Parallel octree-based adaptive finite elements for large-scale geoscience applications

Carsten Burstedde, Omar Ghattas
Georg Stadler, Tiankai Tu, Lucas Wilcox

Institute for Computational Engineering and Sciences
The University of Texas at Austin

Collaboration with George Biros, Michael Gurnis, and Shijie Zhong
under NSF PetaApps Program (OCI-0749334)

October 2007

Outline

1. Parallel octree-based dynamic adaptive mesh refinement—definitions and challenges
2. Our scope and other approches
3. Our approach to adaptive FEM based on parallel octrees
4. Driving application: mantle convection

Outline

1. **Parallel octree-based dynamic adaptive mesh refinement—definitions and challenges**
2. Our scope and other approches
3. Our approach to adaptive FEM based on parallel octrees
4. Driving application: mantle convection

Parallel octree-based dynamic adaptive mesh refinement

Adaptive mesh refinement

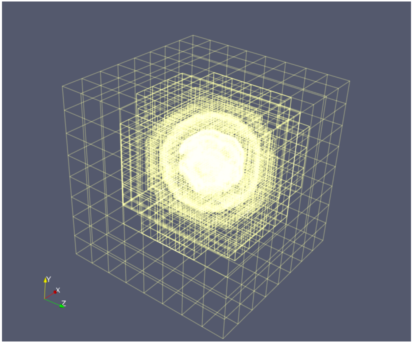
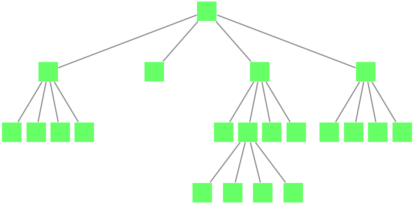


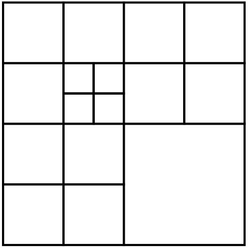
Figure: Adaptively refined mesh

- AMR essential for resolving physical phenomena that vary over a wide range of scales
- we estimate a factor of $\sim 10^2 - 10^3$ savings in the number of dofs for our application

Parallel octree-based dynamic adaptive mesh refinement

Octree-based meshing

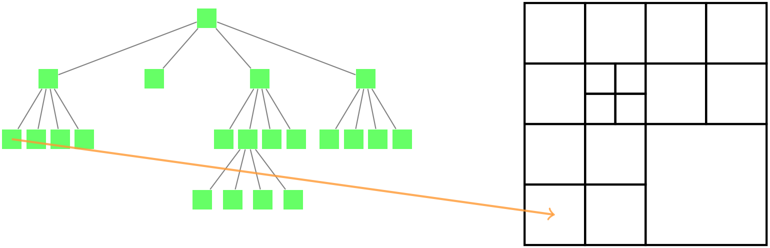




- The mesh is based on an octree: nodes in the tree map to hexahedral elements
- Octrees allow simple treatment of connectivity information
- Octree-based meshes are a tradeoff between geometric flexibility and simplicity

Parallel octree-based dynamic adaptive mesh refinement

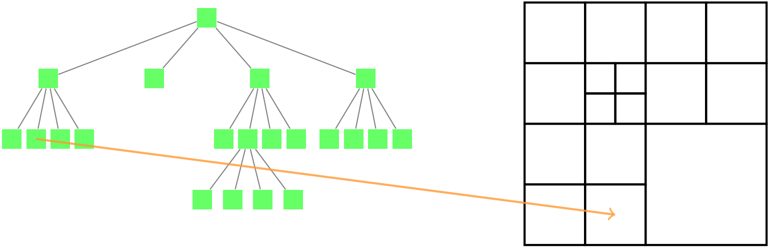
Octree-based meshing



- The mesh is based on an octree: nodes in the tree map to hexahedral elements
- Octrees allow simple treatment of connectivity information
- Octree-based meshes are a tradeoff between geometric flexibility and simplicity

Parallel octree-based dynamic adaptive mesh refinement

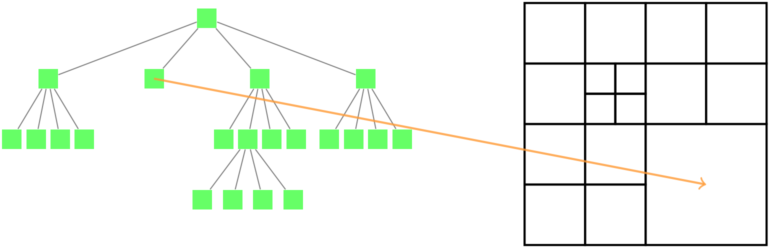
Octree-based meshing



- The mesh is based on an octree: nodes in the tree map to hexahedral elements
- Octrees allow simple treatment of connectivity information
- Octree-based meshes are a tradeoff between geometric flexibility and simplicity

Parallel octree-based dynamic adaptive mesh refinement

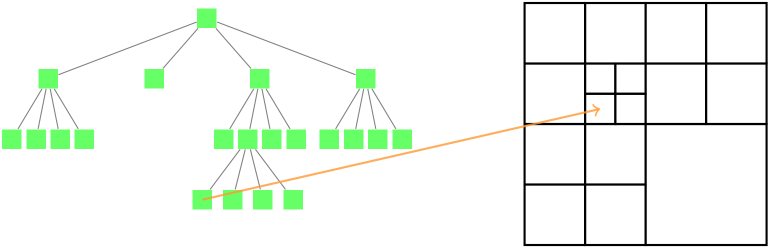
Octree-based meshing



- The mesh is based on an octree: nodes in the tree map to hexahedral elements
- Octrees allow simple treatment of connectivity information
- Octree-based meshes are a tradeoff between geometric flexibility and simplicity

Parallel octree-based dynamic adaptive mesh refinement

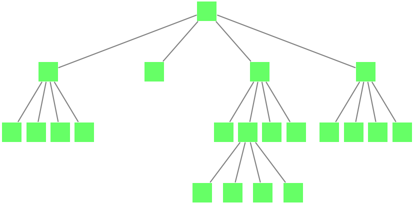
Octree-based meshing

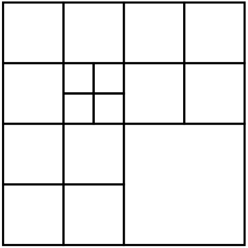


- The mesh is based on an octree: nodes in the tree map to hexahedral elements
- Octrees allow simple treatment of connectivity information
- Octree-based meshes are a tradeoff between geometric flexibility and simplicity

Parallel octree-based dynamic adaptive mesh refinement

Parallel implementation

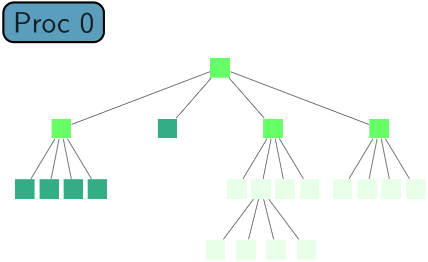


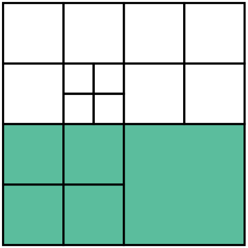


- Octree partitioned among processors
- Underlying mesh operations parallelized according to the octree partitioning

Parallel octree-based dynamic adaptive mesh refinement

Parallel implementation

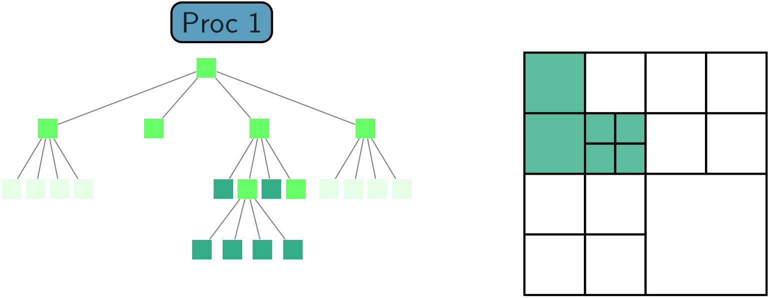




- Octree partitioned among processors
- Underlying mesh operations parallelized according to the octree partitioning

Parallel octree-based dynamic adaptive mesh refinement

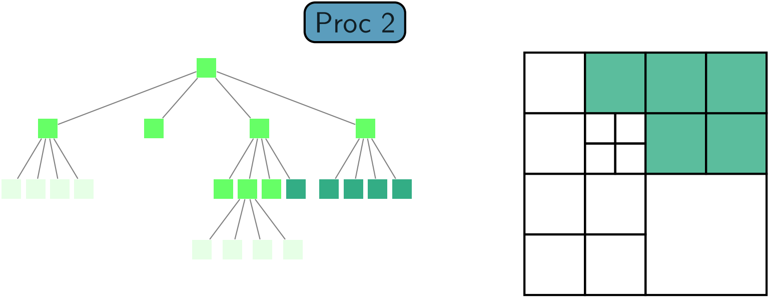
Parallel implementation



- Octree partitioned among processors
- Underlying mesh operations parallelized according to the octree partitioning

Parallel octree-based dynamic adaptive mesh refinement

Parallel implementation

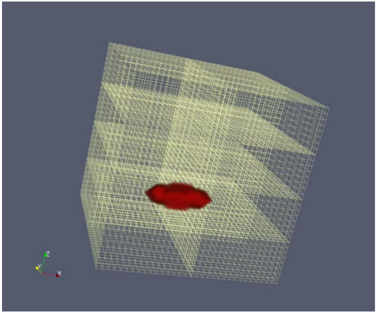


- Octree partitioned among processors
- Underlying mesh operations parallelized according to the octree partitioning

Parallel octree-based dynamic adaptive mesh refinement

Dynamic mesh adaptation

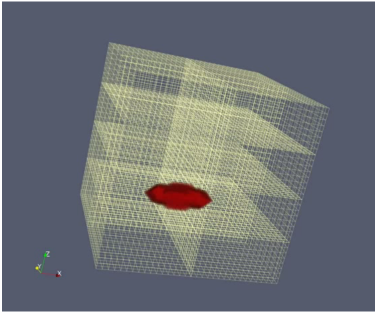
- For time-dependent problems the mesh needs to be adapted dynamically
- Mesh adaptation needs to run simultaneously with application



Parallel octree-based dynamic adaptive mesh refinement

Dynamic mesh adaptation

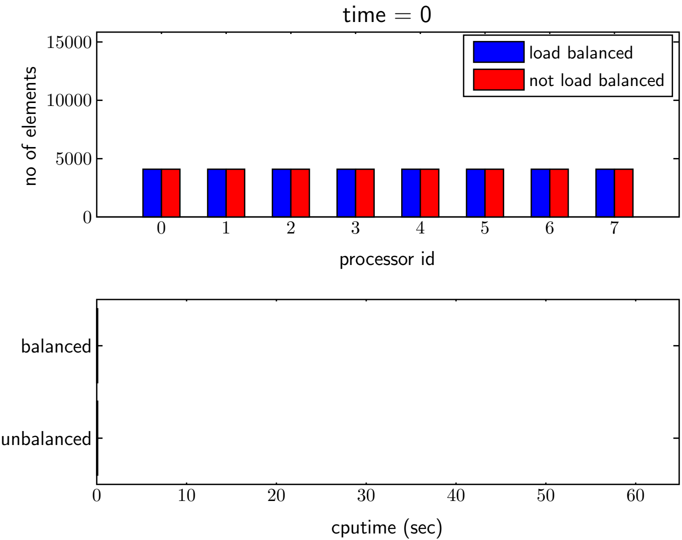
- For time-dependent problems the mesh needs to be adapted dynamically
- Mesh adaptation needs to run simultaneously with application



