Block-Structured Adaptive Mesh Refinement

Lecture 4

- Geometry
  - Embedded Boundary
  - Software support embedded boundaries

Approaches to geometry

Curvilinear adaptive grids

Over set grid – generalizes curvilinear

Embedded boundary or Cartesian Grid methods

- Grid generation is tractable – CART3D
- Discretization issues are well-understood away from boundary
- Straightforward coupling to structured AMR

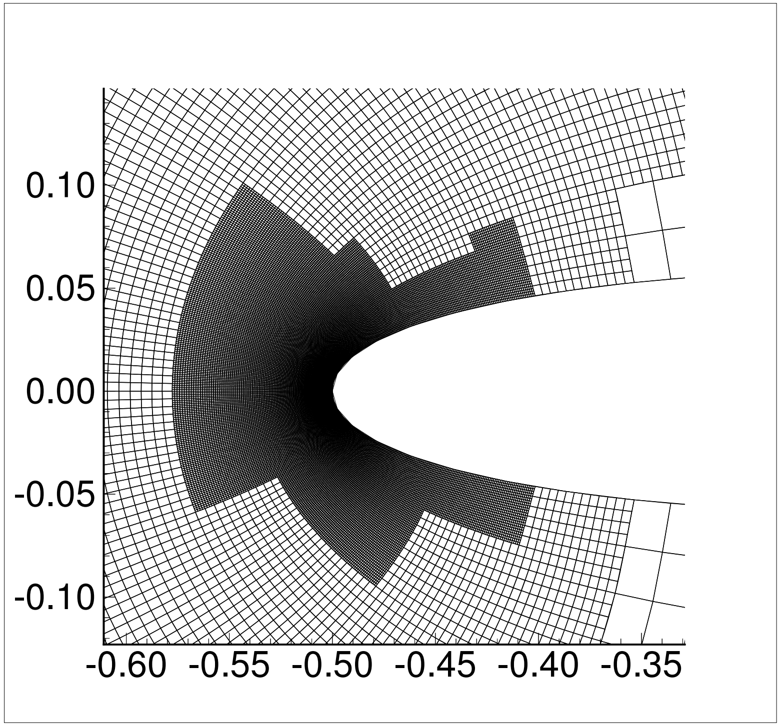
Approaches to geometry

Curvilinear adaptive grids

Over set grid – generalizes curvilinear

Embedded boundary or Cartesian Grid methods

- Grid generation is tractable – CART3D
- Discretization issues are well-understood away from boundary
- Straightforward coupling to structured AMR



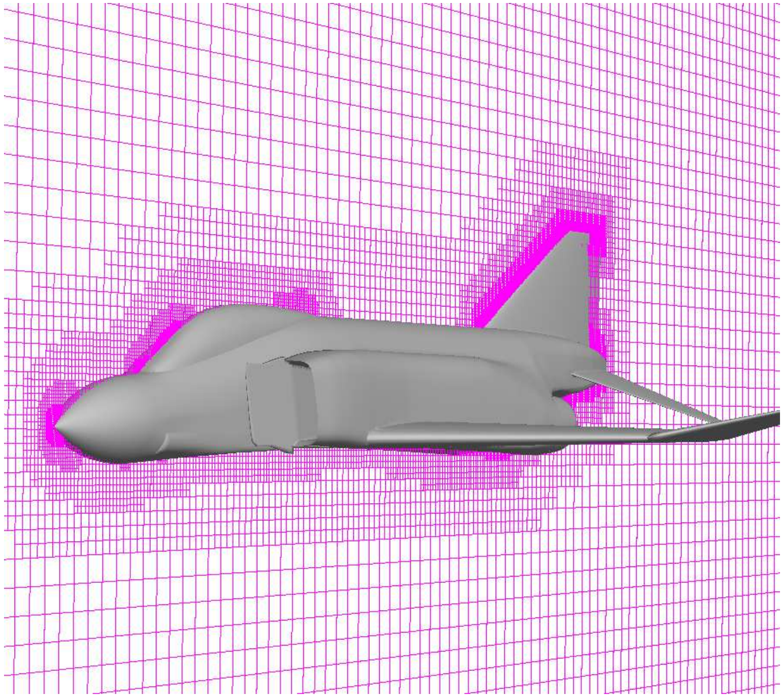
Approaches to geometry

Curvilinear adaptive grids

Over set grid – generalizes curvilinear

Embedded boundary or Cartesian Grid methods

- Grid generation is tractable – CART3D
- Discretization issues are well-understood away from boundary
- Straightforward coupling to structured AMR



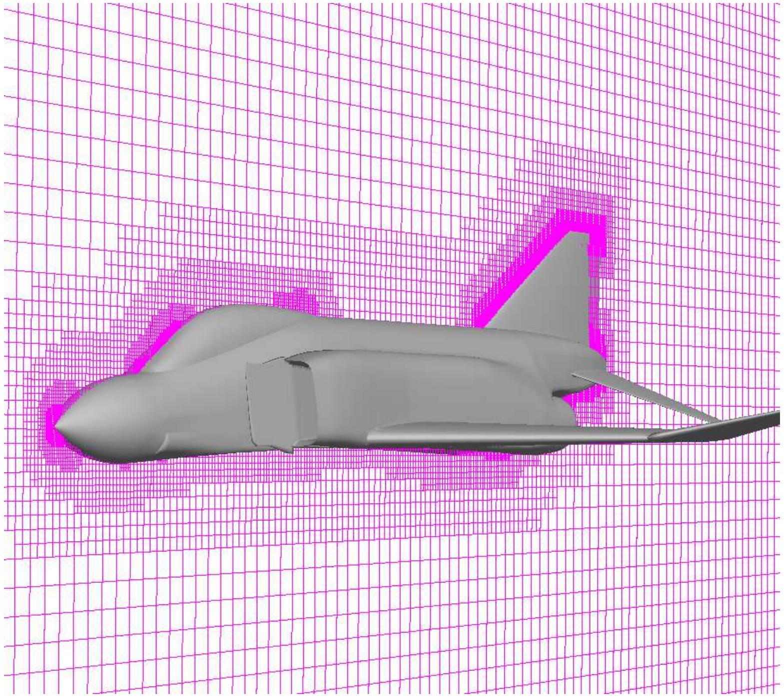
Approaches to geometry

Curvilinear adaptive grids

Over set grid – generalizes curvilinear

Embedded boundary or Cartesian Grid methods

- Grid generation is tractable – CART3D
- Discretization issues are well-understood away from boundary
- Straightforward coupling to structured AMR



