Block-Structured AMR for Exascale Applications

John Bell

Computational Research Division
Lawrence Berkeley National Laboratory

Turbulent Flow Simulation at the Exascale: Opportunities and Challenges Workshop
Washington, DC
August 4-5, 2015

PDE-based multiphysics applications

Many PDE-based applications require high-fidelity simulation of multi-scale / multi-physics phenomena

- Combustion
- Fusion
- Astrophysics
- Atmospheres
- Climate
- Cosmology

Characteristic of these problem areas is that they couple a number of different physical processes across a range of length and time scales

How can we exploit the structure of these problems in developing efficient simulation methodology for exascale architectures?

- What are the characteristics of these type of problems
- Specialized formulations to reflect relationship between temporal scales
- Adaptive mesh refinement for spatially varying resolution requirements

Multiphysics AMR

Algorithmic components

- Advective transport
- Diffusive processes
  - Viscosity
  - Thermal conductivity
  - Species transport
  - Diffusion approximations to radiation
- Reactions
- Elliptic constraints
  - Low Mach number constraints
  - Self-gravity

AMR applications can be built out of basic components for hyperbolic, elliptic and parabolic PDEs

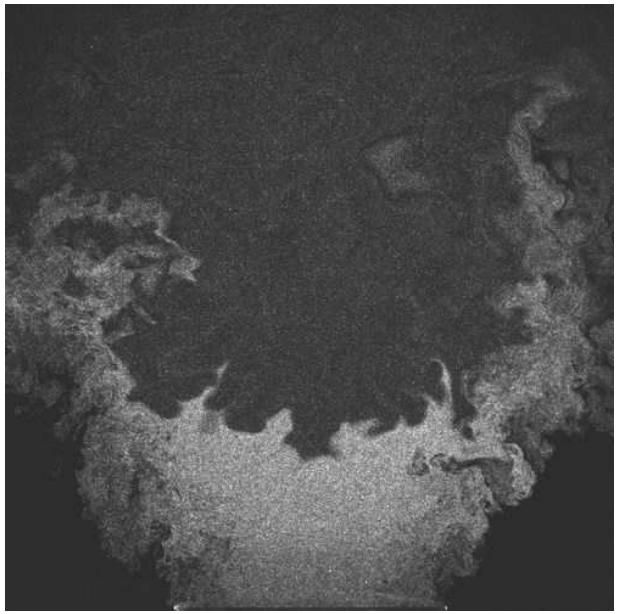
A combustion example

Spatial Scales

- Domain: $\approx$ 10 cm
- Flame thickness: $\delta_T \approx 1$ mm
- Integral scale: $\ell_t \approx 2 - 6$ mm

Temporal Scales

- Flame speed $O(10^2)$ cm/s
- Mean Flow: $O(10^3)$ cm/s
- Acoustic Speed: $O(10^5)$ cm/s



Mie Scattering Image

Fast chemical time scales

Exploit mathematical structure to compute more efficiently

- Low Mach number formulation
- AMR

Low Mach number equations

$$\textbf{Momentum} \quad \frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = -\nabla \pi + \nabla \cdot \mathbf{\Pi}$$

$$\textbf{Energy} \quad \frac{\partial \rho h}{\partial t} + \nabla \cdot \left(\rho h \vec{U}\right) - \frac{D p_0}{D t} = \nabla \cdot (\lambda \nabla T) + \sum_k \nabla \cdot (h_k \mathcal{F}_k)$$

$$\textbf{Species} \quad \frac{\partial (\rho Y_k)}{\partial t} + \nabla \cdot (\rho U Y_k) = \nabla \cdot \mathcal{F}_k + m_k \dot{\Omega}_k$$

$$\textbf{Mass} \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0$$

Equation of state $p_0 = p(\rho, T, Y_k)$ becomes a constraint on the evolution

Low Mach number system can be advanced at fluid time scale instead of acoustic time scale but . . .