⇒ Problem: Mesh and application data need to be redistributed among processors ⇒ **dynamic load balancing**

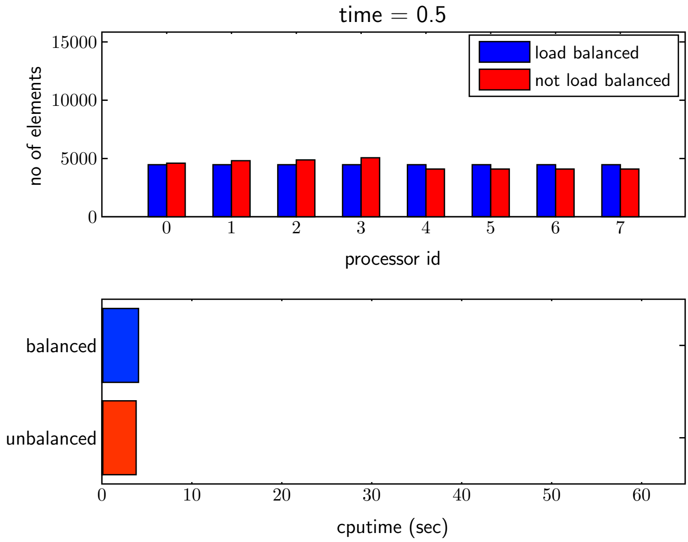
Parallel octree-based dynamic adaptive mesh refinement

Load balancing



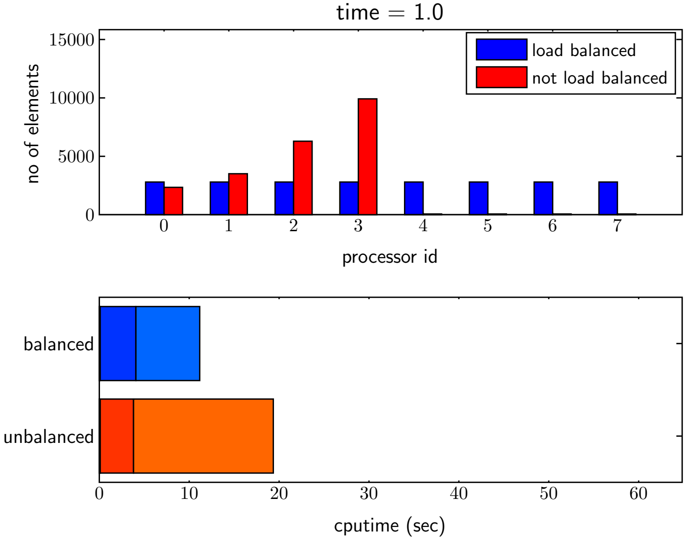
Parallel octree-based dynamic adaptive mesh refinement

Load balancing



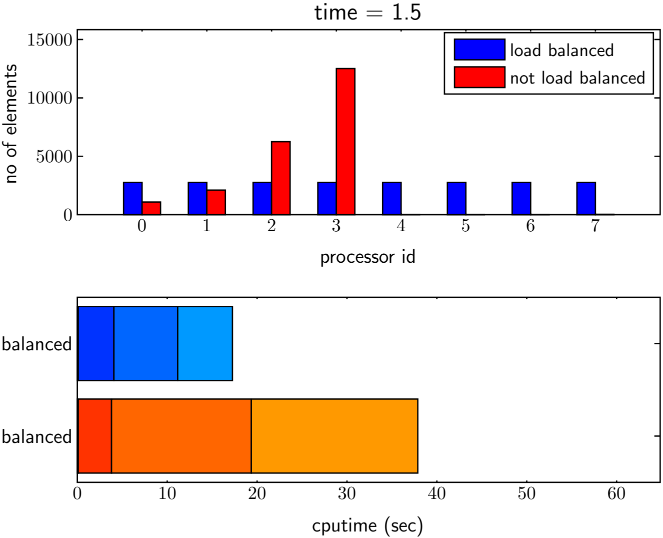
Parallel octree-based dynamic adaptive mesh refinement

Load balancing



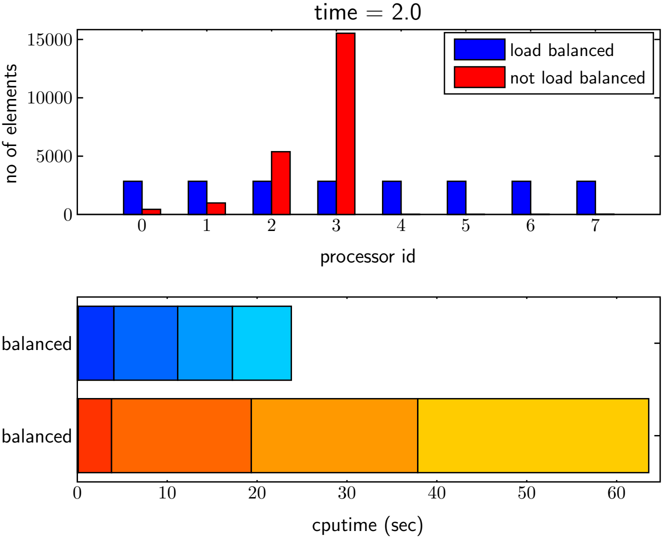
Parallel octree-based dynamic adaptive mesh refinement

Load balancing



Parallel octree-based dynamic adaptive mesh refinement

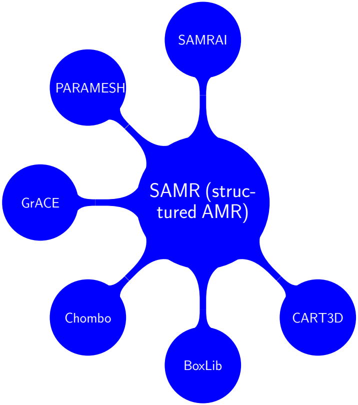
Load balancing



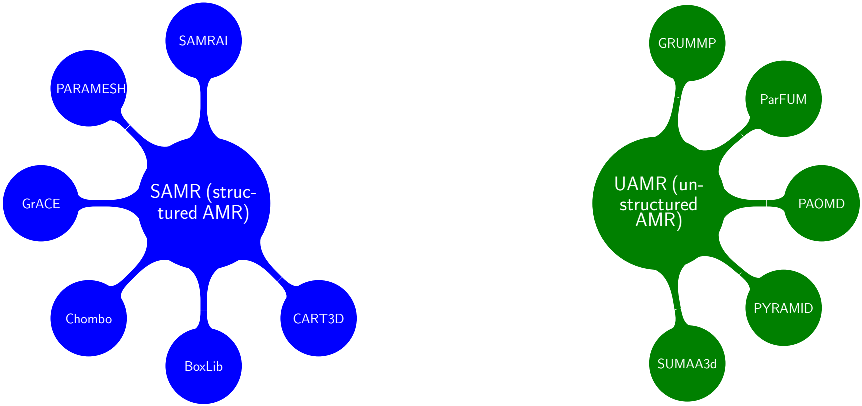
Overview

1. Parallel octree-based dynamic adaptive mesh refinement—definitions and challenges
2. **Our scope and other approches**
3. Our approach to adaptive FEM based on parallel octrees
4. Driving application: mantle convection

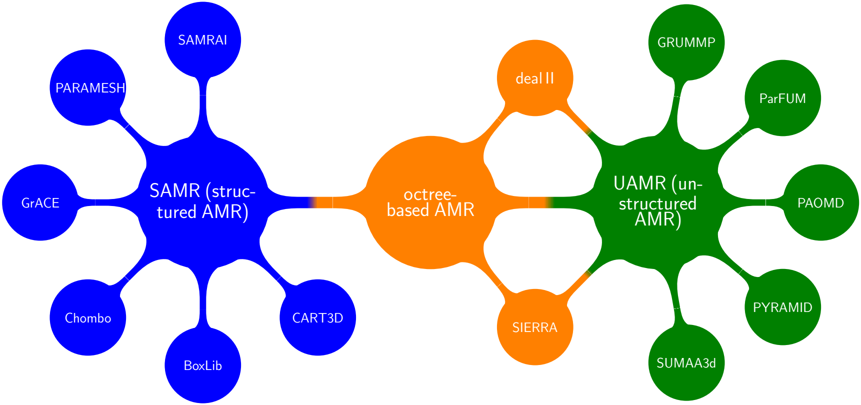
Approaches to adaptive mesh refinement in parallel



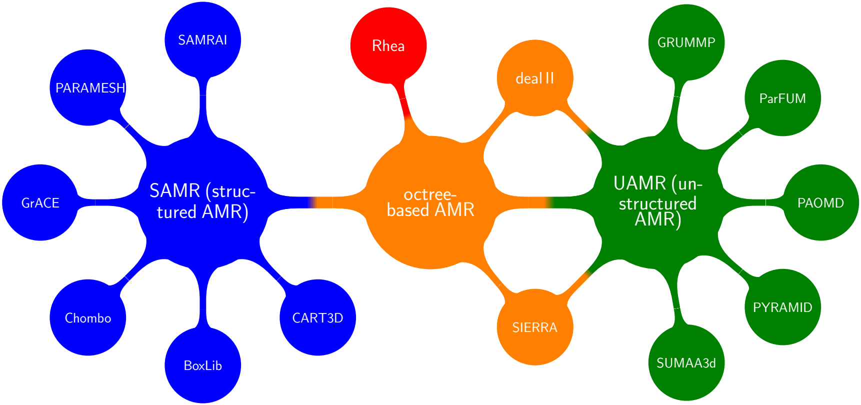
Approaches to adaptive mesh refinement in parallel



Approaches to adaptive mesh refinement in parallel



Approaches to adaptive mesh refinement in parallel



Our scope

We are interested in

- adaptive mesh refinement/coarsening for finite elements that **scales to large** (i.e., > 1K) **numbers of processors**

Our scope

We are interested in

- adaptive mesh refinement/coarsening for finite elements that **scales to large** (i.e., > 1K) **numbers of processors**
- an **end-to-end approach**, i.e., all components (meshing, solver, analysis and visualization) run in parallel and tightly coupled on the same machine

Our scope

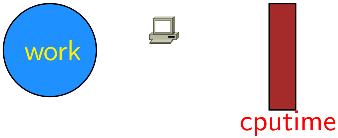
We are interested in

- adaptive mesh refinement/coarsening for finite elements that **scales to large** (i.e., > 1K) **numbers of processors**
- an **end-to-end approach**, i.e., all components (meshing, solver, analysis and visualization) run in parallel and tightly coupled on the same machine
- very large **applications in geosciences**