References

- Chern and Colella, 1987
- Youngs et al., 1990
- Berger and Leveque, 1991
- Pember et al., 1994
- Johansen and Colella 1998
- Colella et al., to appear

Preliminaries

Primary variables defined at cell centers

Λ_c – Volume fraction of cut cell $\equiv V_c/h^2$

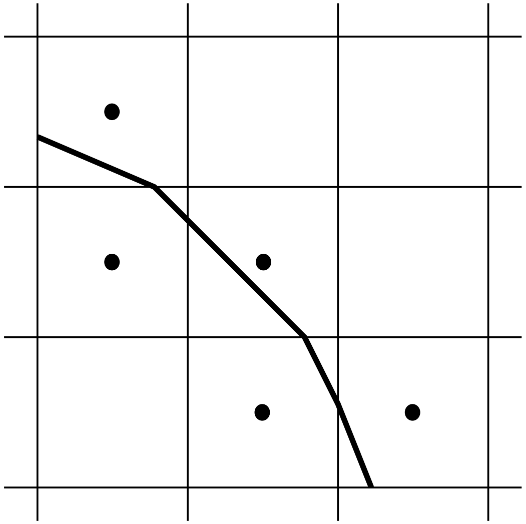
α – aperture $\equiv$ edge length

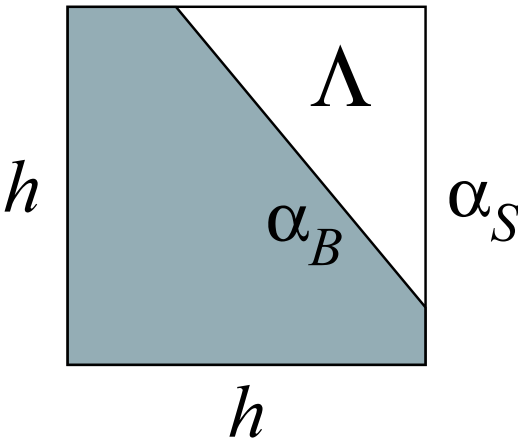
Solve multiphysics applications using EB & AMR

- Develop solvers for classical PDEs
- Decompose applications into component processes

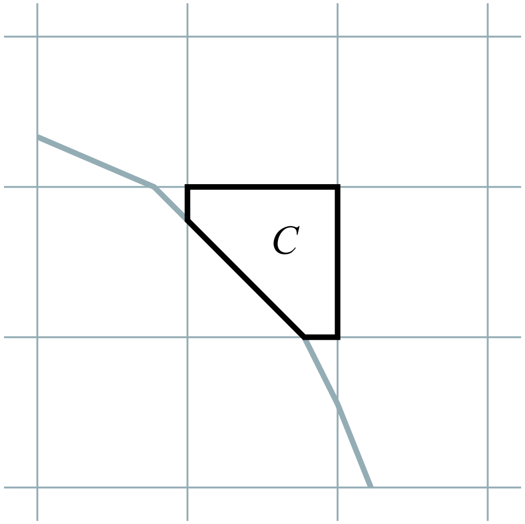
Issues

- Accuracy
- Stability





EB – Conservation Laws



$$U_t + \vec{F}(U) = 0$$

Finite volume discretizaton

$$\int_{t^n}^{t^{n+1}} \int_C U_t + \vec{F}\, dx\, dt = 0$$

$$h^2 \Lambda_c U^{n+1} = h^2 \Lambda_c U^n + \Delta t (\sum_s \alpha_s F_s + \alpha_B F_B)$$

or

$$U^{n+1} = U^n + \frac{\Delta t}{h^2 \Lambda_c} (\sum_s \alpha_s F_s + \alpha_B F_B)$$

where F_s and F_B are explicitly computed fluxes

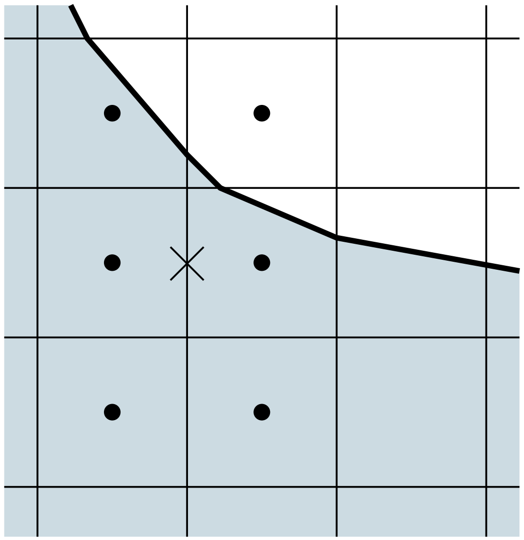
- How to compute fluxes
- How to handle small-cell stability

