Hyperbolic Conservation Laws
And
Visualization and Data Analysis
In Chombo

Terry J. Ligocki
tjligocki@lbl.gov

Applied Numerical Algorithm Group
Lawrence Berkeley National Laboratory
Berkeley, CA, USA

IPAM - UCLA
March 28, 2005

Overview

- **Hyperbolic Conservation Laws**
  - Introduction
  - Examples
  - Discretization
  - Algorithm
  - Implementation
  - Additional Notes
- **Visualization and Data Analysis**
  - Introduction
  - Design/Architecture
  - Capabilities (Demonstration and Movies)
  - Features
- **Remarks**
  - Software Availability
  - Acknowledgments

Hyperbolic Conservation Laws - Introduction

- Hyperbolic Conservation Laws can be written in the form:

$$\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{t}} + \boldsymbol{\nabla} \cdot \vec{\boldsymbol{F}}(\boldsymbol{U}) = \boldsymbol{S}$$

- More explicit form:

$$\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{t}} + \sum_{\boldsymbol{d}=0}^{\mathbf{D}-1} \frac{\partial \boldsymbol{F}^{\boldsymbol{d}}(\boldsymbol{U})}{\partial \boldsymbol{x}^{\boldsymbol{d}}} = \boldsymbol{S}$$

- Changing to primitive variables, $\boldsymbol{W} = \boldsymbol{W}(\boldsymbol{U})$:

$$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{t}} + \sum_{\boldsymbol{d}=0}^{\mathbf{D}-1} \boldsymbol{A}^{\boldsymbol{d}}(\boldsymbol{W}) \frac{\partial \boldsymbol{W}^{\boldsymbol{d}}}{\partial \boldsymbol{x}^{\boldsymbol{d}}} = \boldsymbol{S}'$$

$$\boldsymbol{A}^{\boldsymbol{d}} = \boldsymbol{\nabla}_{\boldsymbol{U}} \boldsymbol{W} \cdot \boldsymbol{\nabla}_{\boldsymbol{U}} \boldsymbol{F}^{\boldsymbol{d}} \cdot \boldsymbol{\nabla}_{\boldsymbol{W}} \boldsymbol{U}$$

$$\boldsymbol{S}' = \boldsymbol{\nabla}_{\boldsymbol{U}} \boldsymbol{W} \cdot \boldsymbol{S}$$

Hyperbolic Conservation Laws - Examples

- 2D Gas Dynamics (Compressible Euler Equations):

$$\begin{aligned}
U &= (\rho, \rho u_1, \rho u_2, \rho E) \\
F^1 &= (\rho u_1, \rho u_1^2 + p, \rho u_1 u_2, \rho u_1 E + u_1 p) \\
F^2 &= (\rho u_2, \rho u_1 u_2, \rho u_2^2 + p, \rho u_2 E + u_2 p) \\
S &= 0 \\
W &= (\rho, u_1, u_2, E)
\end{aligned}$$

$$\begin{aligned}
p &= (\gamma - 1)\rho e \\
e &= (E - \frac{1}{2}(u_1^2 + u_2^2))
\end{aligned}$$

Hyperbolic Conservation Laws - Examples

- Ideal MHD:

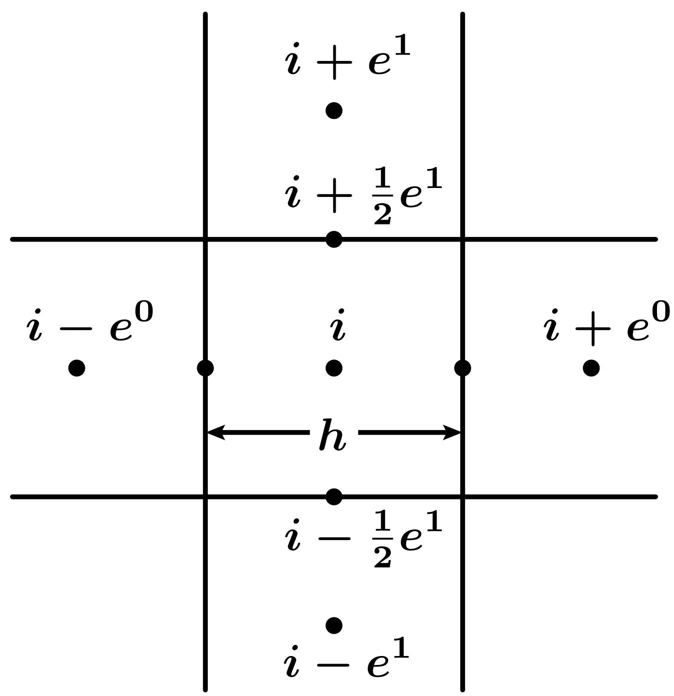
$$
\begin{aligned}
U &= (\rho, \rho\vec{u}, \vec{B}, \rho E) \\
F &= (\rho\vec{u}, \\
&\qquad \rho\vec{u}\vec{u} + (P + \tfrac{1}{8\pi}|\vec{B}|^2)I - \tfrac{1}{4\pi}\vec{B}\vec{B}, \\
&\qquad \vec{u}\vec{B} - \vec{B}\vec{u}, \\
&\qquad (\rho E + P + \tfrac{1}{8\pi}|\vec{B}|^2)\vec{u} - \tfrac{1}{4\pi}(\vec{u}\cdot\vec{B})\vec{B}) \\
S &= 0 \\
W &= (\rho, \vec{u}, \vec{B}, E)
\end{aligned}
$$

$$
\rho E = (\frac{1}{2}\rho|\vec{u}|^2 + \tfrac{1}{8\pi}|\vec{B}|^2 + \tfrac{1}{\gamma-1}P)
$$

$$
\nabla\cdot\vec{B} = 0
$$

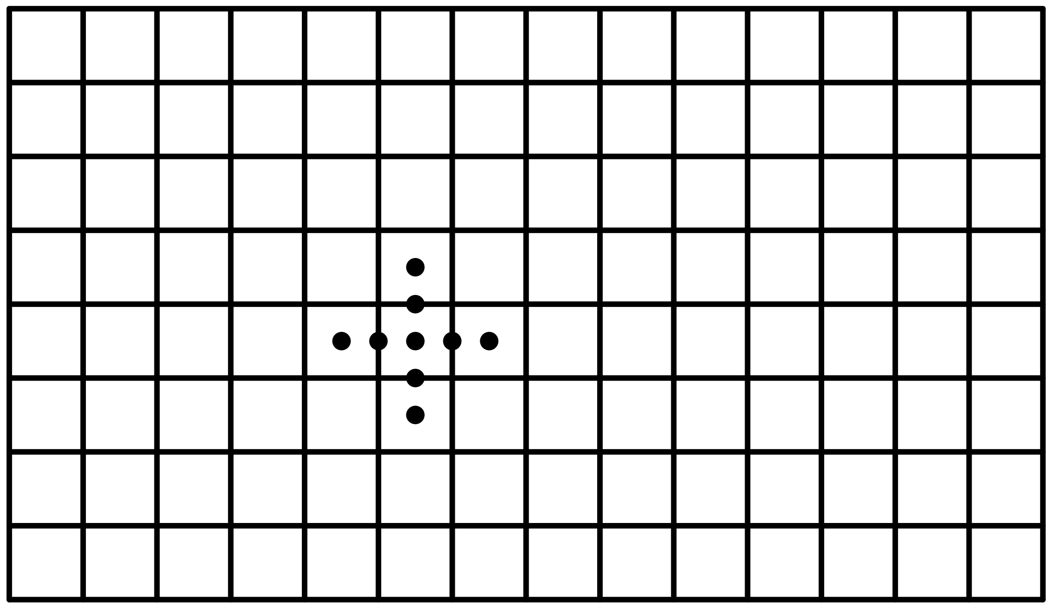
Hyperbolic Conservation Laws - Discretization

- Notation and indexing: i is a spatial index and n is a time index:

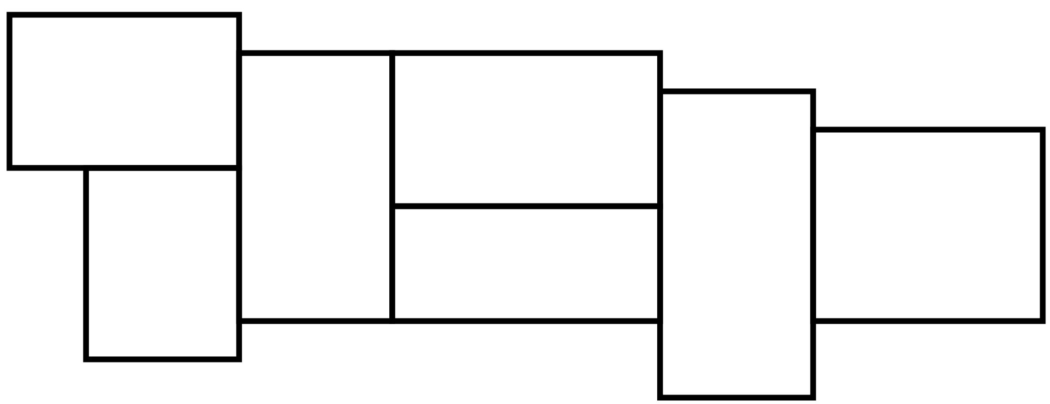


- The spatial index and the time index are related to physical coordinates via h and Δt, respectively

- Cells are grouped into boxes:

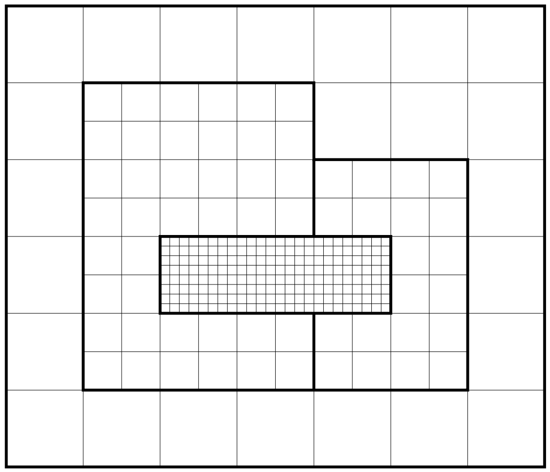


- Boxes are grouped into levels:

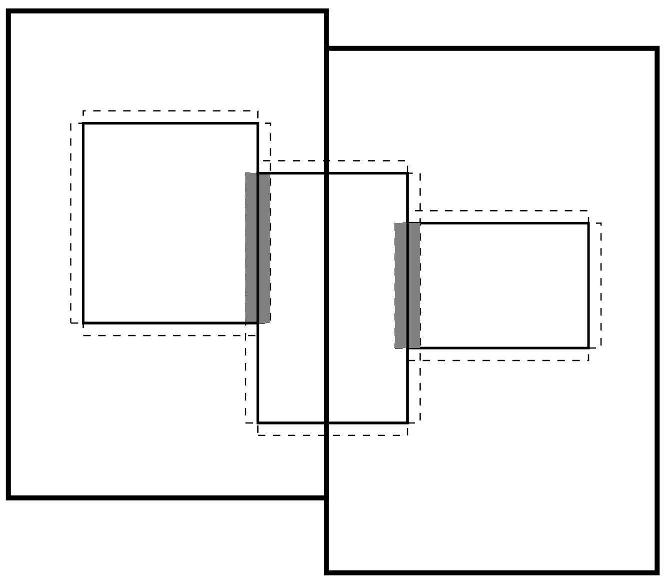


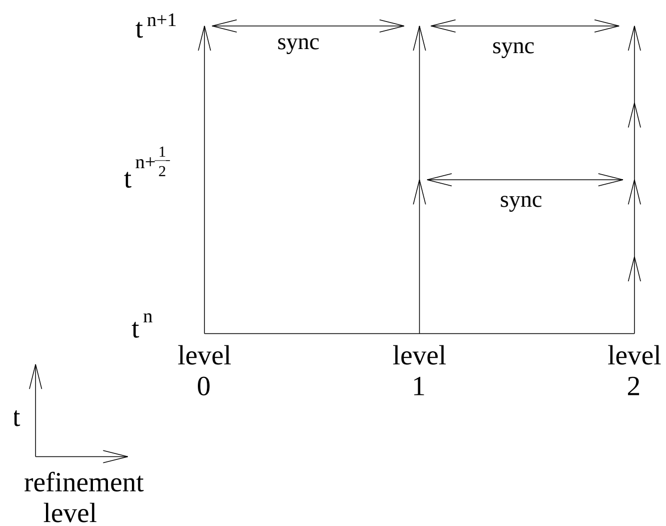
Hyperbolic Conservation Laws - Discretization