Scalings

Weak and strong scaling

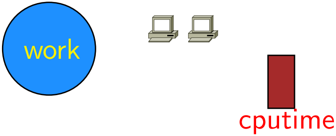
Strong scaling: Keep workload constant and increase the number of processors



Scalings

Weak and strong scaling

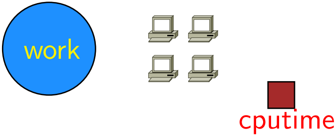
Strong scaling: Keep workload constant and increase the number of processors



Scalings

Weak and strong scaling

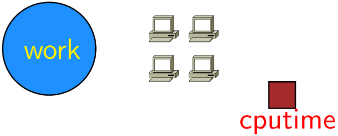
Strong scaling: Keep workload constant and increase the number of processors

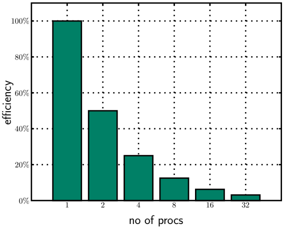


Scalings

Weak and strong scaling

Strong scaling: Keep workload constant and increase the number of processors

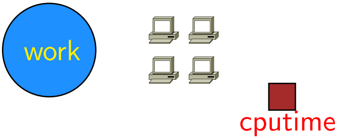


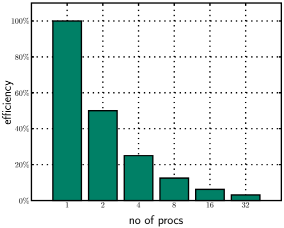


Scalings

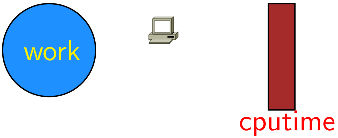
Weak and strong scaling

Strong scaling: Keep workload constant and increase the number of processors





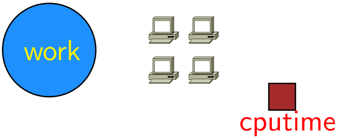
Weak scaling: Increase problem size and number of processors simultaneously

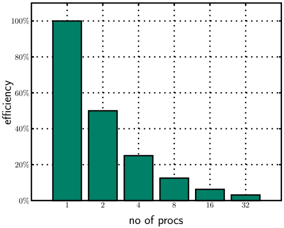


Scalings

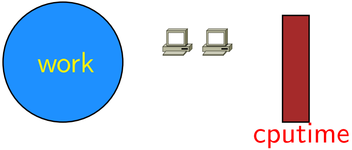
Weak and strong scaling

Strong scaling: Keep workload constant and increase the number of processors





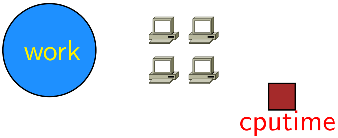
Weak scaling: Increase problem size and number of processors simultaneously

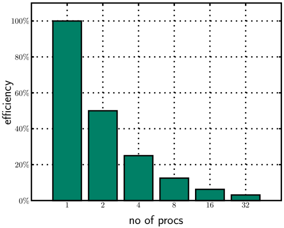


Scalings

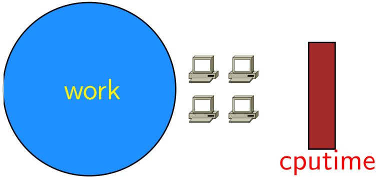
Weak and strong scaling

Strong scaling: Keep workload constant and increase the number of processors





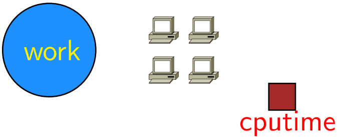
Weak scaling: Increase problem size and number of processors simultaneously

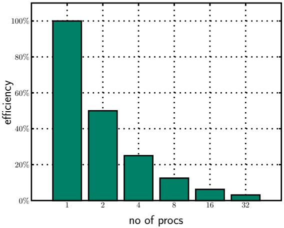


Scalings

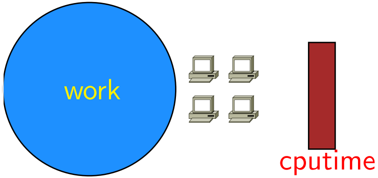
Weak and strong scaling

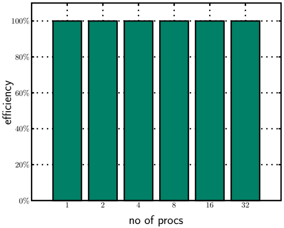
Strong scaling: Keep workload constant and increase the number of processors





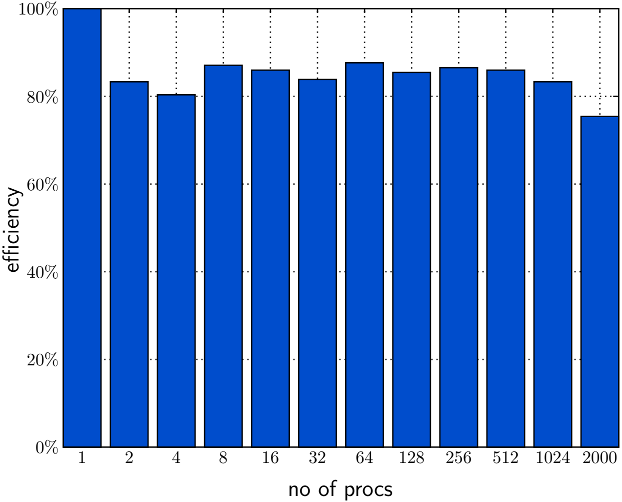
Weak scaling: Increase problem size and number of processors simultaneously





Scalings

Results for Rhea: weak scalings for advection-diffusion equation with adaptive mesh refinement/coarsening on up to 2000 processors: efficiency per time step

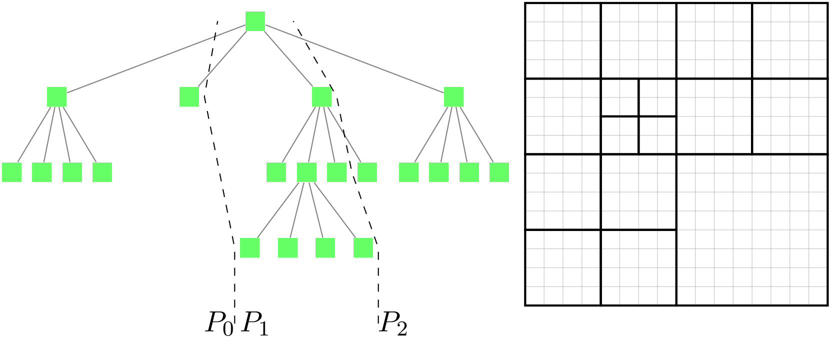


Overview

1. Parallel octree-based dynamic adaptive mesh refinement—definitions and challenges
2. Our scope and other approches
3. **Our approach to adaptive FEM based on parallel octrees**
4. Driving application: mantle convection

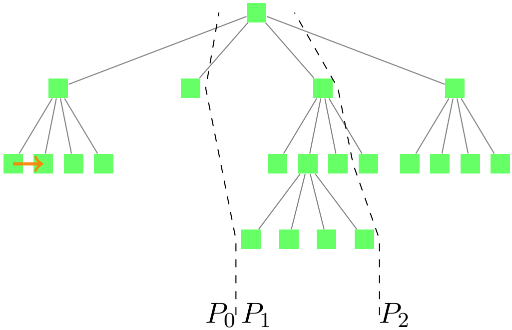
Adaptive FEM based on parallel octrees

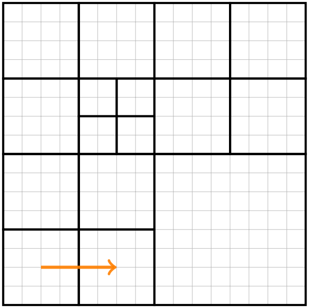
start with initial mesh



Adaptive FEM based on parallel octrees

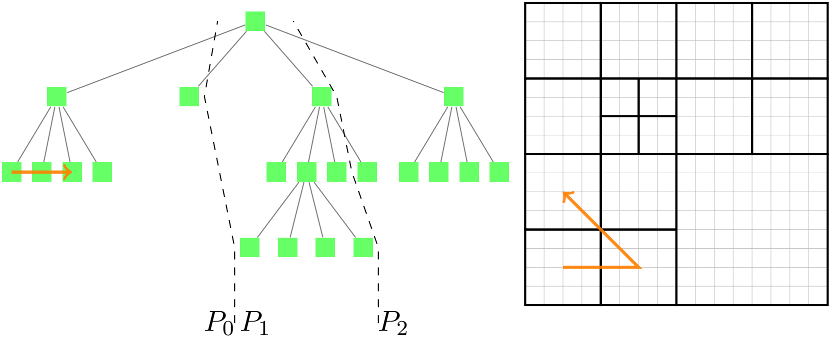
unique global ordering using a space-filling curve





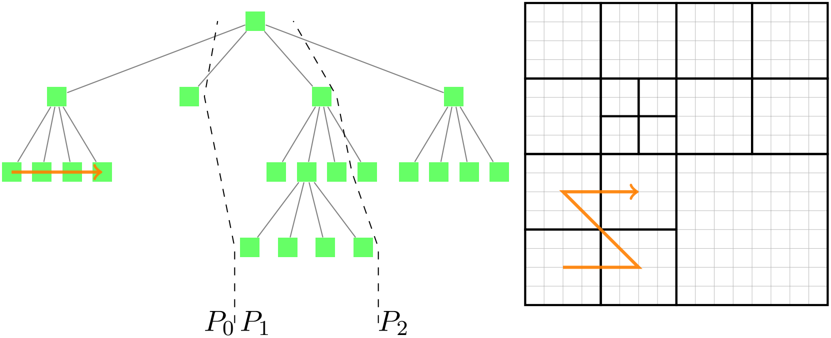
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



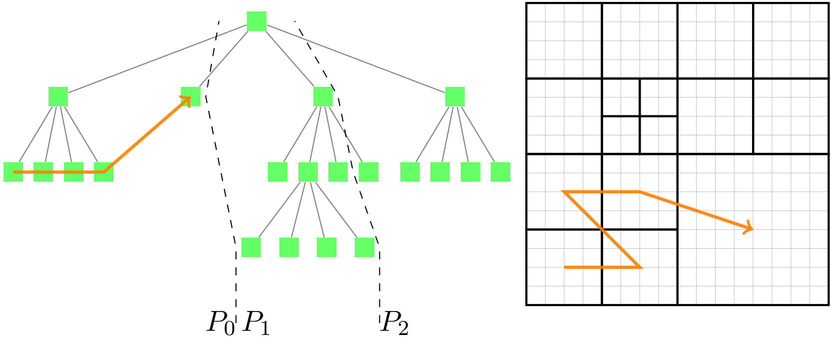
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