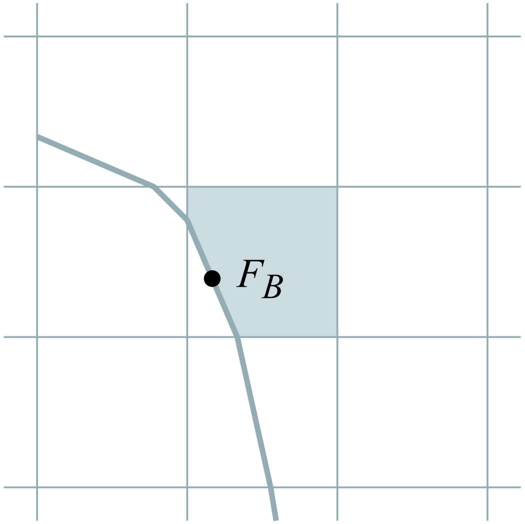
Fluxes – version 1

There are several variations on how to do these things

A simple way to compute fluxes

- Extend state to compute fluxes using Godunov scheme for all edges of a cut cell
  - Volume weighted sum of values in a neighborhood of point
  - Modify Godonov scheme to use "essential" stencil for edges with $\alpha_s = 0$
- F_B computed by solving Riemann problem in local coordinates to boundary





Update

One could update using

$$U^{n+1,cu} = U^n + \frac{\Delta t}{h^2 \Lambda_c} \sum_s \alpha_s F_s + \alpha_B F_B$$

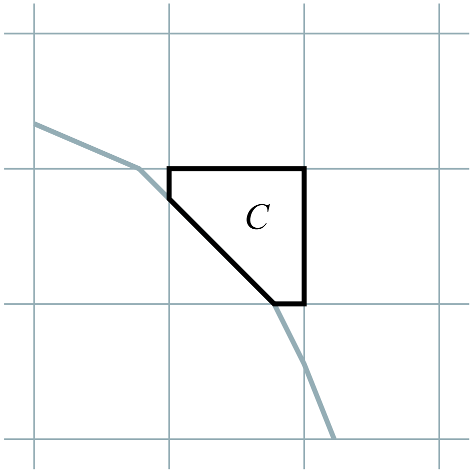
This defines a conservative update but the time step for cut cells decreases as Λ_c decreases.

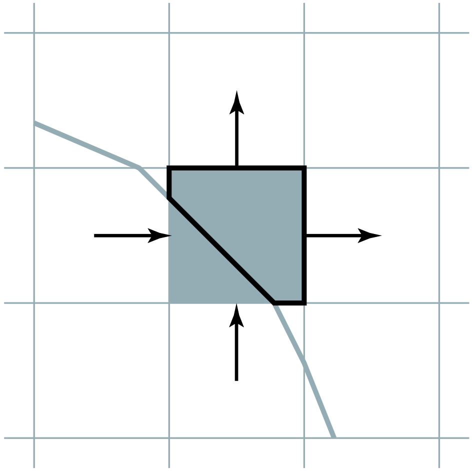
We would like a conservative update that is stable at full-cell CFL

Define a reference state

$$U^{n+1,ref} = U^n + \frac{\Delta t}{h^2} \sum_s F_s$$

which represents update as though there were no boundary in the cut cell





Update cont'd

Define

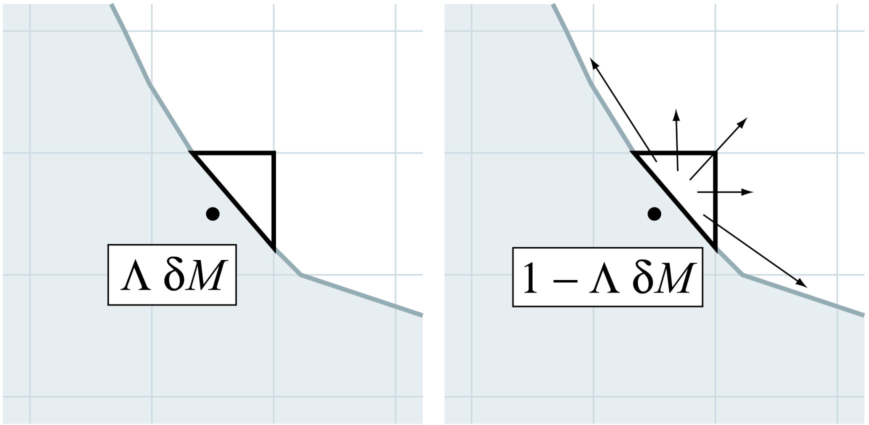
$$\delta M = h^2 \Lambda_c (U^{n+1,cu} - U^{n+1,ref})$$

Compute stable update

$$U^{n+1,p} = U^{n+1,ref} + \frac{\delta M}{h^2}$$

Redistribute $(1 - \Lambda_c)\delta M$ to neighboring cells

- Volume weighted
- Mass weighted (gas dynamics)

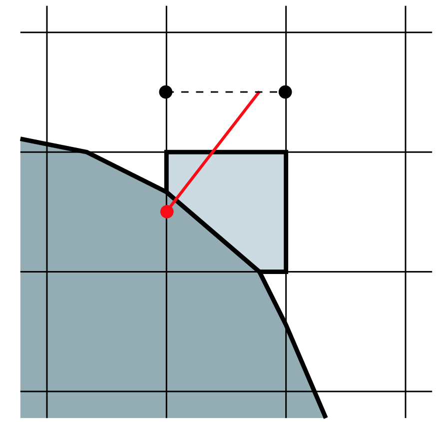


Recover full CFL time step

Enhancements to base algorithm

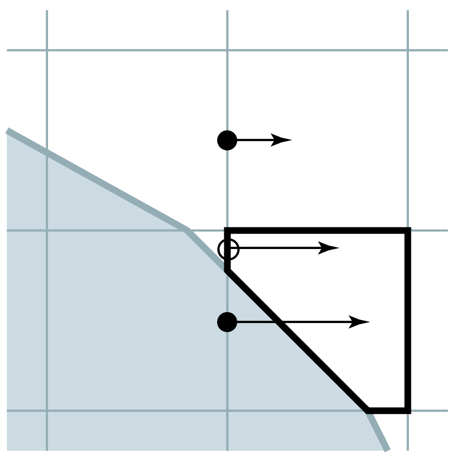
Extended states (Colella et al., to appear)

