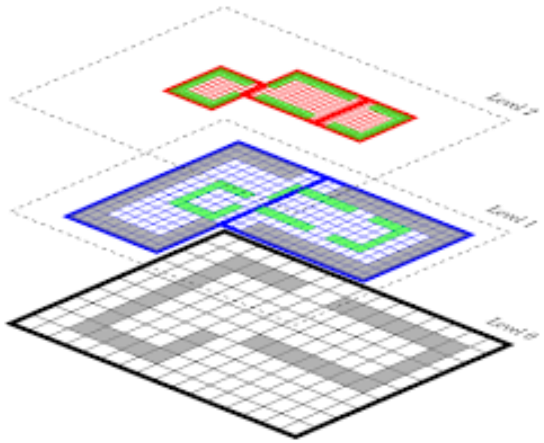
Blob2D



Science and Technology Facilities Council
Hartree Centre

Multi-level hp-adaptivity with arbitrary hanging nodes

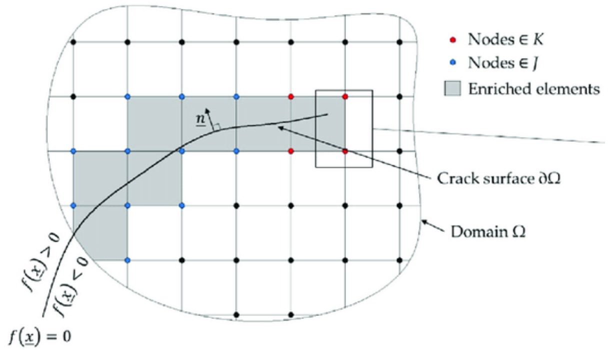
AMReX

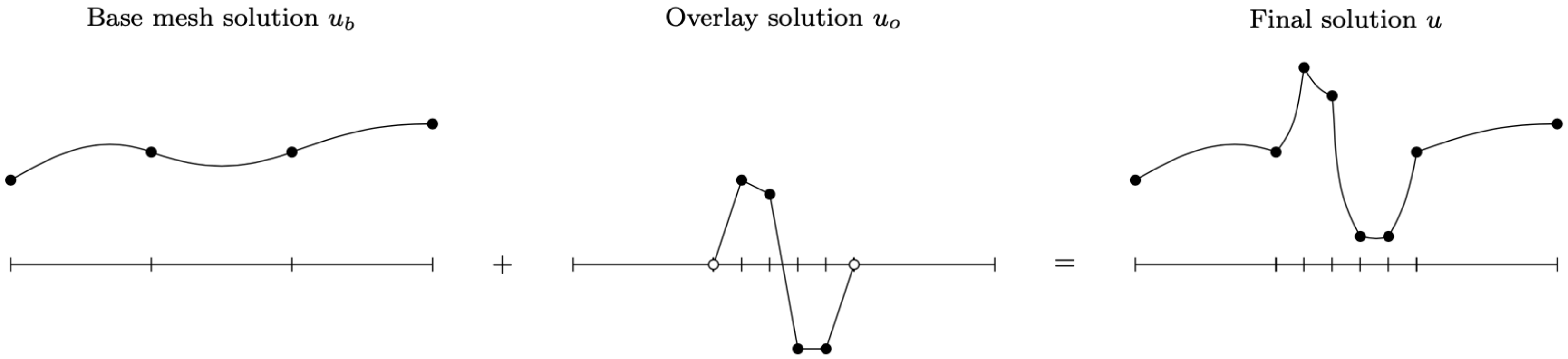


AMReX solves by levels, which is not the case in most convectional FE codes

Principle of superposition is widely used to model fracture using XFEM

$$u(\boldsymbol{x}) = \sum_i N_i(\boldsymbol{x})\hat{u}_i + \sum_i N_i(\boldsymbol{x})\psi(\boldsymbol{x})\hat{a}_i.$$

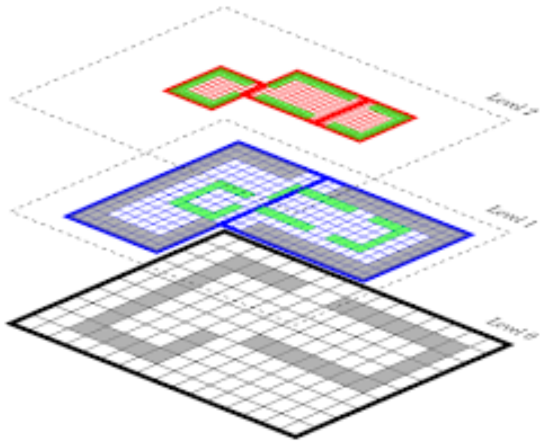




Reference
Multi-level hp-FEM: dynamically changing high-order mesh refinement with arbitrary hanging nodes – Nils Dietrisch Zander (2016)