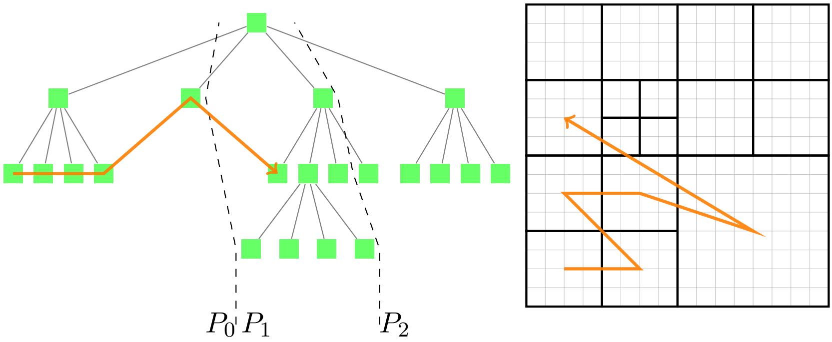
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



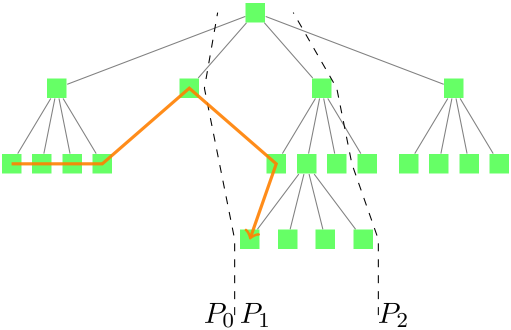
Adaptive FEM based on parallel octrees

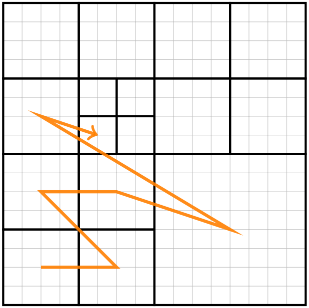
unique global ordering using a space-filling curve



Adaptive FEM based on parallel octrees

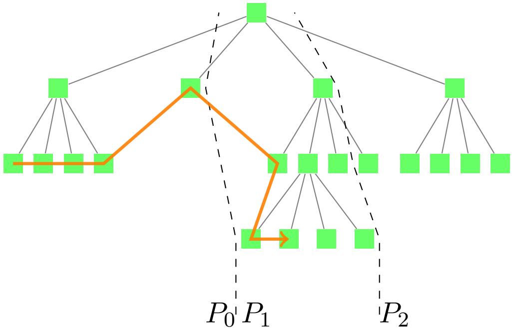
unique global ordering using a space-filling curve

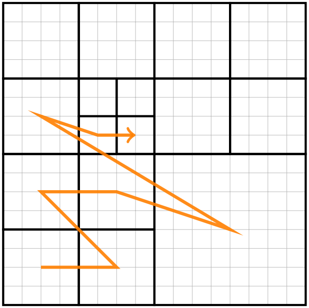




Adaptive FEM based on parallel octrees

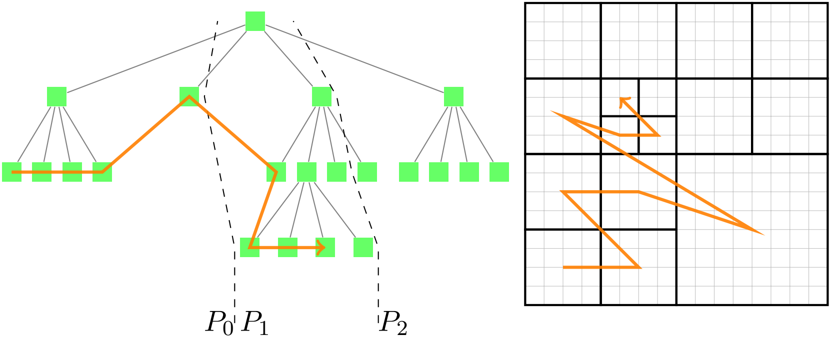
unique global ordering using a space-filling curve





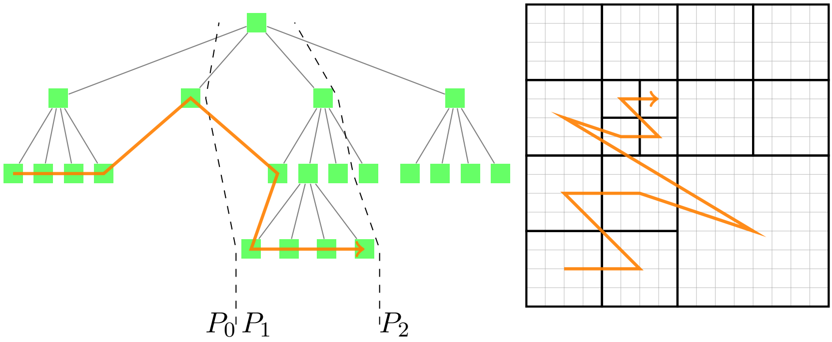
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



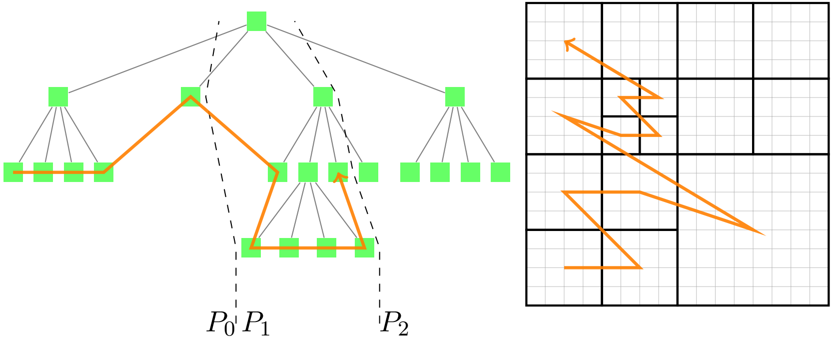
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



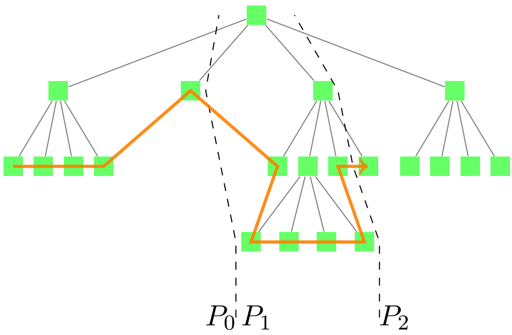
Adaptive FEM based on parallel octrees

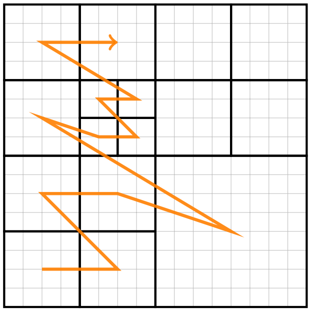
unique global ordering using a space-filling curve



Adaptive FEM based on parallel octrees

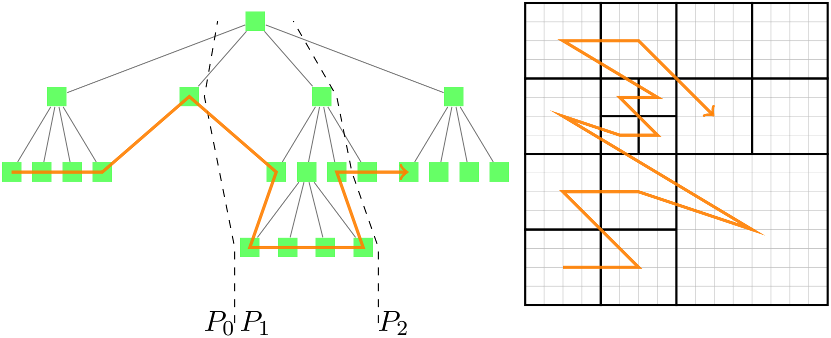
unique global ordering using a space-filling curve





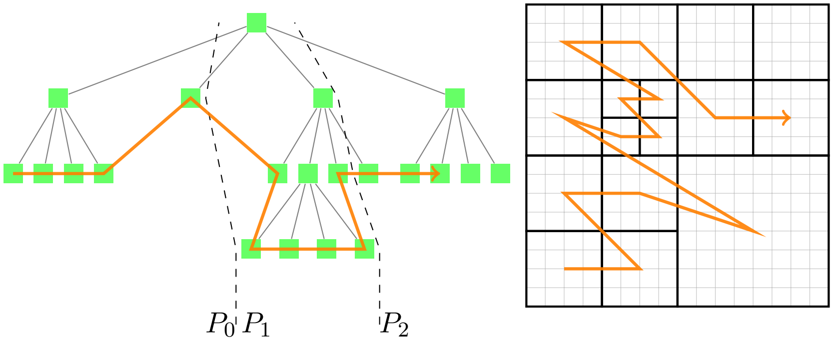
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



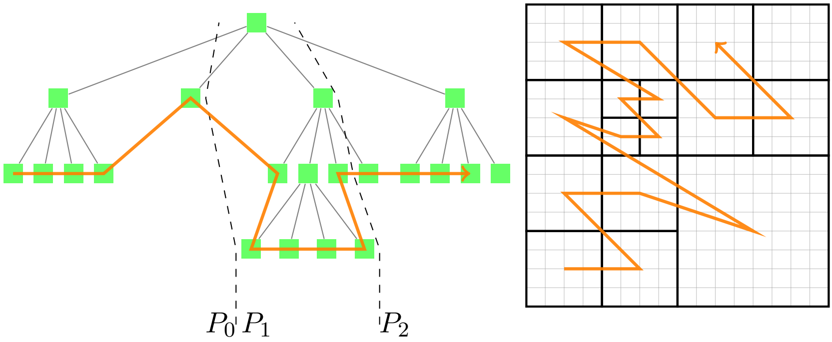
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



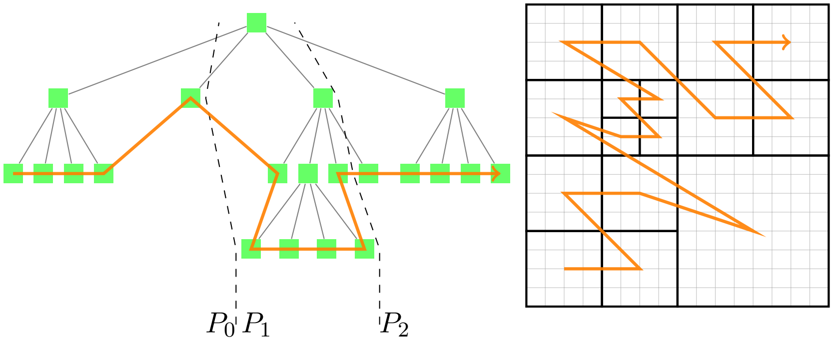
Adaptive FEM based on parallel octrees

unique global ordering using a space-filling curve



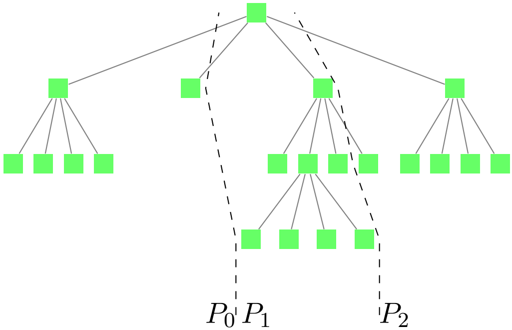
Adaptive FEM based on parallel octrees

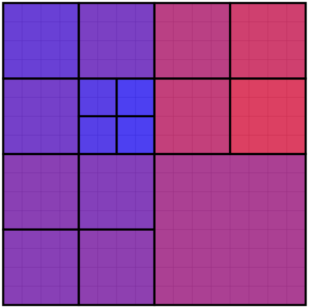
unique global ordering using a space-filling curve