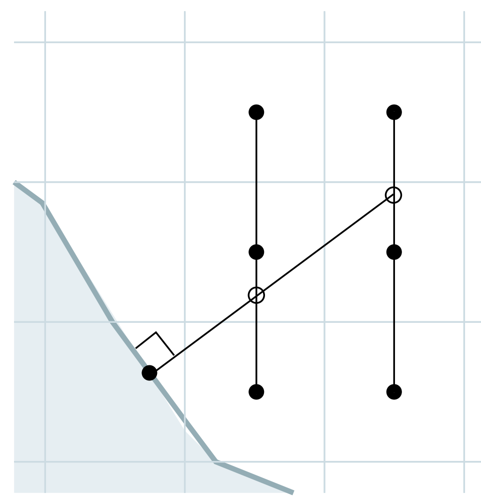
- Extrapolate along normal direction
- Do not use data in adjacent cell



Fluxes (Johansen and Colella, JCP 1998)

- Interpolate fluxed to centroid of edges
- Higher-order boundary flux in normal direction





Analysis

Modified equation

$$\frac{\partial \mathbf{U}^{mod}}{\partial t} + \partial \vec{F}(U^{mod}) = \tau$$

τ localized

- $O(h^2)$ interior
- $O(h/\Lambda)$ at boundary

Error

- $O(h^2)$ is boundary is noncharacteristic
- $O(h)$ in L^∞ and $O(h^2)$ in L^1 if boundary is characteristic

Poisson equation

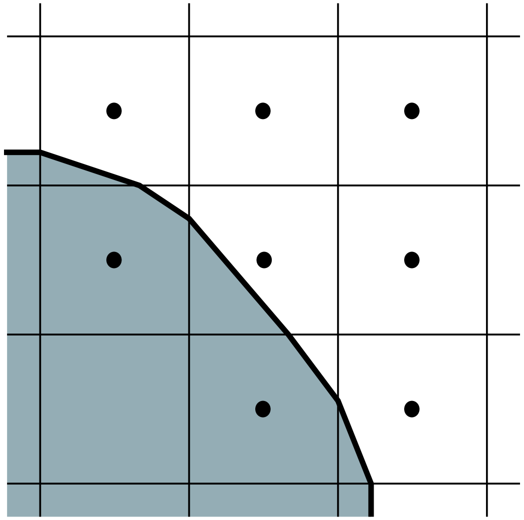
Solve elliptic PDE on embedded boundary

$$\Delta\phi = \rho$$

Want a cell-centered finite volume discretization

$$\nabla \cdot \nabla\phi = \rho$$

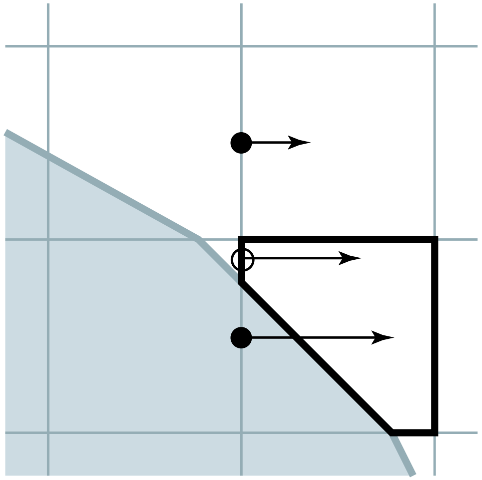
so $\nabla\phi$ acts like a flux



$$\sum_s \alpha_s \frac{\partial\phi}{\partial n}_s + \alpha_B \frac{\partial\phi}{\partial n}_B = \Lambda_c h^2 \rho$$

EB Poisson discretization

Evalute $\partial\phi/\partial n$ using Johansen–Colella flux Leads to well-conditioned linear system with approximately "elliptic" spectral properties

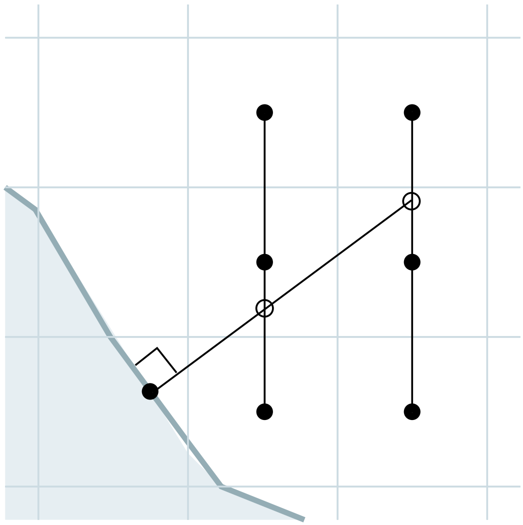


Modified equation gives

$$\Delta\phi_h = \rho + \tau$$

where τ is first-order near boundary and second-order away from boundary

Smoothing property of inverse operator gives error, $\phi - \phi_h = \Delta^{-1}\tau = O(h^2)$