Requires effective integration techniques for this more complex formulation

Low Mach number constraint – general EOS

Low Mach number model is a system of PDE's evolving subject to a constraint given by equation of state

Differentiate the EOS along particle paths

$$\frac{dp_0}{dt} + \rho \frac{\partial p}{\partial \rho} \nabla \cdot U = \frac{\partial p}{\partial T} \frac{DT}{Dt} + \sum_k \frac{\partial p}{\partial y_k} \frac{Dy_k}{Dt}$$

An equation for T can be derived form the enthalpy equation by writing $h = h(p, T, Y_k)$. Using this equation and the species equation, we can rewrite the Lagrangian derivatives in terms of thermodynamic variables and their spatial derivatives to give

$$\frac{1}{\Gamma_1 p_0} \frac{dp_0}{dt} + \nabla \cdot U = H$$

In open domain p_0 is constant and this defines a constraint on U. In closed system average of H defines evolution of p_0

Use generalization of approaches for incompressible flow to integrate low Mach number system

Generalized projection formulation

Fractional step scheme (open domains)

- Advance velocity and thermodynamic variables
  - Specialized advection algorithms
  - Diffusion
  - Stiff reactions
- Project solution back onto constraint – variable coefficient elliptic PDE, multigrid

Stiff kinetics relative to fluid dynamical time scales

$$\frac{\partial(\rho Y_k)}{\partial t} + \nabla \cdot (\rho U Y_k) = \nabla \cdot \mathcal{F}_k + m_k \dot{\Omega}_k$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho U h) = \nabla \cdot (\lambda \nabla T) + \sum_k \nabla \cdot (h_k \mathcal{F}_k)$$

Spectral deferred correction

- Treat processes separately using algorithms appropriate for that process
- Couple processes iteratively by rewriting system as an integral in time

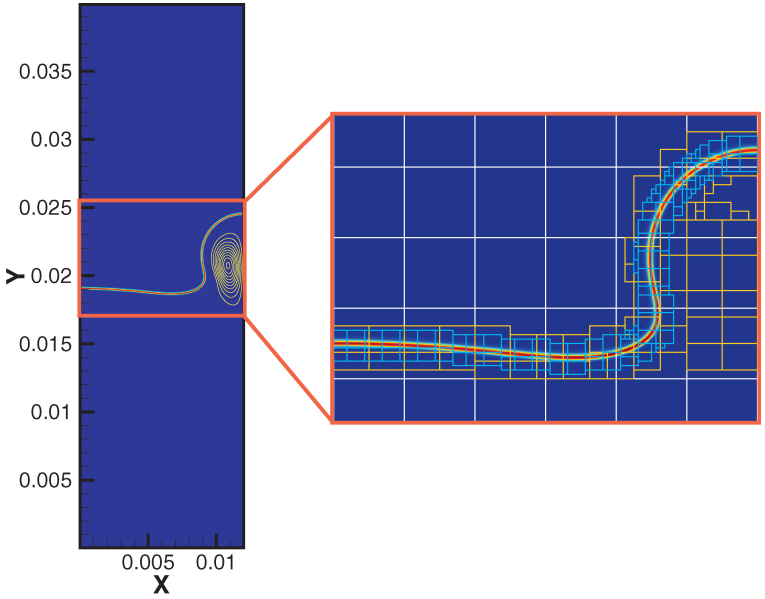
AMR

Block-structured hierarchical grids

- Amortize irregular work

Each grid patch (2D or 3D)

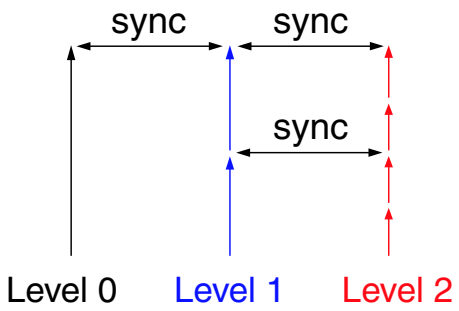
- Logically structured, rectangular
- Refined in space and time by evenly dividing coarse grid cells
- Dynamically created/destroyed