Adaptive FEM based on parallel octrees

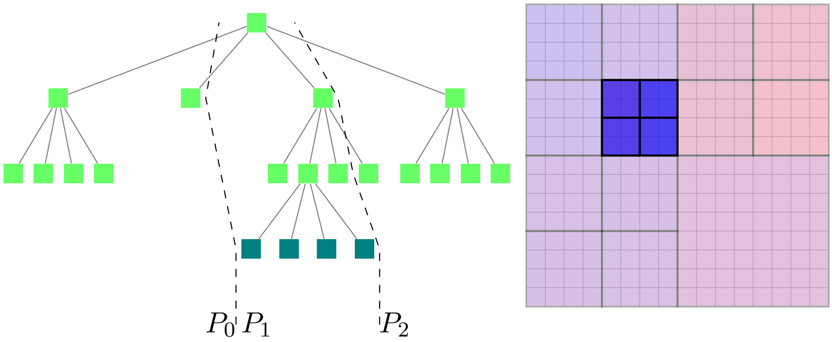
estimate error for each element





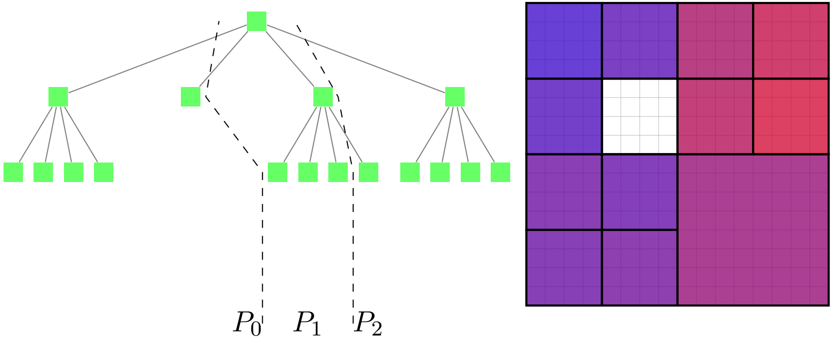
Adaptive FEM based on parallel octrees

select elements for coarsening



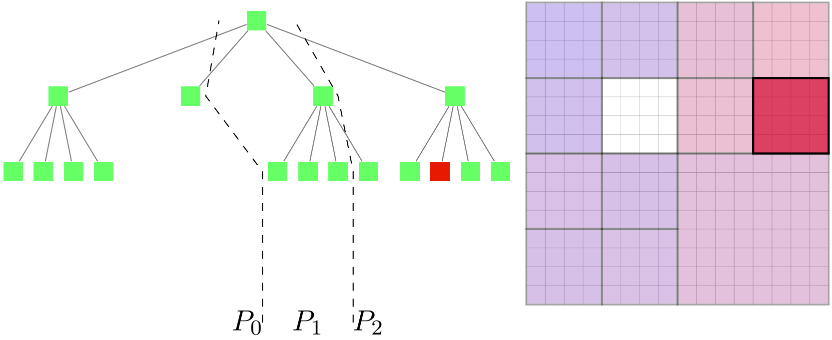
Adaptive FEM based on parallel octrees

coarsening elements



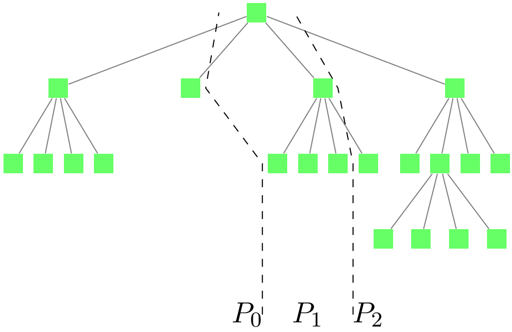
Adaptive FEM based on parallel octrees

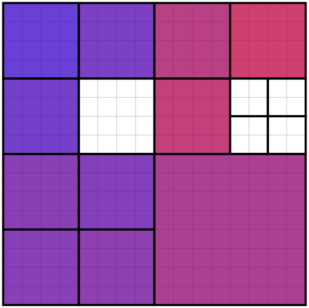
select elements for refinement



Adaptive FEM based on parallel octrees

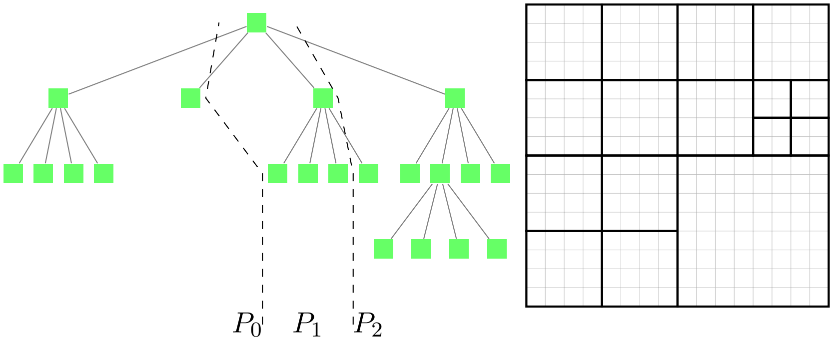
refine elements





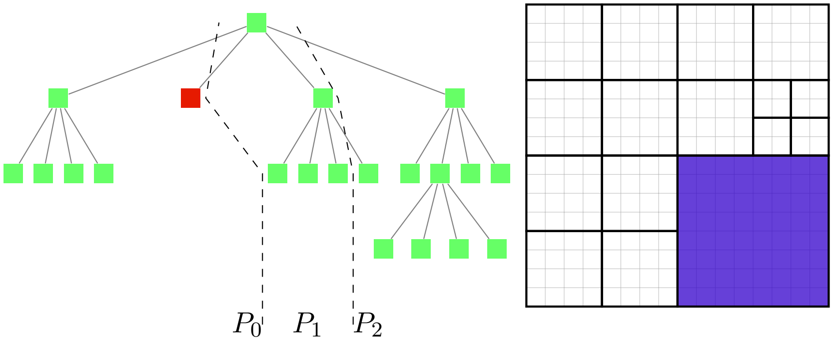
Adaptive FEM based on parallel octrees

2 to 1 balance



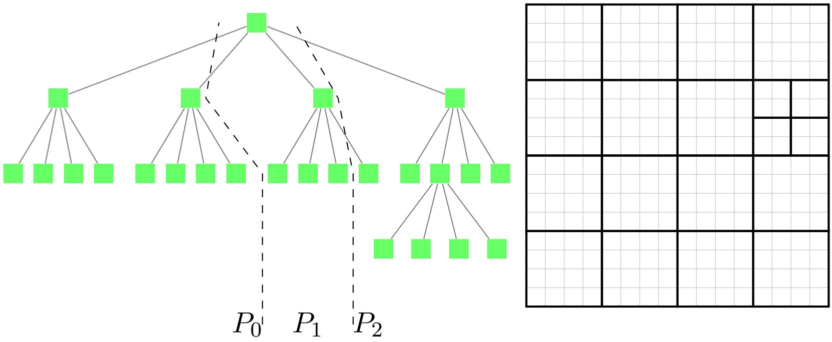
Adaptive FEM based on parallel octrees

2 to 1 balance: select elements for refinement



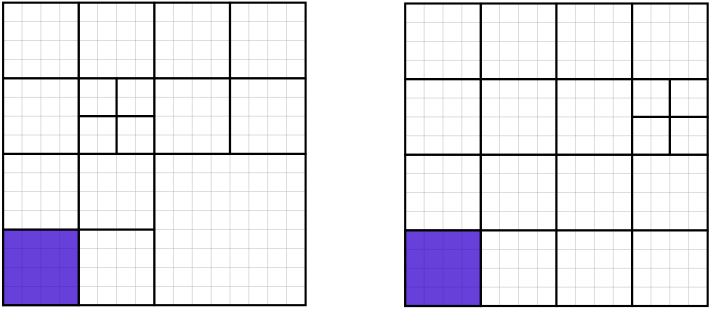
Adaptive FEM based on parallel octrees

2 to 1 balance refine



Adaptive FEM based on parallel octrees

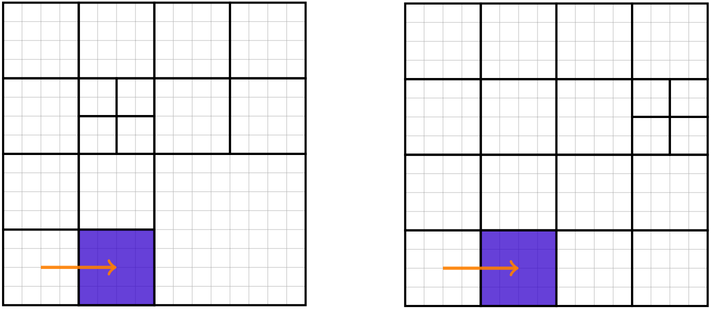
project fields



copy field values

Adaptive FEM based on parallel octrees

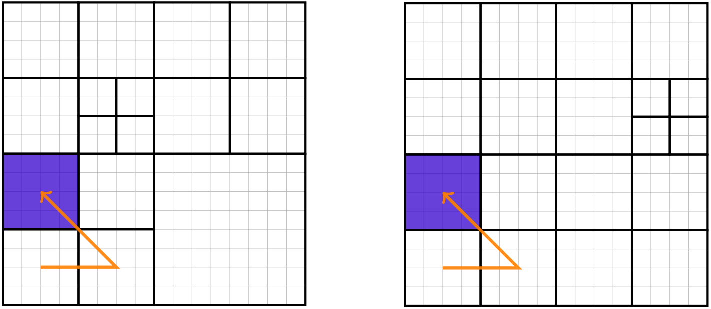
project fields



copy field values

Adaptive FEM based on parallel octrees

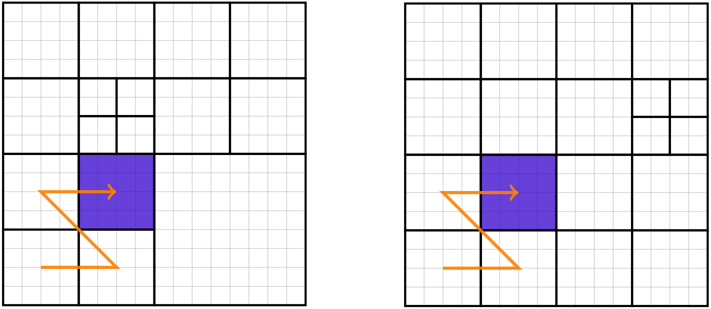
project fields



copy field values

Adaptive FEM based on parallel octrees

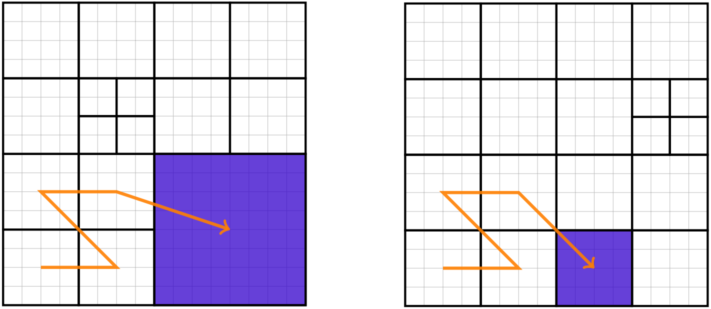
project fields



copy field values

Adaptive FEM based on parallel octrees

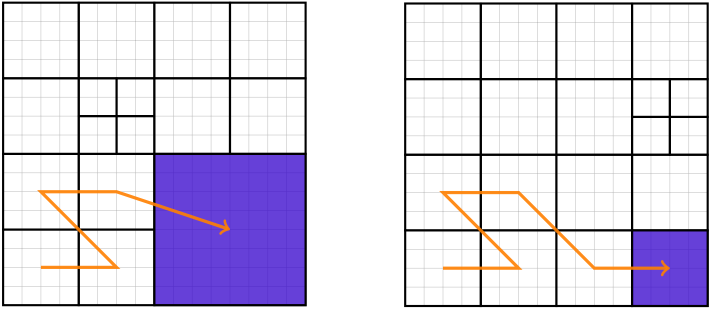
project fields



interpolate coarse → fine

Adaptive FEM based on parallel octrees

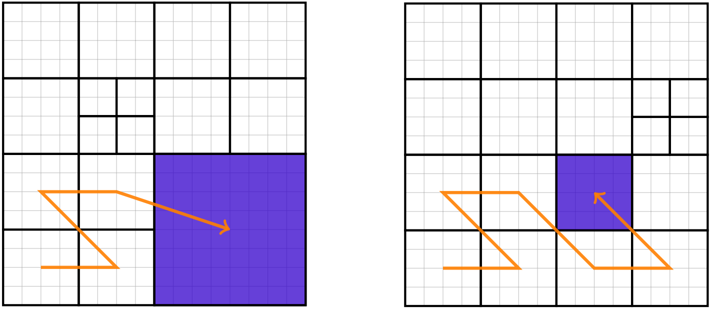
project fields



interpolate coarse $\longrightarrow$ fine

Adaptive FEM based on parallel octrees

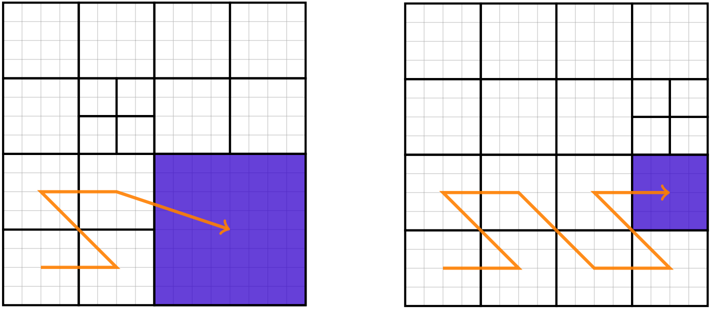