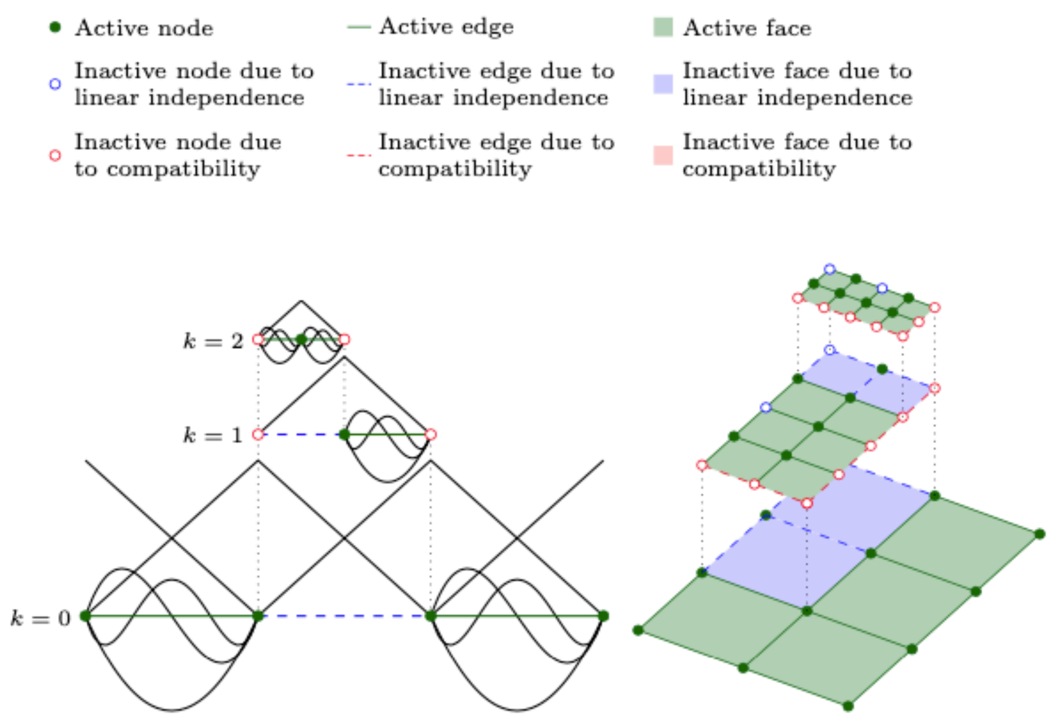
Multi-level hp-adaptivity with arbitrary hanging nodes

AMReX



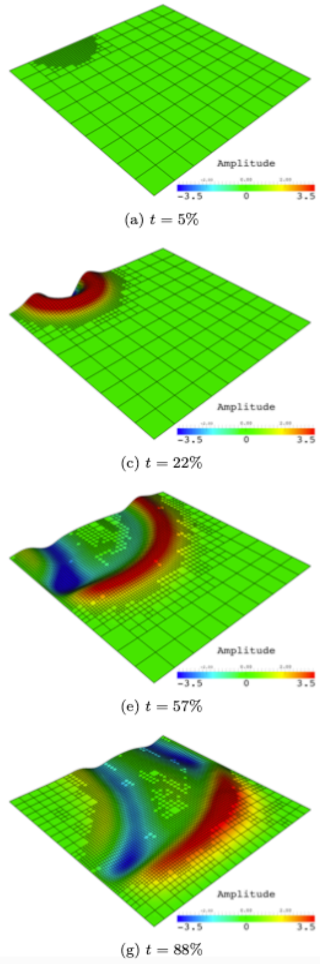
AMReX solves by levels, which is not the case in most convectional FE codes

Reference
Multi-level hp-FEM: dynamically changing high-order mesh refinement with arbitrary hanging nodes – Nils Dietrisch Zander (2016)



(a) One-dimensional case (b) Two-dimensional case

(1) In refine-by-superposition approach, the global continuity is ensured by applying homogeneous Dirichlet boundary conditions on the overlay solution.
(2) The second important aspect to be considered is that the overlay mesh must not introduce a linear dependence between the shape functions of the two refinement levels.