2D adaptive grid hierarchy

Subcycling:

- Advance level ℓ, then
  - Advance level $\ell + 1$
    level ℓ supplies boundary data
  - Synchronize levels ℓ and $\ell + 1$



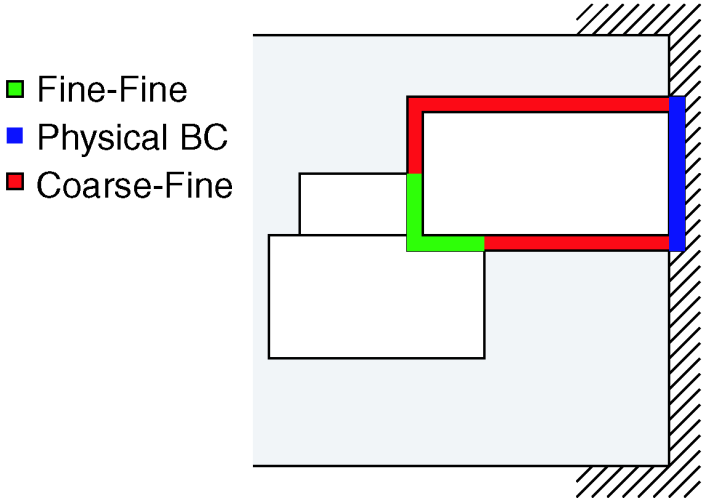
AMR Synchronization

Coarse grid supplies Dirichlet data as boundary conditions for the fine grids.

Errors take the form of flux mismatches at the coarse/fine interface.

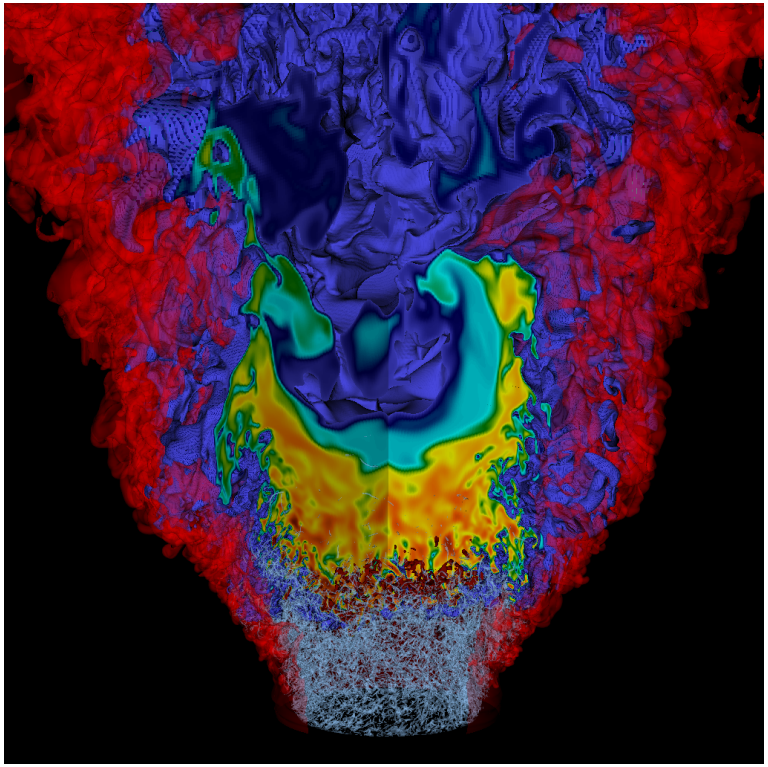
Design Principles:

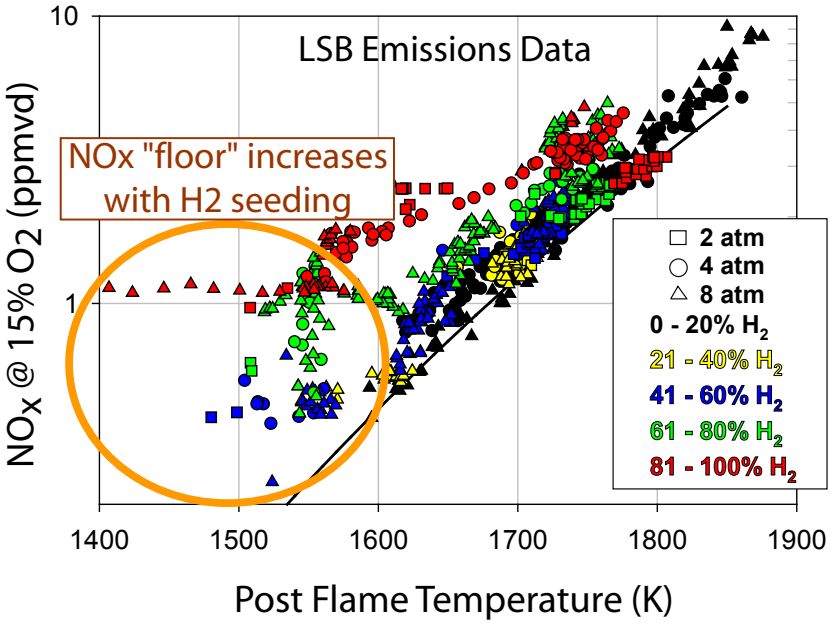
- Define what is meant by the solution on the grid hierarchy.
- Identify the errors that result from solving the equations on each level of the hierarchy "independently".
- Solve correction equation(s) to "fix" the solution.
- Correction equations match the structure of the process they are correcting.