project fields



interpolate coarse $\longrightarrow$ fine

Adaptive FEM based on parallel octrees

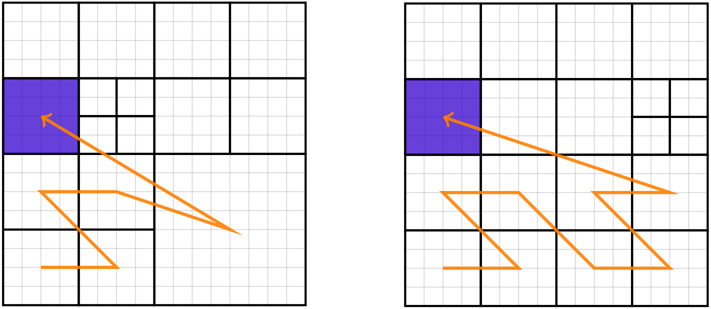
project fields



interpolate coarse → fine

Adaptive FEM based on parallel octrees

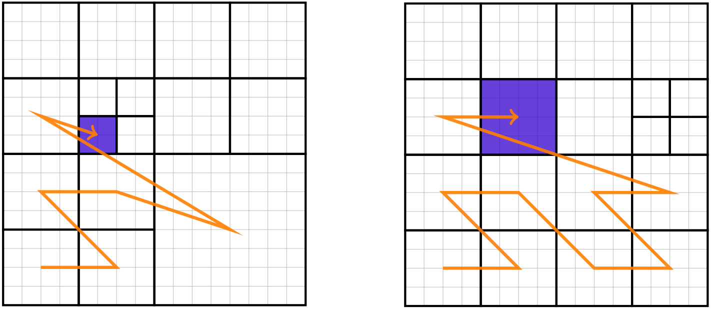
project fields



copy field values

Adaptive FEM based on parallel octrees

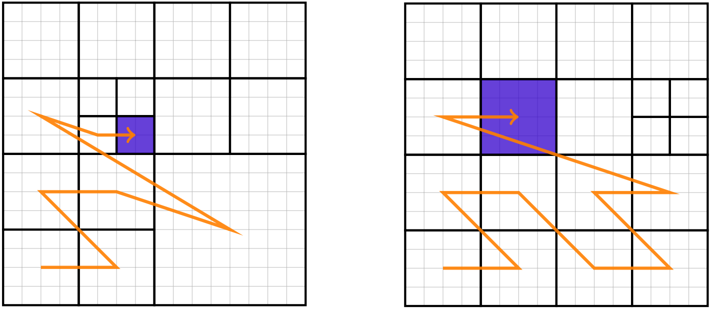
project fields



interpolate fine $\longrightarrow$ coarse

Adaptive FEM based on parallel octrees

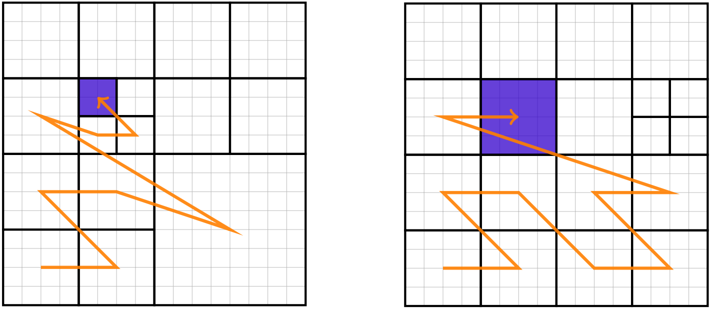
project fields



interpolate fine → coarse

Adaptive FEM based on parallel octrees

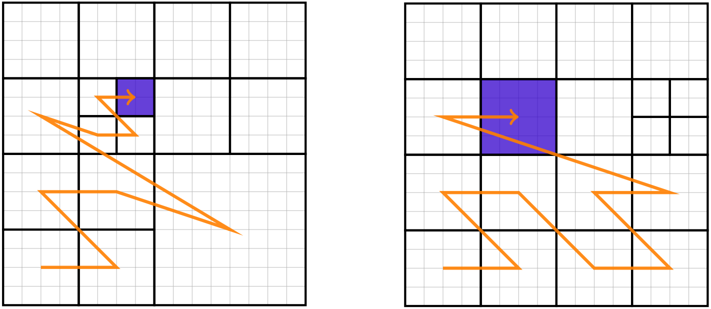
project fields



interpolate fine → coarse

Adaptive FEM based on parallel octrees

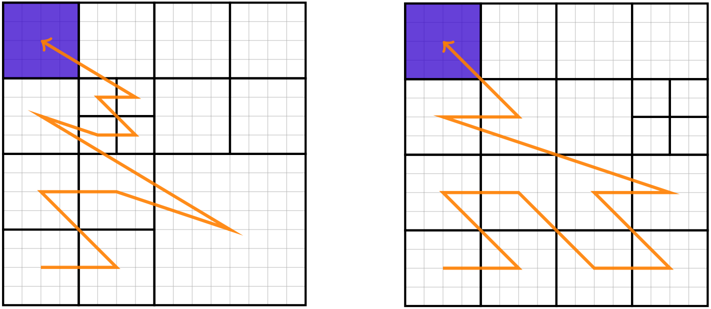
project fields



interpolate fine → coarse

Adaptive FEM based on parallel octrees

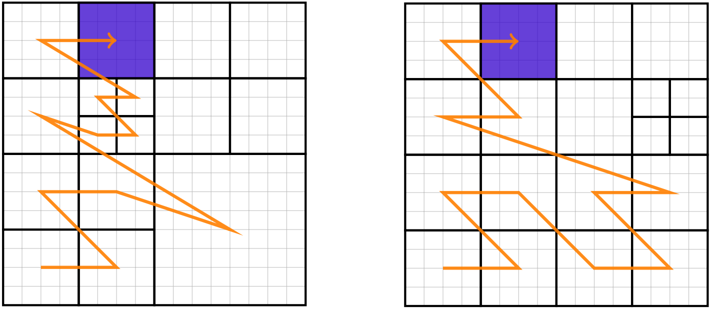
project fields



copy field values

Adaptive FEM based on parallel octrees

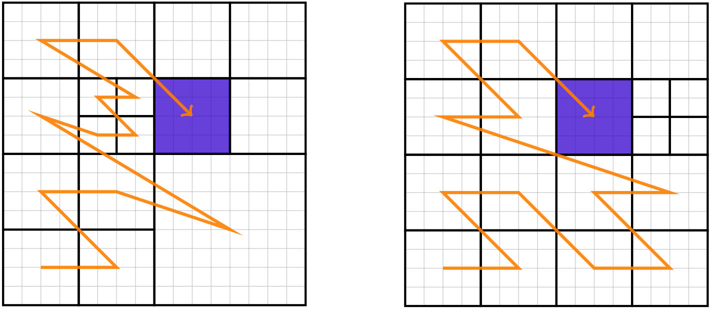
project fields



copy field values

Adaptive FEM based on parallel octrees

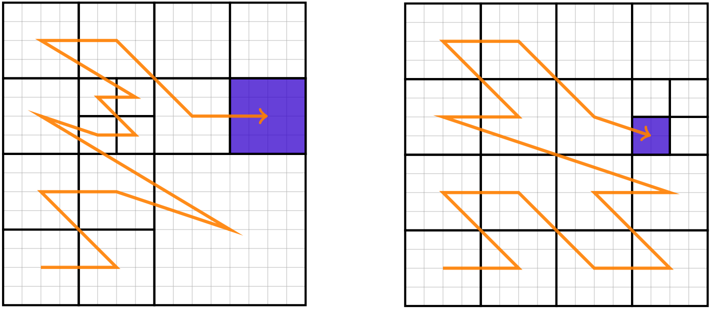
project fields



copy field values

Adaptive FEM based on parallel octrees

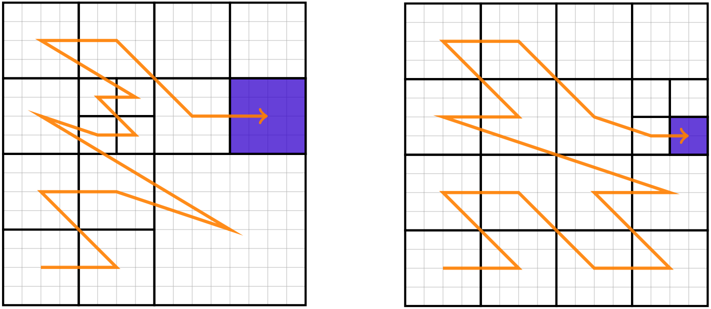
project fields



interpolate coarse → fine

Adaptive FEM based on parallel octrees

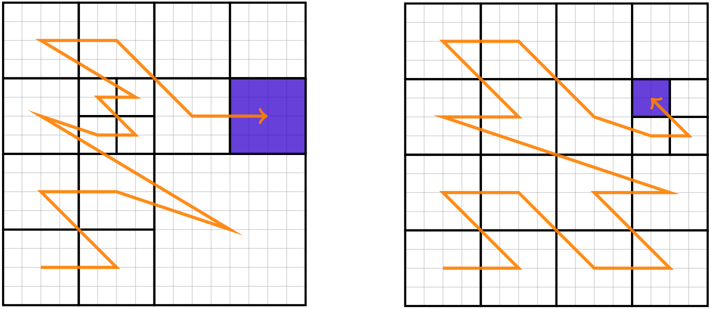
project fields



interpolate coarse → fine

Adaptive FEM based on parallel octrees

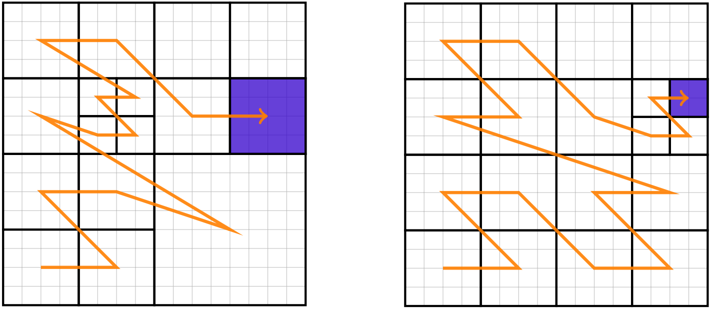
project fields



interpolate coarse → fine

Adaptive FEM based on parallel octrees

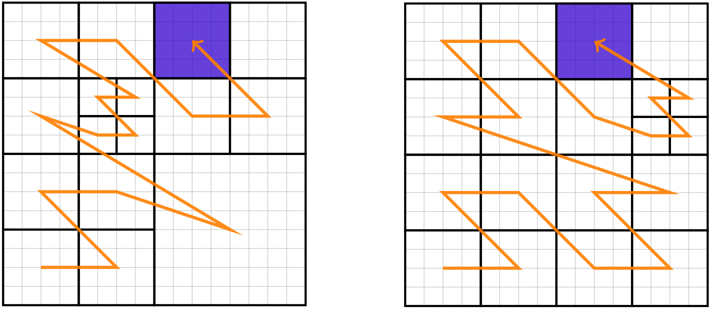
project fields



interpolate coarse $\longrightarrow$ fine