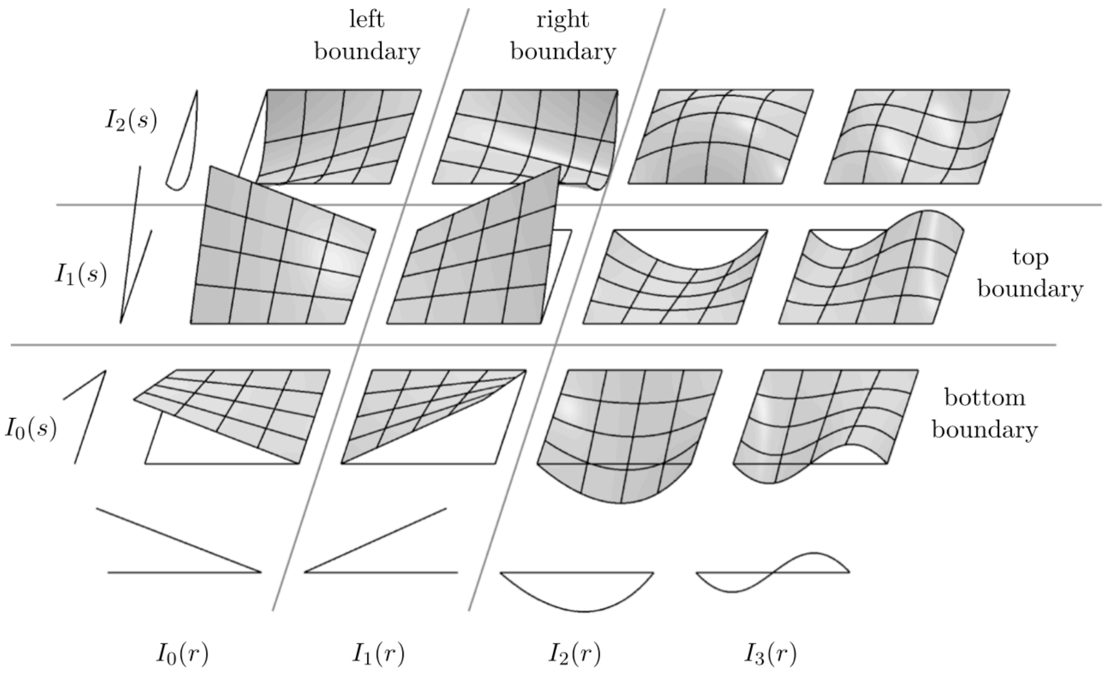


Demonstrated for wave propagation Zander (2016)

Science and Technology Facilities Council
Hartree Centre

High-order Hierarchical basis functions using Integrated Legendre polynomials

$$u(\boldsymbol{x}) = \sum_i N_i(\boldsymbol{x})\hat{u}_i + \sum_i N_i(\boldsymbol{x})\psi(\boldsymbol{x})\hat{a}_i.$$



Legendre polynomials (Spectral elements)

$$L_0(r) = 1$$
$$L_1(r) = r$$
$$L_i(r) = \frac{1}{n}\left[(2i-1)rL_{i-1}(r) - (i-1)L_{i-2}(r)\right] \quad i = 2, 3, \ldots, p,$$

$$\int_{-1}^{1} L_i L_j \ \mathrm{d}r = \begin{cases} \frac{2}{2i+1}, & \text{if } i = j \\ 0, & \text{else} \end{cases}$$

Integrated Legendre polynomials (Hierarchical elements)

$$P_i(r) = \sqrt{\frac{2i-1}{2}} \int_{-1}^{r} L_{i-1}(t)dt = \frac{1}{\sqrt{4i-2}}\left(L_i(r) - L_{i-2}(r)\right) \quad i = 2, 3, \ldots$$

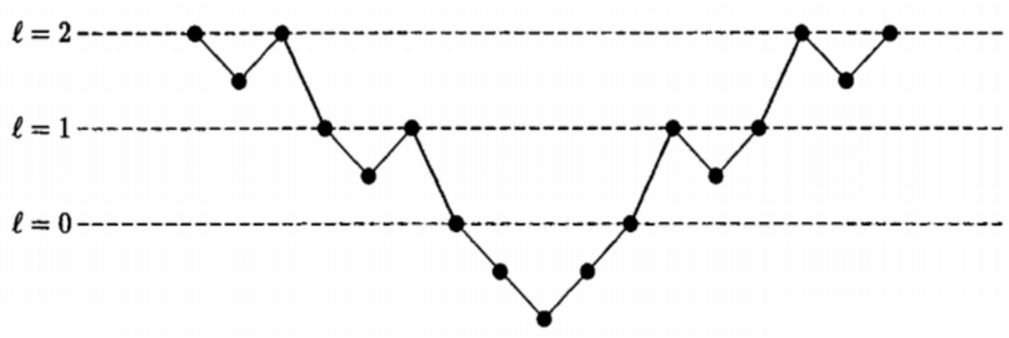
$$\int_{-1}^{1} P_i'(x)P_j'(x)dx = 0, \quad \text{if} \quad i \neq j$$

Hierarchical basis functions can be extended to solve electromagnetic problems
(High Order Finite Element Methods for Electromagnetic Field Computation, Zaglmayr, 2006)

Science and Technology Facilities Council
Hartree Centre

Next steps

- Main aim for this year is expected to be replacing the external solvers (PETSc -> Hypre/MUMPS) with the internal AMReX MLMG solver, or Hypre directly, for improved performance, as well as portability to GPUs.
- (Arbitrary) higher order implementation
- Test case with adaptive mesh refinement



MLMG reference
A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier–Stokes Equations
- Almgren et al. (1998)

Science and Technology Facilities Council

Hartree Centre

The Team at Hartree:

Karthik Chockalingam, Alex Grant, Olha Ivanyshyn Yaman

External Collaborators:

UKAEA – FARSCAPE/NEPTUNE

Brandon Runnels, Professor, Iowa State

Thank you

hartree.stfc.ac.uk @HartreeCentre STFC Hartree Centre hartree@stfc.ac.uk