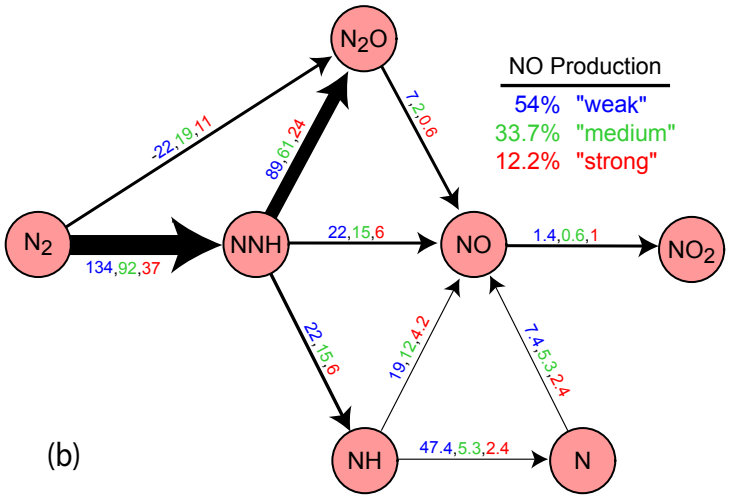


Preserves properties of single-grid algorithm

LMC simulation of low swirl burner emissions







Emissions chemistry (15 species and 58 reactions) – Glarborg

Effective resolution of 4096^3

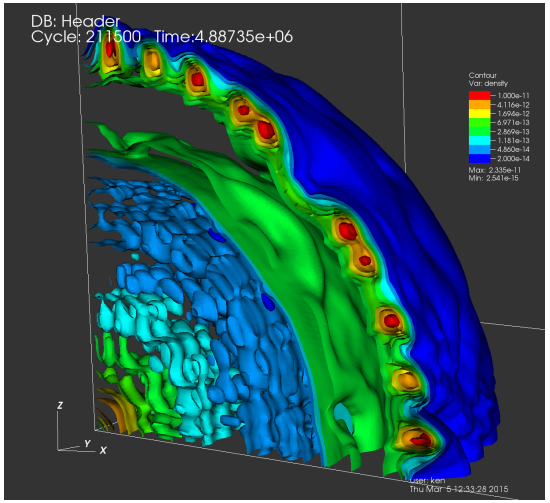
Compressible astrophysics and cosmology

CASTRO

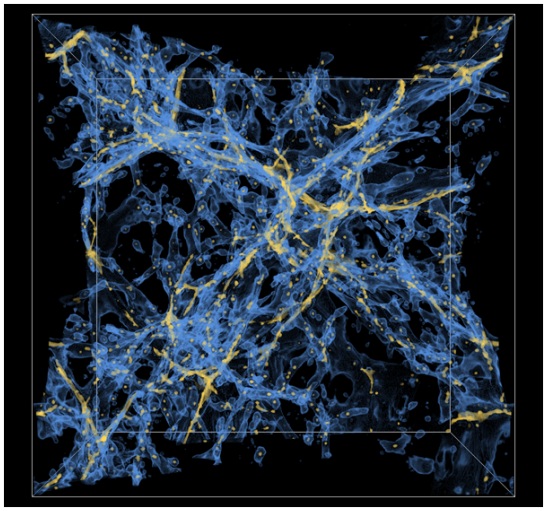
- Compressible flow
- Gray and multigroup flux-limited diffusion
- Nuclear burning
- Self-gravity

Nyx

- Compressible flow
- Self-gravity
- Particle representation of dark matter
- Expanding universe



Pulsational pair instability supernova



Lyman alpha forest

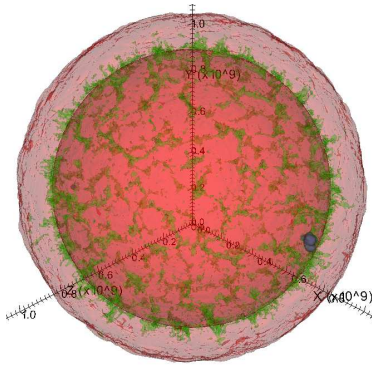
Low Mach number stratified flow

MAESTRO

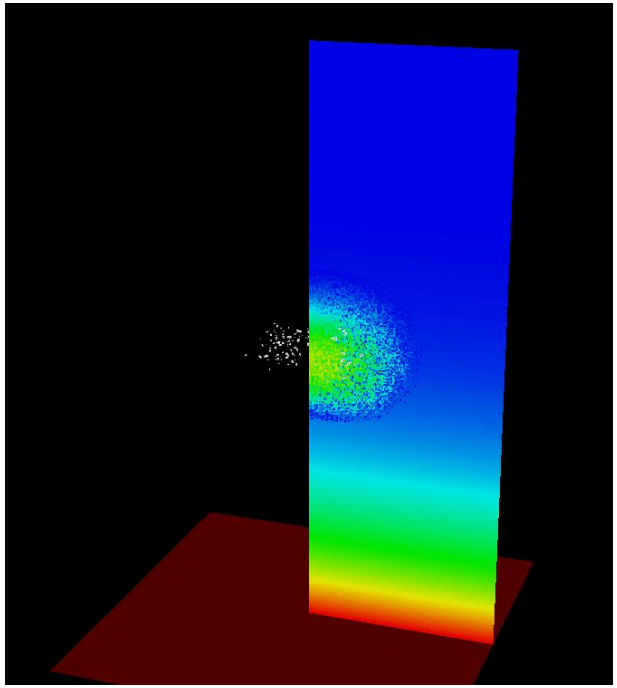
- Low Mach number astrophysical flows
- Stratified “atmosphere”
- General equation of state
- Nuclear reactions