Adaptive FEM based on parallel octrees

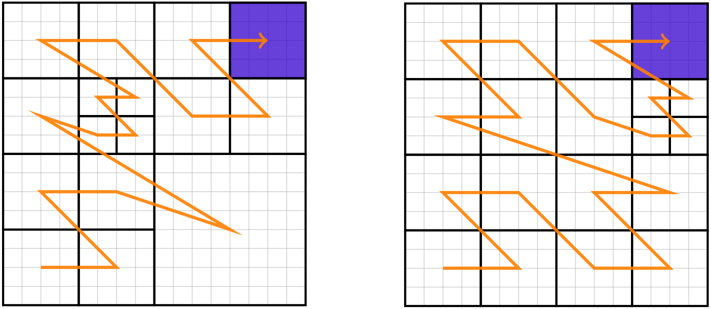
project fields



copy field values

Adaptive FEM based on parallel octrees

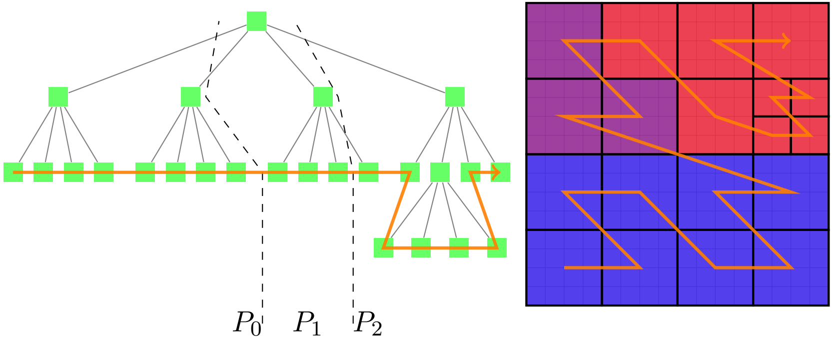
project fields



copy field values

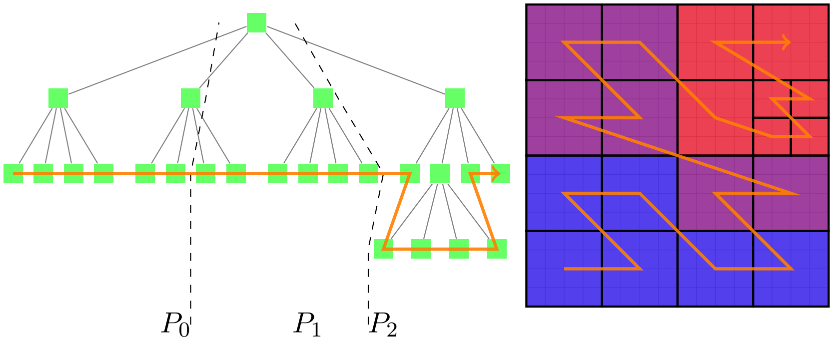
Adaptive FEM based on parallel octrees

initial partitioning



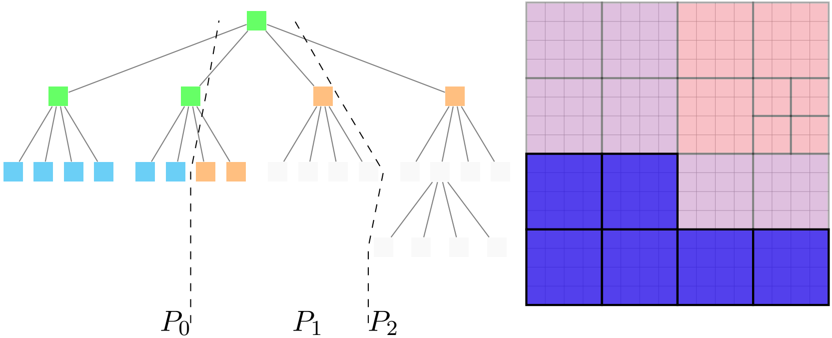
Adaptive FEM based on parallel octrees

load balanced partitioning



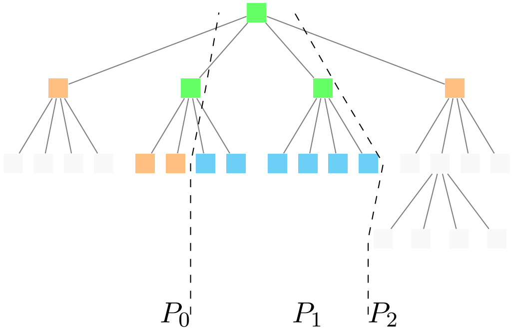
Adaptive FEM based on parallel octrees

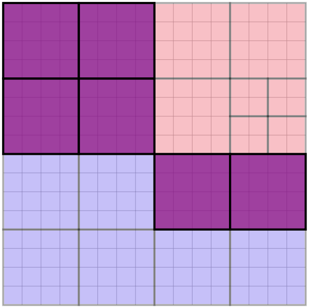
P_0 portion of the tree



Adaptive FEM based on parallel octrees

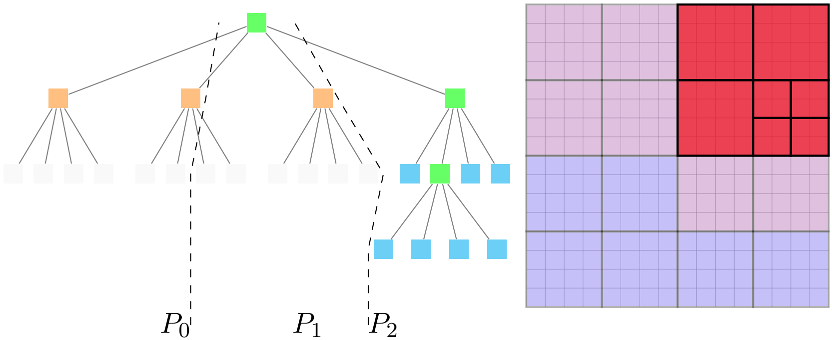
P_1 portion of the tree





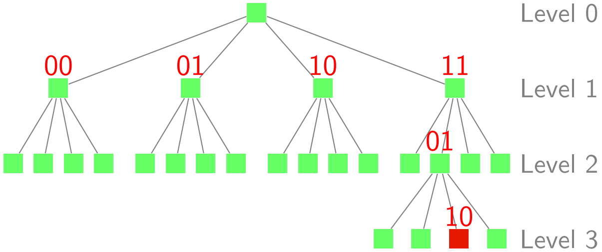
Adaptive FEM based on parallel octrees

P_2 portion of the tree



Adaptive FEM based on parallel octrees

addressing an octant



concat bit-patterns

110110

↓

pad zeros

11011000

↓

append the level

11011000011

Adaptive FEM based on parallel octrees

addressing an octant

get the coordinate (12,10)

↓

use binary (0000,0000)

↓

interleave the bits

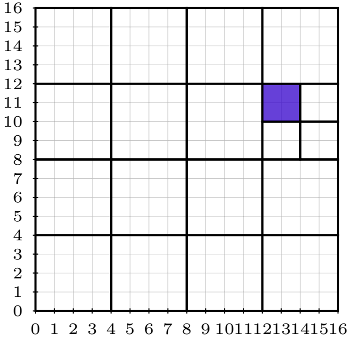
1100 1010

11011000

↓

append the level

11011000011



Overview

1. Parallel octree-based dynamic adaptive mesh refinement—definitions and challenges
2. Our scope and other approches
3. Our approach to adaptive FEM based on parallel octrees
4. **Driving application: mantle convection**

Driving application: Mantle convection

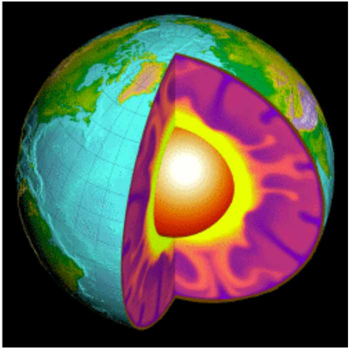


Figure: Taken from US Geological Survey `https://www.usgs.com`

- main control on **thermal** and **geological evolution**
- central for understanding
  - **plate tectonics**
  - **volcanism**
  - **dynamics of the solid earth**