- Levels at different resolutions are nested:



- This nesting allows the coarser level to define the boundary conditions for the finer level:

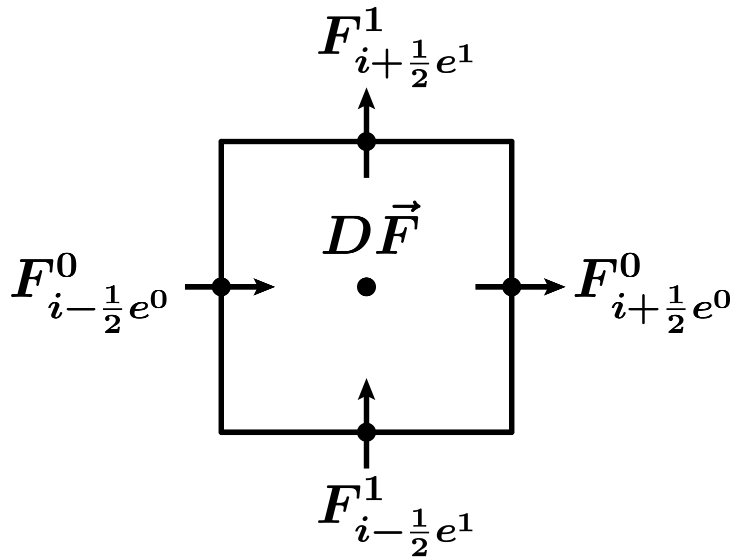




Hyperbolic Conservation Laws - Discretization

- Consider a single level (collection of boxes) at a fixed resolution
- Approximate the divergence of the flux in each cell of each box:

$$\nabla \cdot \vec{F} \approx D\vec{F} \equiv \frac{1}{h} \sum_{d=0}^{D-1} (F^d_{i+\frac{1}{2}e^d} - F^d_{i-\frac{1}{2}e^d})$$



- This is exact if $\nabla \cdot \vec{F}$ was a cell average and the $F^d_{i\pm\frac{1}{2}e^d}$ were face averages (divergence theorem)

- Second-order accurate in space if fluxes are second-order accurate
- Update the solution:

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^{n} - \boldsymbol{\Delta t}(\boldsymbol{D}\vec{\boldsymbol{F}})\ ,\ \vec{\boldsymbol{F}} = \vec{\boldsymbol{F}}(\boldsymbol{U}^{n})$$

- The critical element is the accurate computation of $\boldsymbol{F}^{d}$ in space and time
- Second-order accuracy in time is achieved by using a predictor-corrector method

Hyperbolic Conservation Laws - Algorithm

Given $\boldsymbol{U}_i^n$ and $\boldsymbol{S}_i^n$, we want to compute a second-order accurate estimate of the fluxes:

$$\boldsymbol{F}^{n+\frac{1}{2}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \approx \boldsymbol{F}^d(\boldsymbol{x}_0 + (\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d)\boldsymbol{h}, \boldsymbol{t}^n + \frac{1}{2}\boldsymbol{\Delta t})$$

1. Compute the effect of the normal derivative terms and the source term on the extrapolation in space and time from cell centers to faces. For $\mathbf{0} \leq \boldsymbol{d} < \mathrm{D}$:

$$\boldsymbol{W}_{\boldsymbol{i},\pm,d} = \boldsymbol{W}_{\boldsymbol{i}}^n + \frac{1}{2}(\pm \boldsymbol{I} - \frac{\boldsymbol{\Delta t}}{\boldsymbol{h}}\boldsymbol{A}_{\boldsymbol{i}}^d)\boldsymbol{P}_\pm(\boldsymbol{\Delta}^d \boldsymbol{W}_{\boldsymbol{i}})$$

$$\boldsymbol{A}_{\boldsymbol{i}}^d = \boldsymbol{A}^d(\boldsymbol{W}_{\boldsymbol{i}})$$

$$\boldsymbol{P}_\pm(\boldsymbol{W}) = \sum_{\pm \boldsymbol{\lambda}_k > 0} (\boldsymbol{l}_k \cdot \boldsymbol{W})\boldsymbol{r}_k$$

$$\boldsymbol{W}_{\boldsymbol{i},\pm,d} = \boldsymbol{W}_{\boldsymbol{i},\pm,d} + \frac{\boldsymbol{\Delta t}}{2}\boldsymbol{\nabla}_U \boldsymbol{W} \cdot \boldsymbol{S}_{\boldsymbol{i}}^n$$

where $\boldsymbol{\lambda_k}$ are eigenvalues of $\boldsymbol{A_i^d}$, and $\boldsymbol{l_k}$ and $\boldsymbol{r_k}$ are the corresponding left and right eigenvectors.