Atmospheric modeling

- Ideal gas equation of state
- Replace nuclear reactions with moisture physics



Subchandra ignition



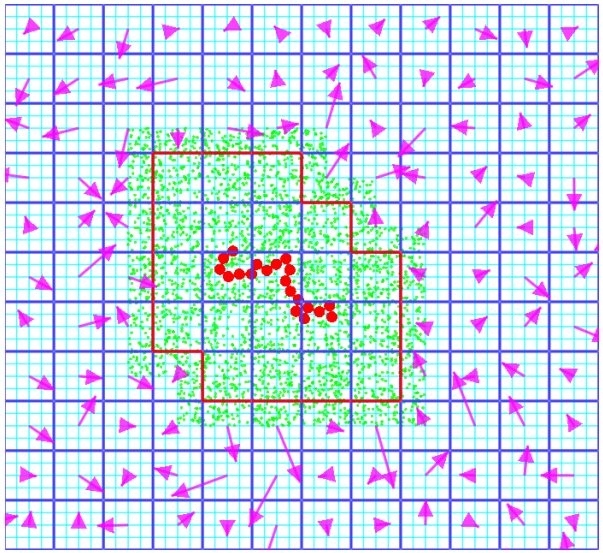
Cloud formation

Hybrid methods for multiscale problems

Look at systems where physical description changes with scale

Example: hybrid fluid / particle algorithms

- Molecular description – correct but expensive
- Continuum CFD – cheap but doesn't model correct physics in some regions
- Hybrid – Use different models for the physics in different parts of the domain
  - Molecular model only where needed
  - Cheaper continuum model in the bulk of the domain



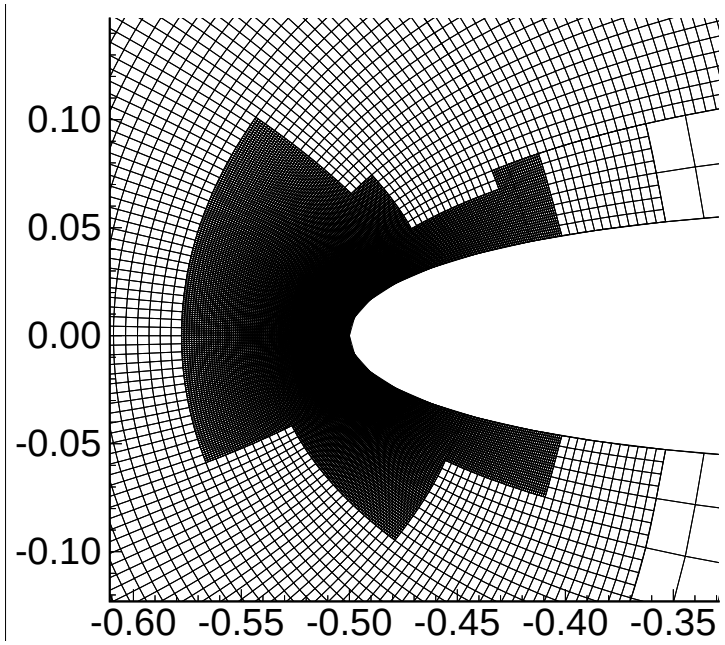
AMR paradigm can be used to construct hybrid algorithms.