Driving application: Mantle convection

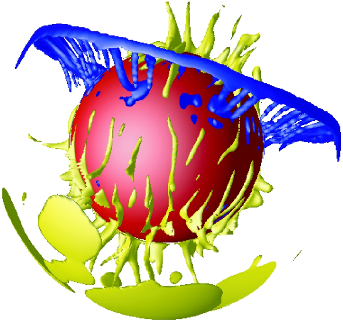


Figure: Isosurfaces of temperature field; plot courtesy of Shijie Zhong

Driving application: Mantle convection

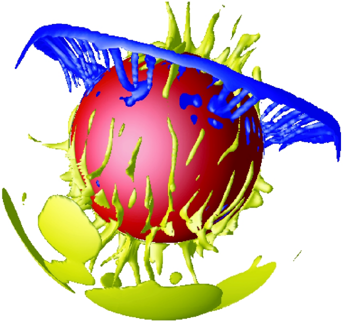


Figure: Isosurfaces of temperature field; plot courtesy of Shijie Zhong

Resolution down to $\sim 1\text{km}$ needed to resolve fine structures

- $\Rightarrow \sim 10^{12}$ elements on **uniform grid**
- $\Rightarrow \sim 10^{9}$ elements on **adaptive grid**

Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$

Variables:

- T . . . temperature
- u . . . velocity
- p . . . pressure

Parameters:

- $\mathsf{Ra} \sim 10^6 - 10^9$ . . . Rayleigh number
- γ . . . heat production rate
- $\eta(T) \cong \eta_o \exp(1 - E_o T)$ . . . viscosity
- e_r . . . radial direction

Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$

$T^0 = T(t^0)$

$k := 0$

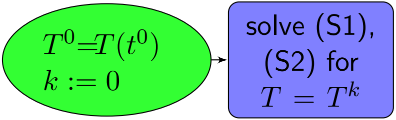
Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$



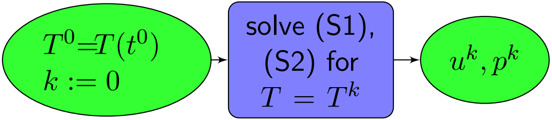
Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$



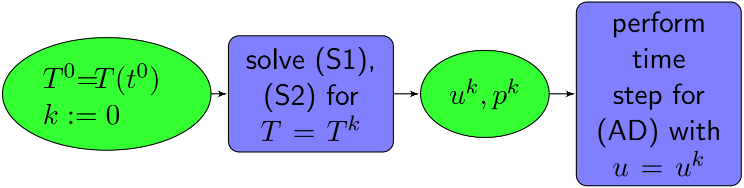
Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$



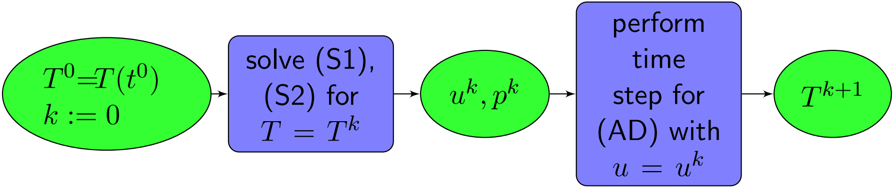
Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$



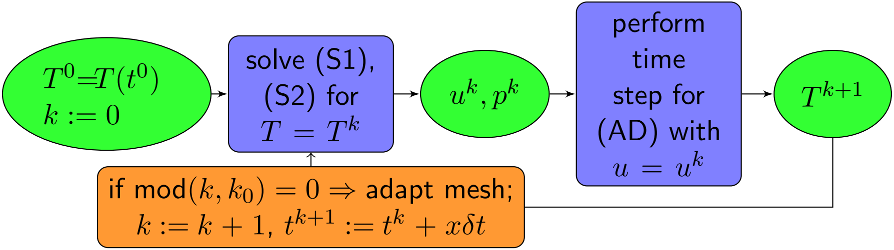
Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathsf{Ra}\, T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$



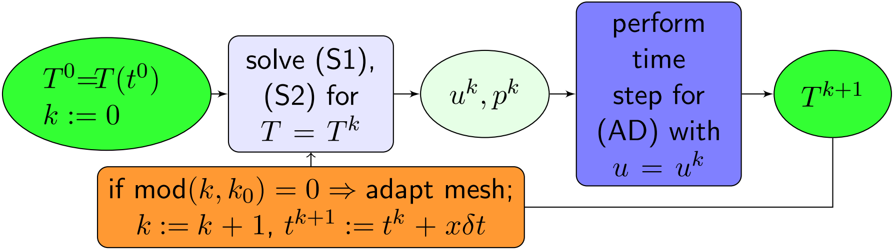
Driving application: Mantle convection

Nondimensional model for mantle convection

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla^2 T - \gamma = 0, \qquad \text{(AD)}$$

$$\nabla \cdot \left[\eta(T)\left(\nabla u + \nabla^\top u\right)\right] - \nabla p + \mathrm{Ra} T e_r = 0, \qquad \text{(S1)}$$

$$\nabla \cdot u = 0. \qquad \text{(S2)}$$



Driving application: Mantle convection

Discretization and Solution

- **Trilinear FEM** for temperature, velocity and pressure (the framework allows the use of higher-order FEM)

Driving application: Mantle convection

Discretization and Solution

- **Trilinear FEM** for temperature, velocity and pressure (the framework allows the use of higher-order FEM)
- Conforming approximation is enforced by algebraic elimination of **hanging** nodes

Driving application: Mantle convection

Discretization and Solution

- **Trilinear FEM** for temperature, velocity and pressure (the framework allows the use of higher-order FEM)
- Conforming approximation is enforced by algebraic elimination of **hanging** nodes
- Solver operates only on **anchored** nodes

Driving application: Mantle convection

Discretization and Solution

- **Trilinear FEM** for temperature, velocity and pressure (the framework allows the use of higher-order FEM)
- Conforming approximation is enforced by algebraic elimination of **hanging** nodes
- Solver operates only on **anchored** nodes
- FEM **stabilization**:
  - Streamline Upwind/Petrov Galerkin (**SUPG**) for advection-diffusion system
  - **Pressure stabilization** for Stokes equation

Driving application: Mantle convection

Discretization and Solution

- **Trilinear FEM** for temperature, velocity and pressure (the framework allows the use of higher-order FEM)
- Conforming approximation is enforced by algebraic elimination of **hanging** nodes
- Solver operates only on **anchored** nodes
- FEM **stabilization**:
  - Streamline Upwind/Petrov Galerkin (**SUPG**) for advection-diffusion system
  - **Pressure stabilization** for Stokes equation
- **AMG-preconditioned MINRES** for the solution of the Stokes equation

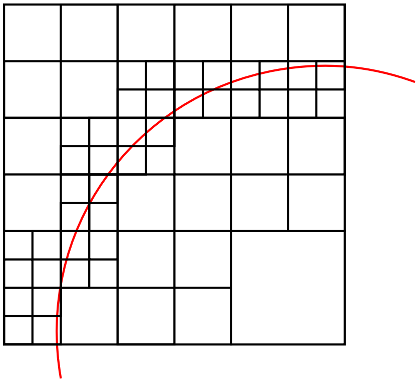
Driving application: Mantle convection

Discretization and Solution

- **Trilinear FEM** for temperature, velocity and pressure (the framework allows the use of higher-order FEM)
- Conforming approximation is enforced by algebraic elimination of **hanging** nodes
- Solver operates only on **anchored** nodes
- FEM **stabilization**:
  - Streamline Upwind/Petrov Galerkin (**SUPG**) for advection-diffusion system
  - **Pressure stabilization** for Stokes equation
- **AMG-preconditioned MINRES** for the solution of the Stokes equation
- α-**timestepping** for advection-diffusion equation (does not require system solves)

Driving application: Mantle convection

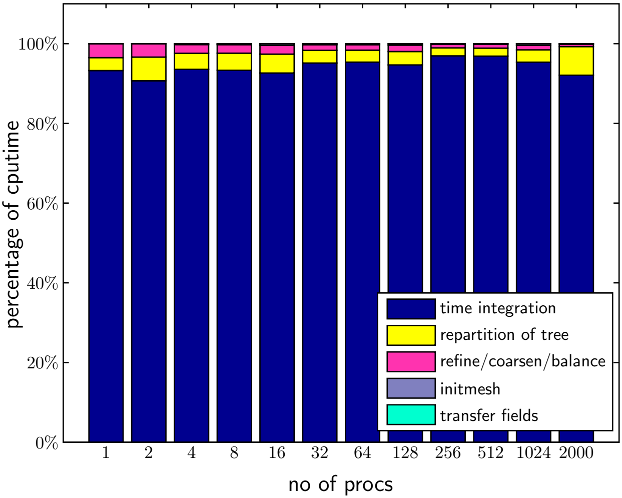
From the cube to spherical geometry



- embed mantle into octree box
- refine near the boundary
- intial approach: use a fictitous domain method to enforce boundary conditions (via penalties or Lagrange multipliers)
- explore immersed finite element method

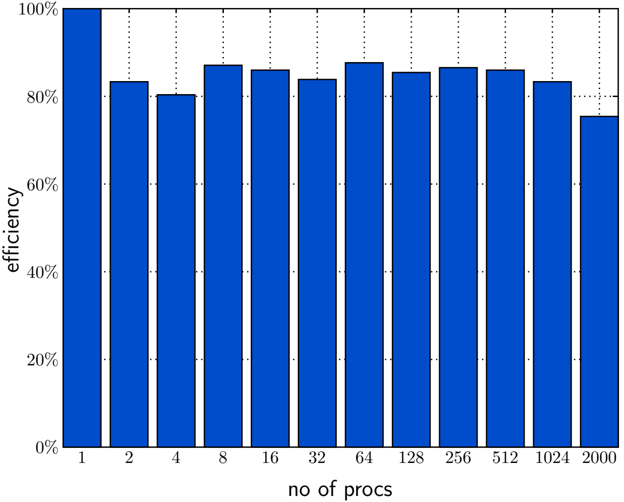
Driving application: Mantle convection

Percentage of wall clock time for Rhea components



Driving application: Mantle convection

Results for Rhea: weak scalings for advection-diffusion equation with adaptive mesh refinement/coarsening on up to 2000 processors: efficiency per time step



Future work

We intend to . . .

- develop a scalable Stokes solver to attack the full mantle convection problem
- make our earth round (i.e., extend to spherical geometry)
- optimize and test the approach for $O(10^5)$ processors
- long term: inverse problem (iRhea)

Conclusions

- All AMR compents (error estimation, coarsening, refinement, balancing, repartitioning) consume less than 8% of total run time on up to 2000 processors
  - efficiency can only improve with (implicit, nonlinear) Stokes solver
- Parallel efficiency drops from 83.3% on 2 procs to 75.4% on 2000 procs
- Parallel adaptive mesh refinement/coarsening for FEM that scales up to thousands of processors is possible!
- In principle, higher-order elements and irregular boundaries are possible