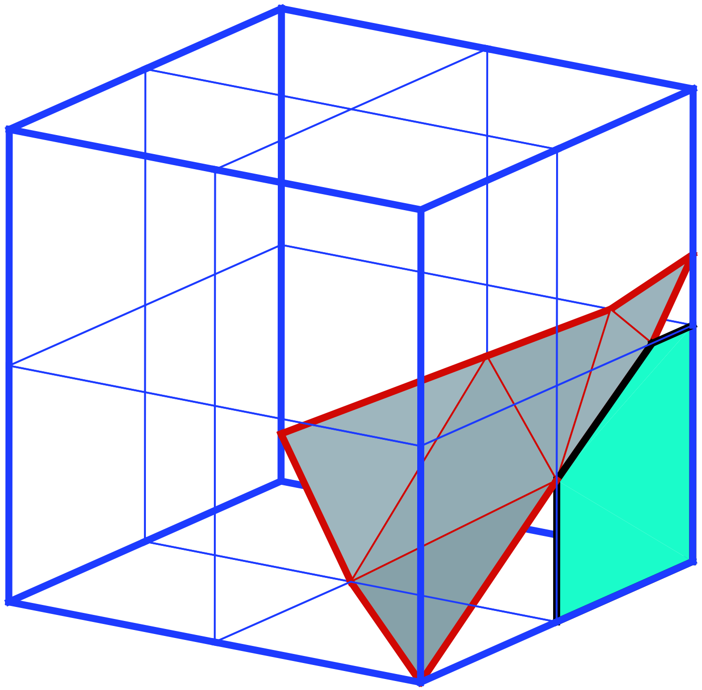


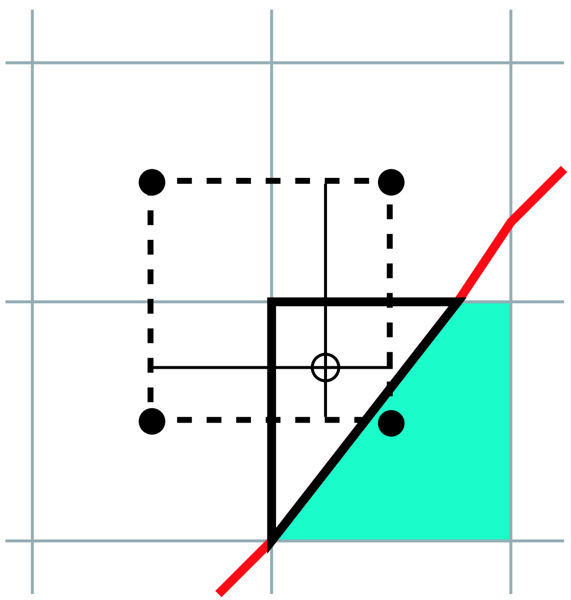
However the matrix is not

- Symmetric
- M-Matrix

Extension to three dimensions

Two possible approaches to extend Johansen–Colella flux to three dimension

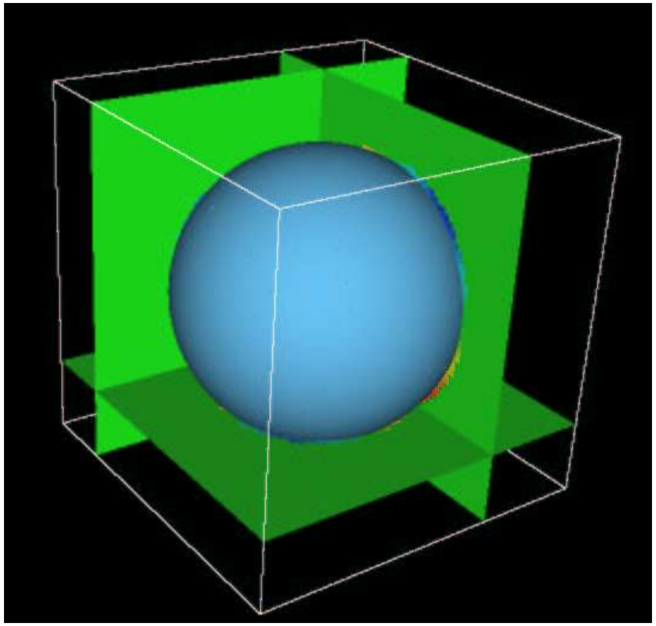


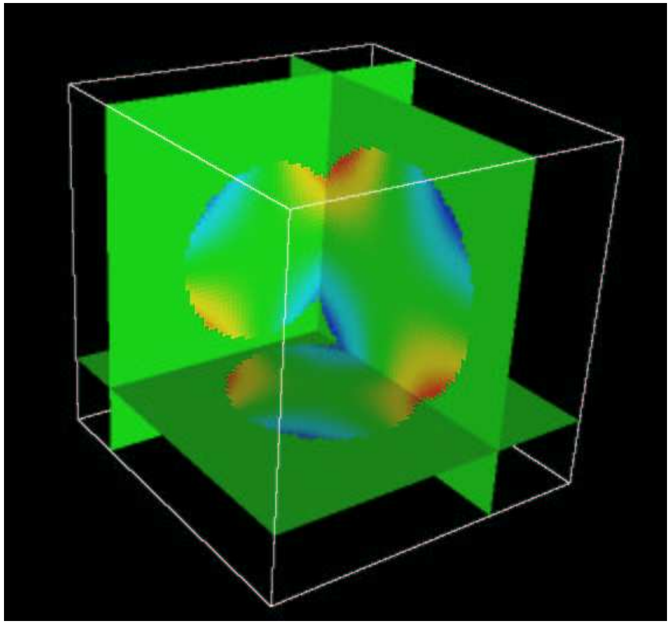


Linear interpolation is unstable; but, bilinear is stable

Poisson solution error – 3D

grid	$\|\epsilon\|_\infty$	p_∞	$\|\epsilon\|_2$	p_2	$\|\epsilon\|_1$	p_1
16^3	4.80×10^{-4}	—	5.17×10^{-5}	—	1.83×10^{-5}	—
32^3	1.06×10^{-4}	2.17	1.25×10^{-5}	2.05	4.41×10^{-6}	2.05
64^3	2.43×10^{-5}	2.13	3.07×10^{-6}	2.02	1.09×10^{-6}	2.02