Key is understanding how different descriptions talk to each other

Complex geometry

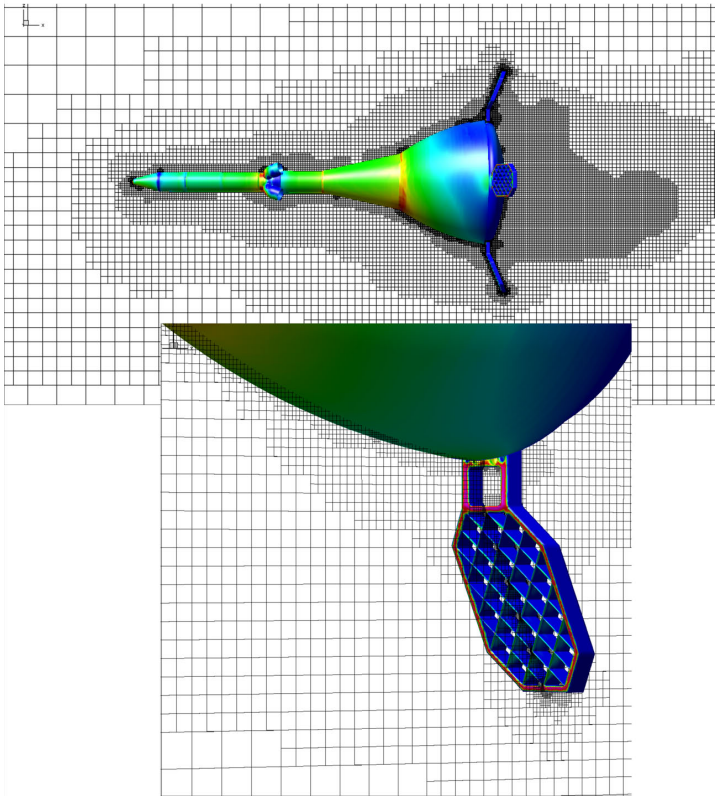
Strategies for complex geometry

- Curvilinear adaptive grids
- Multiblock – multiple logically rectangular grids
  - Traditional mapped multiblock
  - Overset mesh – Henshaw
- Embedded boundary (aka Cartesian grid methods)
- Immersed interface algorithms



Embedded boundary methods

- Geometry treated as tracked interface in Carteisan grid
- Bulk of domain is discretized with standard approaches
- Straightforward coupling to AMR
- Automated grid generation from CAD descriptions (CART3D)
    - NASA uses this approach for design optmization



Courtesy of M. Aftosmis and M. Berger

Implementation

One wants to implement these types of algorithms within a software framework that supports the development of block-structured AMR algorithms

- Represent dynamically changing hierarchical solution
- Manage error estimation and regridding operations
- Orchestrate multistep algorithms and synchronization
- Support for iterative methods for implicit algorithms

There are a number of frameworks that support implementation of these types of algorithms

We use BoxLib

- Data structures
- Operations on those data structures
- Model for parallelization

See Dubey et al., Journal of Parallel and Distributed Computing, 2014.

Hybrid parallelization model

Block-structured AMR admits a natural hybrid parallelization model

- Coarse-grained
  - Distribute grid patches to “nodes”
  - Currently done with MPI
- Fine-grained
  - Use threads to operate on patches
  - Currently done with OpenMP
  - Some strategies work better than others

AMR maps several large patches to each node $\rightarrow$ algorithms can utilize many threads per node

Suggests AMR can be mapped effectively to many-core architectures

Decomposes problem of mapping to an exascale machine

- Distribute patches to nodes
- Distribute work on patches to thread within node

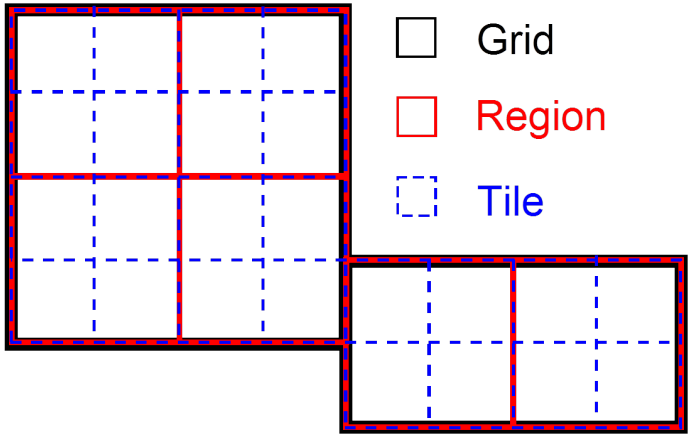
AMR Research Issue 1 – Distribute work within a node