2. Compute estimates of $\boldsymbol{F^d}$ suitable for computing 1D flux derivatives $\frac{\partial \boldsymbol{F^d}}{\partial \boldsymbol{x^d}}$ using a Riemann solver for the interior, $\boldsymbol{R}$, and for the boundary, $\boldsymbol{R_B}$. Here, and in what follows, $\boldsymbol{\nabla_U W}$ need only be first-order accurate, e.g., differ from the value at $\boldsymbol{U_i^n}$ by $\boldsymbol{O(h)}$:

$$\begin{aligned} \boldsymbol{F}^{\mathrm{1D}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e^d}} = & \; \boldsymbol{R}(\boldsymbol{W_{i,+,d}}, \boldsymbol{W_{i+e^d,-,d}}, \boldsymbol{d}) \\ & \mid \; \boldsymbol{R_B}(\boldsymbol{W_{i,+,d}}, (\boldsymbol{i} + \frac{1}{2}\boldsymbol{e^d})\boldsymbol{h}, \boldsymbol{d}) \\ & \mid \; \boldsymbol{R_B}(\boldsymbol{W_{i+e^d,-,d}}, (\boldsymbol{i} + \frac{1}{2}\boldsymbol{e^d})\boldsymbol{h}, \boldsymbol{d}) \end{aligned}$$

3. In 3D compute corrections to $\boldsymbol{W_{i,\pm,d}}$ corresponding to one set of transverse derivatives appropriate to obtain $(\boldsymbol{1}, \boldsymbol{1}, \boldsymbol{1})$

diagonal coupling. In 2D skip this step:

$$\boldsymbol{W}_{i,\pm,d_1,d_2} = \boldsymbol{W}_{i,\pm,d_1} - \frac{\Delta t}{3h}\boldsymbol{\nabla}_U \boldsymbol{W} \cdot (\boldsymbol{F}^{\mathrm{1D}}_{i+\frac{1}{2}e^{d_2}} - \boldsymbol{F}^{\mathrm{1D}}_{i-\frac{1}{2}e^{d_2}})$$

4. In 3D compute fluxes corresponding to corrections made in the previous step. In 2D skip this step:

$$\begin{aligned} \boldsymbol{F}_{i+\frac{1}{2}e^{d_1},d_2} = {} & \boldsymbol{R}(\boldsymbol{W}_{i,+,d_1,d_2}, \boldsymbol{W}_{i+e^{d_1},-,d_1,d_2}, d_1) \\ & \mid \ \boldsymbol{R}_B(\boldsymbol{W}_{i,+,d_1,d_2}, (i + \frac{1}{2}e^{d_1})h, d_1) \\ & \mid \ \boldsymbol{R}_B(\boldsymbol{W}_{i+e^{d_1},-,d_1,d_2}, (i + \frac{1}{2}e^{d_1})h, d_1) \end{aligned}$$

5. Compute final corrections to $\boldsymbol{W}_{i,\pm,d}$ due to the final transverse

derivatives:

$$\text{2D:} \quad \boldsymbol{W}^{n+\frac{1}{2}}_{i,\pm,d} = \boldsymbol{W}_{i,\pm,d} - \frac{\Delta t}{2h} \nabla_U \boldsymbol{W} \cdot (\boldsymbol{F}^{\text{1D}}_{i+\frac{1}{2}e^{d_1}} - \boldsymbol{F}^{\text{1D}}_{i-\frac{1}{2}e^{d_1}})$$

$$\text{3D:} \quad \boldsymbol{W}^{n+\frac{1}{2}}_{i,\pm,d} = \boldsymbol{W}_{i,\pm,d} - \frac{\Delta t}{2h} \nabla_U \boldsymbol{W} \cdot (\boldsymbol{F}_{i+\frac{1}{2}e^{d_1},d_2} - \boldsymbol{F}_{i-\frac{1}{2}e^{d_1},d_2}) - \frac{\Delta t}{2h} \nabla_U \boldsymbol{W} \cdot (\boldsymbol{F}_{i+\frac{1}{2}e^{d_2},d_1} - \boldsymbol{F}_{i-\frac{1}{2}e^{d_2},d_1})$$