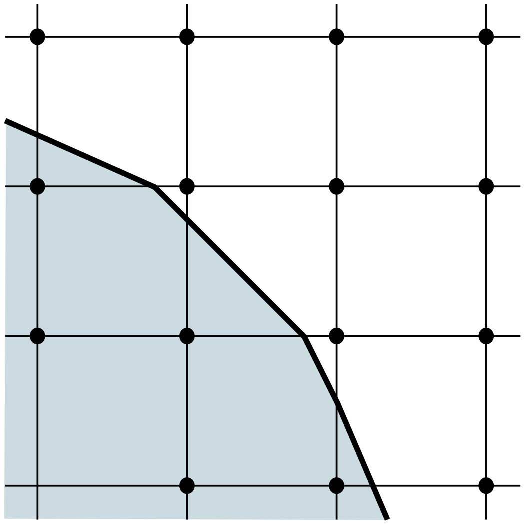




Nodal Projection

Projection performs the decomposition

$$V = U_d + \nabla\phi$$



For cut cells, view as extension of finite element basis extended to cover all of the cut cell

Projection uses homogeneous Neumann boundary conditions at cut cell boundaries

Gives a weak form

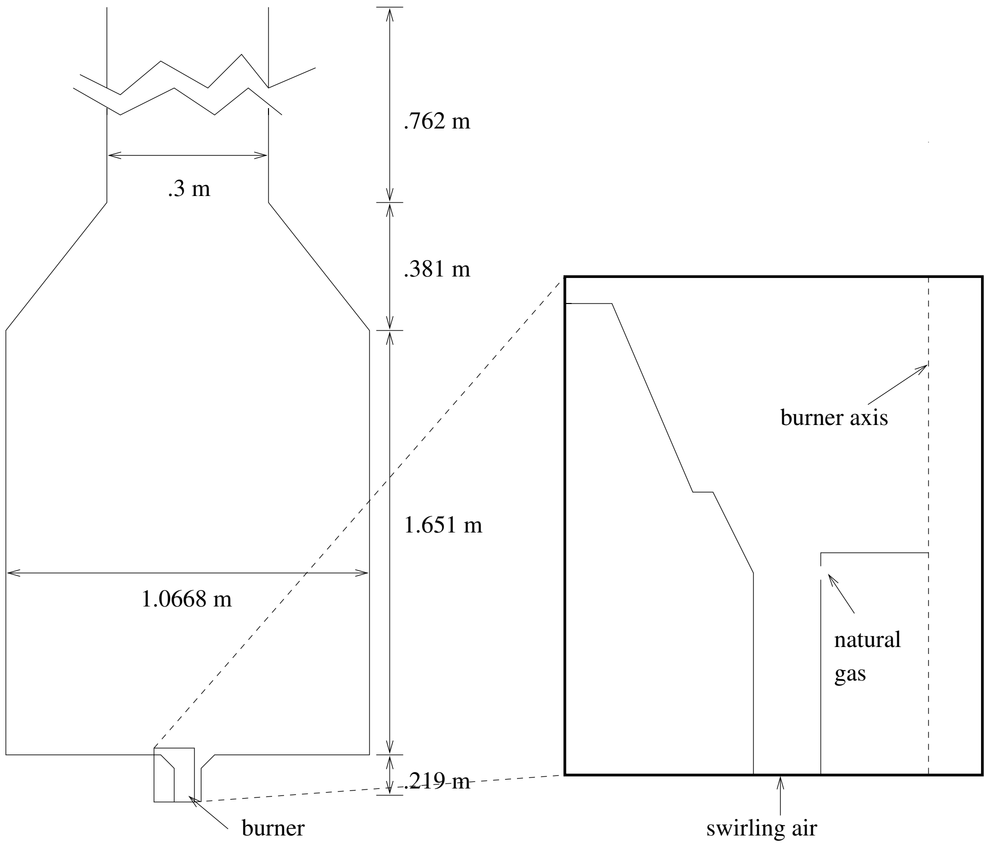
$$\int_\Omega \nabla\phi \cdot \nabla\chi \, dx = \int_\Omega V \cdot \nabla\chi \, dx$$