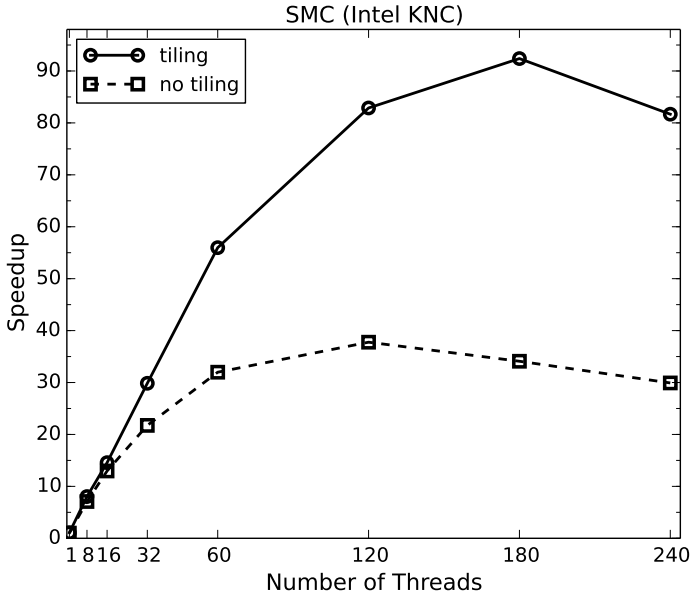
Possibly severe NUMA effects on nodes

- Decompose large patches into smaller pieces
- Distribute data to respect NUMA issues
- For AMR this must be done dynamically

Strategy – hierarchical tiling

- Regional tiles for data distribution
- Logical tiles for loops
  - Manage working set size
  - Minimize cache hits





AMR Research Issue 2 – Distribute patches to nodes

Current data distribution strategy

- Patches defined by regridding procedure
- Estimate work based on floating point
- Algorithm distributes work to nodes to balance estimated load
  - Heuristic knapsack algorithm
  - Space-filling curve

Need more “holistic” data layout strategy

- Communication costs
- Network topology
- Data movement costs to move data
- Non-uniform machine behavior

Current strategy based on collecting performance data during execution to guide data distribution

- Complex multiple objective optimization problem that is hard to formulate and to solve
- Need for heuristics to guide this process

General Math Research Issues

Higher-order discretizations

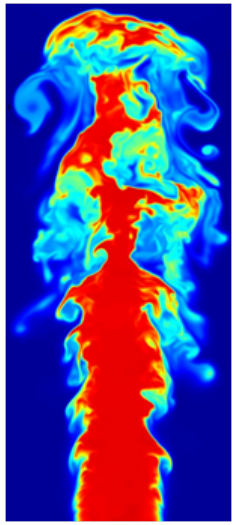
- Reduce number of degrees of freedom
- Increase floating point intensity

Improved coupling algorithms

- Higher-order needs improved coupling strategies for different processes
- Traditional splitting approaches are hard to generalize
- Fully coupled leads to difficult nonlinear systems
- Iterative coupling based on spectral deferred corrections

Multirate integration strategies

- Don't require all processes to use same time step
- Hard to generalize conventional IMEX approaches
- Fits with SDC framework



RNS code

- Multicomponent compressible reacting Navier-Stokes
- Fifth order finite volume
- Fourth-order MLSDC time stepping
- Multirate treatment of chemistry

Summary

Common mathematical ideas and software framework

- General framework for adaptive mesh refinement
- Basic discretizations for various types of PDEs
- Able to model problems from a wide range of applications
- Provides a framework for multiscale simulation
- Isolates “physics” from implementation details

Research issues

- Data distribution and load balancing
- Higher-order discretizations
- Improved coupling strategies
- More general approaches for multiscale systems

Research issues we didn’t discuss

- Embedded UQ
- Incorporating experimental data to improve predictions
- Design / optmization