6. Compute final estimate of fluxes:

$$\boldsymbol{F}^{n+\frac{1}{2}}_{i+\frac{1}{2}e^d} = \boldsymbol{R}(\boldsymbol{W}^{n+\frac{1}{2}}_{i,+,d}, \boldsymbol{W}^{n+\frac{1}{2}}_{i+e^d,-,d}, \boldsymbol{d}) \;|\; \boldsymbol{R}_B(\boldsymbol{W}^{n+\frac{1}{2}}_{i,+,d}, (\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d)\boldsymbol{h}, \boldsymbol{d}) \;|\; \boldsymbol{R}_B(\boldsymbol{W}^{n+\frac{1}{2}}_{i+e^d,-,d}, (\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d)\boldsymbol{h}, \boldsymbol{d})$$

7. Update the solution using the divergence of the fluxes:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t}{h} \sum_{d=0}^{\mathrm{D}-1} (\boldsymbol{F}_{i+\frac{1}{2}e^d}^{n+\frac{1}{2}} - \boldsymbol{F}_{i-\frac{1}{2}e^d}^{n+\frac{1}{2}})$$

- Fourth order slope calculations with limiting and flattening
- Extensions to piecewise parabolic methods (PPM)
- Second-order accurate in space and time
- "Accurate" shock capture - robust and stable
- This is an "unsplit" algorithm for the updating of the conservative quantities, $\boldsymbol{U}$
- Everything has been reduced to computations that can be computed box by box (if ghost cells are used) and all reduced to 1D

Hyperbolic Conservation Laws - Implementation

- All physics independent code has been implemented and requires no modification by the user:
  - The framework for time dependent, adaptive mesh refinement (AMR) computations, including: AMR mesh generation, time step control, interaction between levels
  - All the computations for hyperbolic conservation laws with the exception of a handful of physics dependent routines
  - Parallel computation without modifications to code - only recompilation

Hyperbolic Conservation Laws - Implementation

- Recall Step 1 of the algorithm:

$$W_{i,\pm,d} = W_i^n + \frac{1}{2}(\pm I - \frac{\Delta t}{h} A_i^d) P_\pm(\Delta^d W_i)$$
$$A_i^d = (\nabla_U W)_i \cdot \nabla_U F_i^d \cdot (\nabla_W U)_i$$

- The following physics dependent routines must be provided by the user:
  - Eigen-analysis of the linearization of $A^d(W)$: transformations between characteristic variables (eigenvectors) and primitive variables, computation of eigenvalues
  - The solution to 1D Riemann problems given the primitive variable values on each side of a face
  - Quasilinear update - computation of: $A^d(W) P_\pm(\Delta^d W)/h$
  - Maximum wave speed (in a box) given the conserved

variable values (in the box)

- The transformation of conserved variables to primitive variables
- The computation of fluxes on a face given the value of the primitive variables on the face
- Physical boundary conditions - if the boundaries of the domain are periodic then this is trivial to provide
- Various bookkeeping functions - number of conserved variables, number of primitive variables, etc.

Hyperbolic Conservation Laws - Additional Notes

- Some current work using Chombo's framework:
  - Gas Dynamics - Current example in Chombo library (PLM and PPM)
  - Ideal MHD - Ravi Samtaney (PPPL/ANAG), Rob Crockett (UCB Physics)
  - Self Gravitating Gas Dynamics with MHD and coupling to collisionless particles - Francesco Miniati (ETH)
- Current development:
  - Particle computations
  - Multifluid computations

Visualization and Data Analysis - Introduction

- ChomboVis - visualization and data analysis tool for AMR data
- Some capabilities:
  - Grid display
  - Data slices
  - Contours / Isosurfaces
  - Streamlines
  - Clipping
  - Data selection and spreadsheets
  - State saving and restoring
  - Creation of derived quantities
- Driven by user's needs and funding
- One fulltime developer