Youngs et al. – Full potential adaptive transonic flow solver

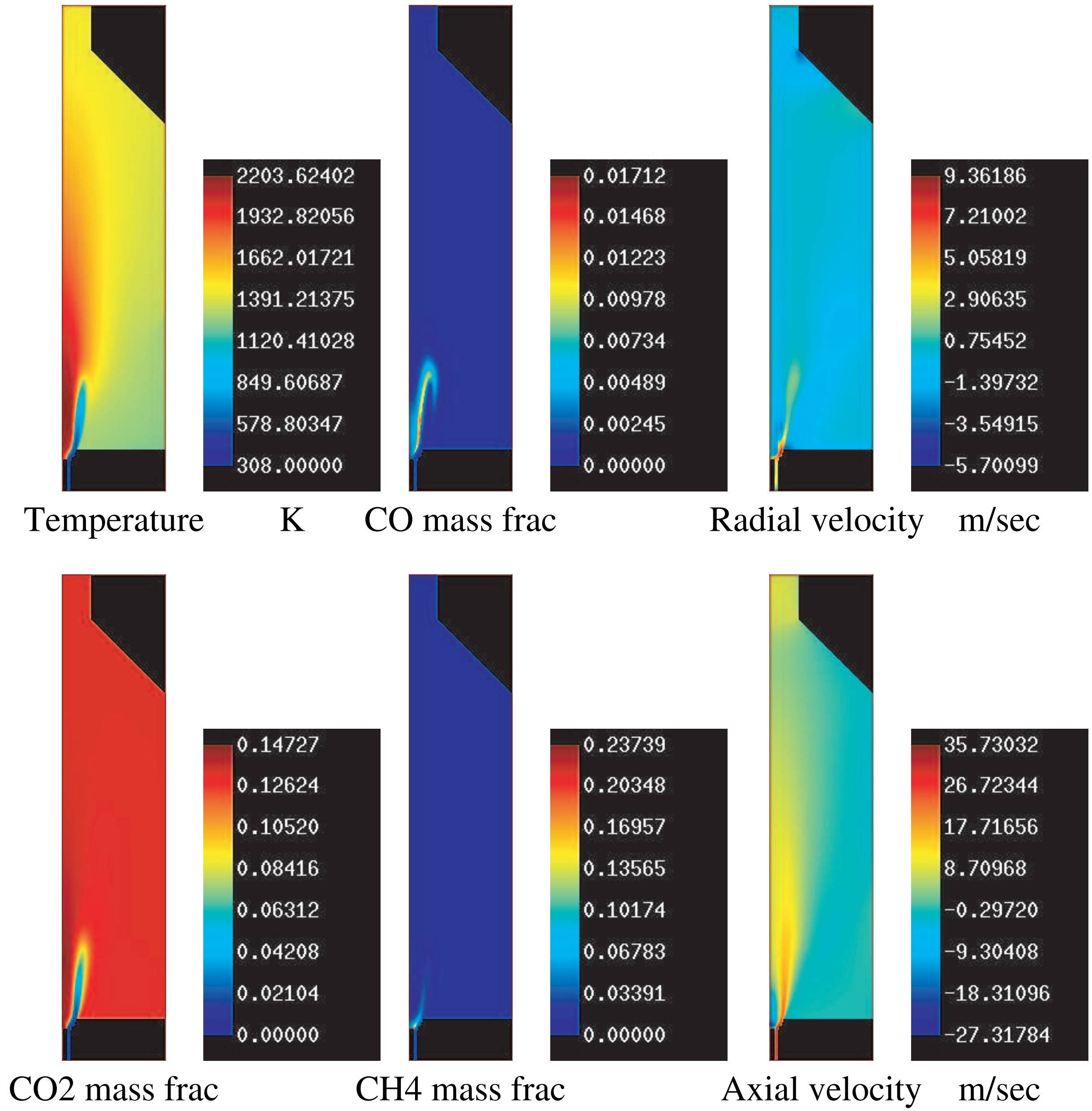
Multiphysics application

Industrial burner

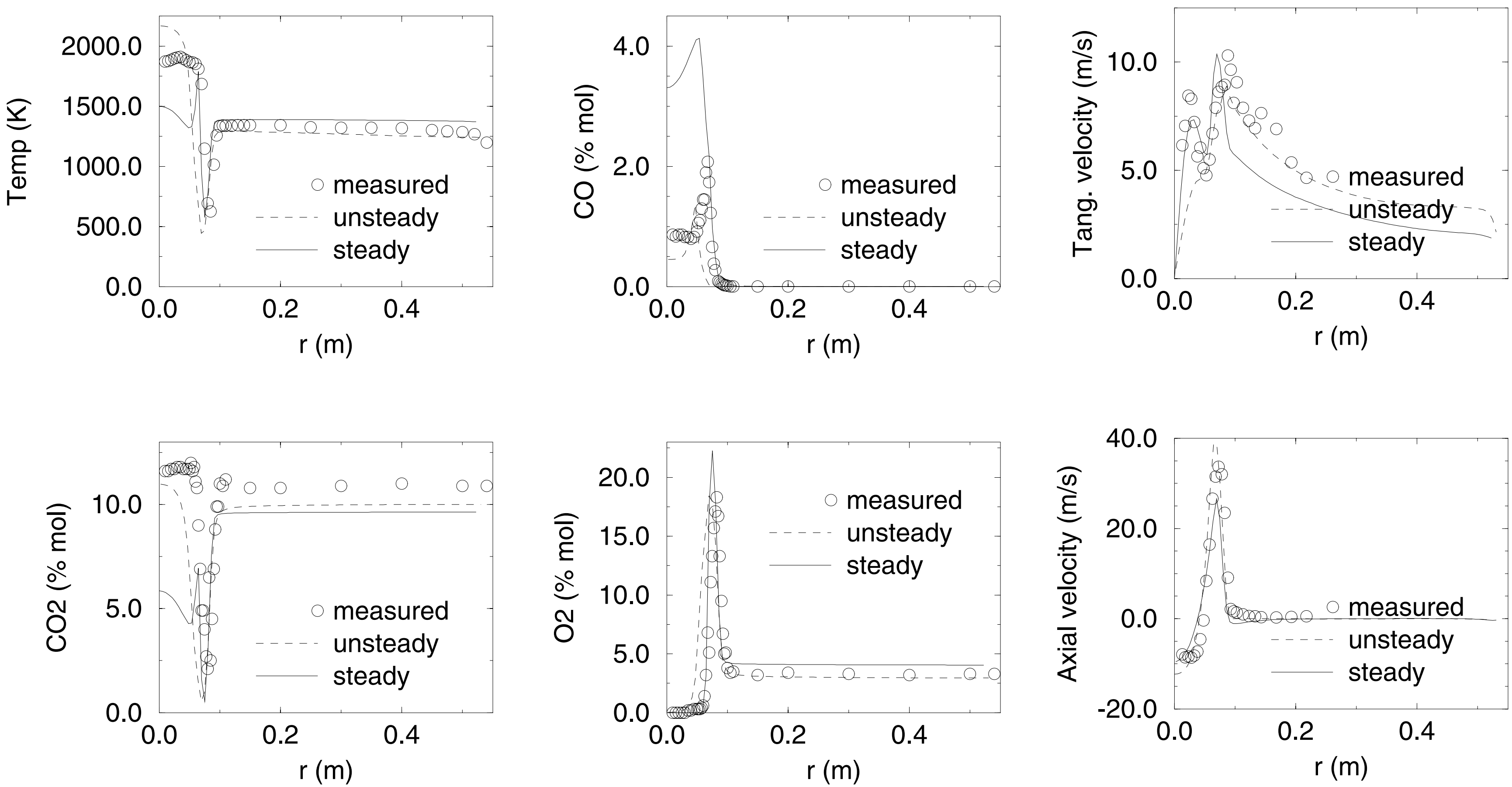
- Low Mach number combustion formuation
- Axisymmetric flow
- $k - \epsilon$ turbulence model
  - Law of the wall
- Discrete ordinates radiation