Visualization and Data Analysis - Design/Architecture

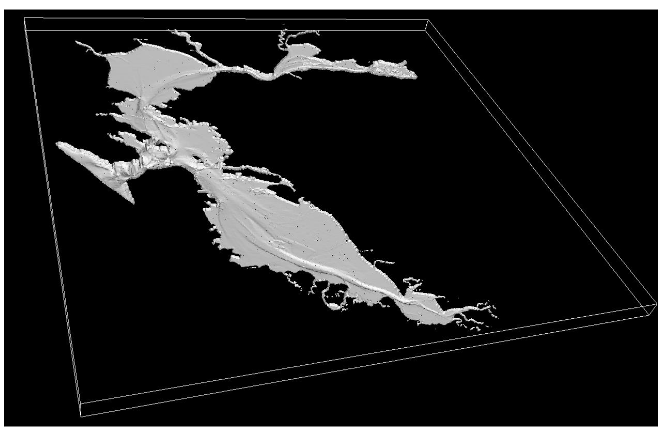
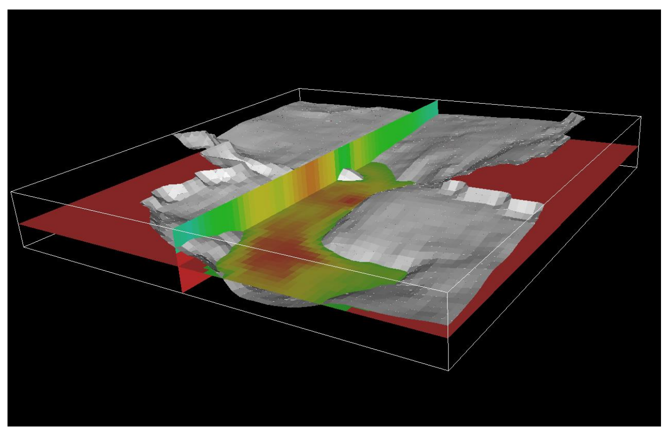
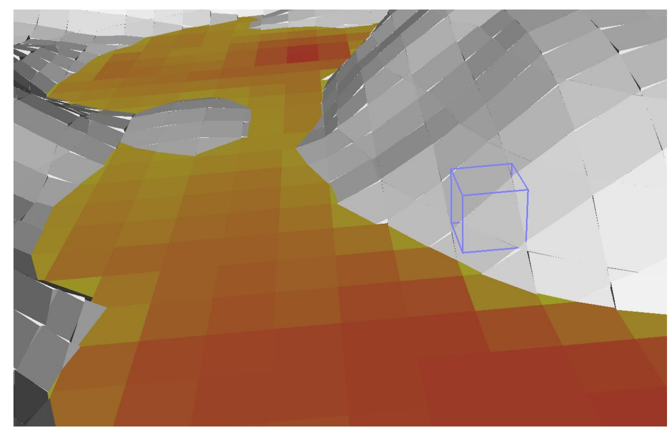
- Built modularly using existing software packages: Python, VTK, Tk, HDF5
- Scripting language with all functionality available
- Data viewing and analysis a core requirement
- Use of OpenGL graphics acceleration including advanced graphics capabilities (e.g., texture mapping)
- Reads and writes data using HDF5 which is machine independent/portable
- Customization via startup file using scripting language
- Data read and stored only on demand
- Non-graphical versions of ChomboVis provided
- Core visualization and data analysis tool of developers

Visualization and Data Analysis - Capabilities

Demonstration and Movies

Visualization and Data Analysis - Features

- Different data centerings
- Multiple tools synchronized (master/slave)
- Offscreen rendering
- Rendering directly to encapsulated PostScript (vector output)
- Particles
- Embedded Boundaries
- Multifluids

Remarks - Software Availability

- Software and documentation is available locally on “joshuatree” under “/usr/local/chombo”
- Also available on the ANAG WWW site: **http://seesar.lbl.gov/anag** under “**Software**”
- E-mail to the developers:
  - chombo@hpcrd.lbl.gov (Chombo)
  - chombovis@hpcrd.lbl.gov (ChomboVis)
- This talk is available at:
  - “joshuatree” under “/usr/local/chombo” as “talk-March28.pdf”
  - **http://seesar.lbl.gov/anag/staff/ligocki/index.html** under the IPAM link

Remarks - Acknowledgments

- DOE Applied Mathematical Sciences Program
- DOE HPCC Program
- DOE SciDAC Program
- NASA Earth and Space Sciences Computational Technologies Program