Burner simulation results



Burner experimental comparisons



AMR considerations

Embedded boundary + structured AMR is basically straightforward

- If coarse / fine boundaries aren't near the embedded boundary there is basically nothing to do

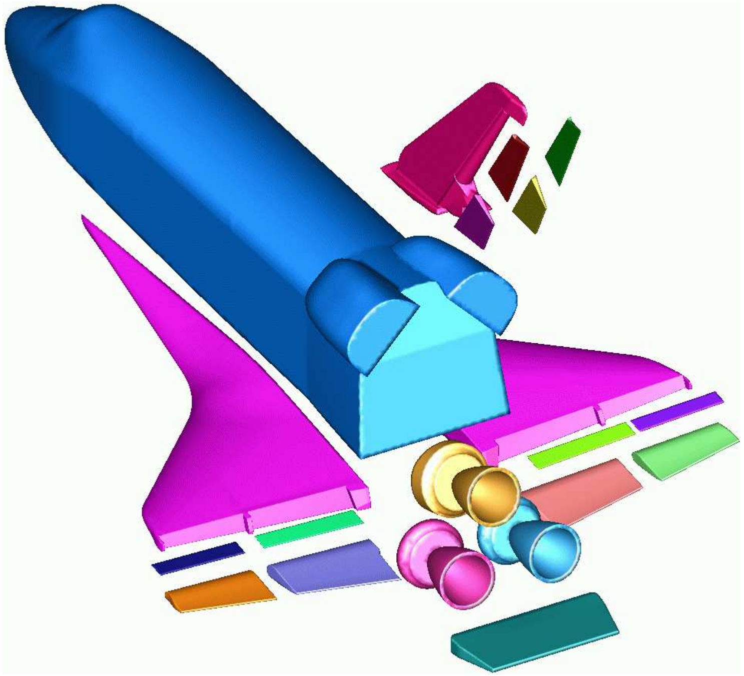
When coarse / fine boundaries intersect cut cells

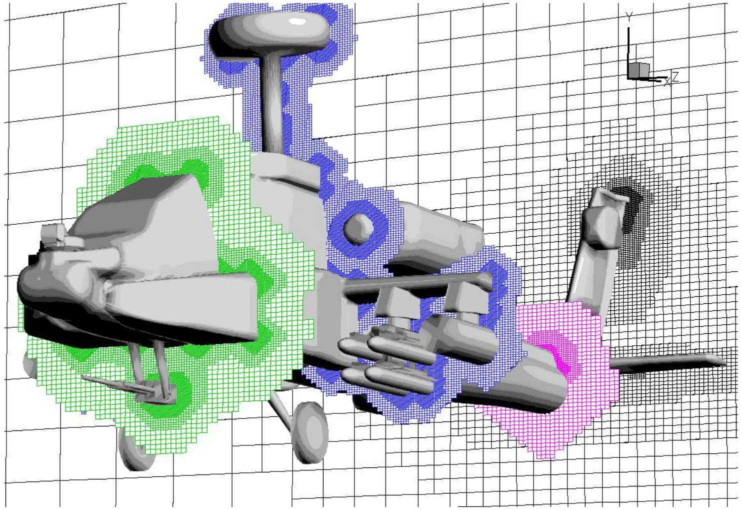
- Modify hyperbolic redistribution
  - Follows basic AMR design principles
  - Keep track of redistributions across coarse / fine boundary
  - Adjust data to correct errors (analogous to reflux)
- Modify Johansen – Colella flux formulae
  - Drop to first-order for hyperbolic if necessary
  - Use first-order least squares fit to define boundary flux for elliptic
  - Since these modifications are localized to a co-dimension 2 subset of the domain they do not effect accuracy

Embedded Boundary Software

Grid generation software – Cart3D

- Component based approach
- Fix-up triangulations
- Generate cut cell information
- http://people.nas.nasa.gov/ aftosmis/cart3d/cart3Dhome.html





Packages supporting EB discretizations

EBChombo – LBNL

BEARCLAW – Univ. of Washington and Univ. of North Carolina

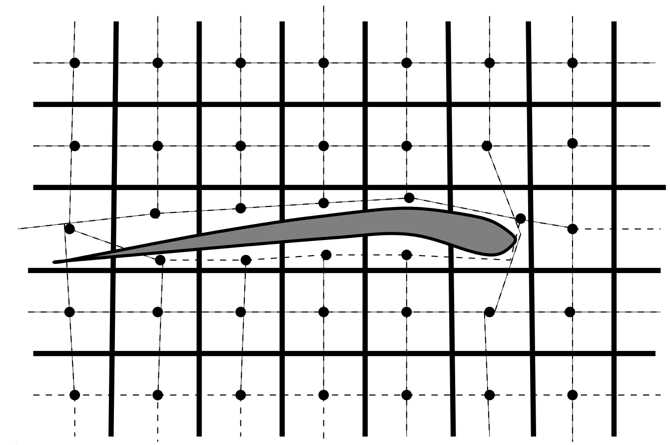
CART3D – NASA Ames

It is beyond the scope of this lecture to discuss EB software issues in detail

We can examine the analogs of some of the data structures discussed before

EB Software Design – EBChombo

We generalize rectangular array abstractions to represent more general general graphs that map into the rectangular lattice $\mathbb{Z}^D$. The nodes of the graph are the control volumes, while the arcs of the graph are the faces across which fluxes are defined.



	BoxLib	EB Chombo
Z^D	–	`EBIndexSpace`
Index	`IntVect`	`VolIndex, FaceIndex`
Region of Z^D	`Box`	`EBISBox`
Union of rectangles	`BoxArray`	`EBISLayout`
Rectangular array	`Fab`	`EBCellFAB, EBFaceFAB`
Looping construct	`FabIterator`	`VoFIterator, FaceIterator`