The focus of this minisymposium is on the large-scale simulation of problems governed by systems of partial differential equations on multicore platforms. There are many challenges including scalability on massively parallel platforms and efficiency of algorithms on multicore processors. This minisymposium will explore challenges and various details concerning the efficient use of massively parallel multicore platforms for engineering applications.

Abstract: We present a suite of AMR hydrodynamics codes for astrophysical applications developed at the Center for Computational Sciences and Engineering at LBNL. MAESTRO is suitable for low Mach number flows and CASTRO is a general compressible code. Both codes scale to 100k-200k cores using a hybrid MPI/OpenMP approach on the Jaguar XT5 supercomputer at OLCF. We are currently studying a variety of astrophysical phenomena including Type Ia supernovae.
Outline

• Overview of MAESTRO and CASTRO
• Numerical Implementation
• Parallel Implementation
• Scaling and Scientific Results

• Collaborators
  – LBNL Center for Computational Sciences and Engineering
    • Ann Almgren, Mike Lijewski, Candace Gilet, John Bell
  – Stony Brook University Dept. of Physics and Astronomy
    • Mike Zingale, Chris Malone
MAESTRO and CASTRO

- Finite volume, block structured adaptive mesh refinement (AMR) codes for astrophysical phenomena
  - System of advection-reaction-diffusion equations
  - Modular equation of state
  - Modular reaction network
  - Massively parallel
    - MAESTRO scales to 100K cores
    - CASTRO scales to 200K+ cores
- CASTRO is a general compressible code
- MAESTRO is a low Mach number code, designed for efficient simulation of low speed flows (relative to the sound speed).
MAESTRO and CASTRO

- Both codes are mature and are being actively used in scientific investigations.

**MAESTRO**
- Type Ia supernovae; pre-ignition (Zingale, Stony Brook)
- Type I X-Ray Bursts (Malone, Stony Brook)
- Classical novae (Krueger, Calder, Stony Brook)
- Convection in massive stars (Gilet, LBNL)

**CASTRO**
- Type Ia supernovae; post-ignition (Ma, Woosley, UCSC)
- Core-collapse supernovae (Nordhaus, Burrows, Princeton)
- Pair instability supernovae (Chen, Heger, U. of Minnesota; Joggerst, LANL)
General Framework

- Finite Volume: solution in each Cartesian cell represents the average over the cell

\[ \rho, \mathbf{u}, p, T, \text{ etc.} \]

- AMR: Block-structured approach with logically rectangular grids
Example: Type Ia Supernova Explosion

- Temperature plot and zoom-in of a 3D Type Ia supernova explosion in CASTRO (Ma, Woosley, UCSC)

- Our software can handle many levels of AMR and scales well for problems with datasets that are 100GB – 1TB.
Software Overview

• BoxLib software framework provides set of tools for finite-volume block-structured AMR applications
  – C++ / Fortran90
  – Subcycling in time (CASTRO only)

• Parallel I/O
  – Peak I/O at NERSC (approx 13 GB/s) is comparable with NERSC benchmarks

• Hierarchical programming model
  – Hybrid MPI/OpenMP approach.
MAESTRO Overview

- MAESTRO is a low Mach number hydrodynamics solver for astrophysical flows.
  - Mach number: \( M = \frac{U}{c} \)
  - In the low Mach number regime, \( M = O(10^{-2}) \), acoustic waves carry little energy so we derive an equation set which excludes them.
  - Time steps are constrained by advective CFL, not acoustic CFL.

\[
\Delta t_{\text{compressible}} < \max \left\{ \frac{\Delta x}{|u| + c} \right\} \quad \Delta t_{\text{lowMach}} < \max \left\{ \frac{\Delta x}{|u|} \right\}
\]

\[
\Delta t_{\text{lowMach}} \sim \frac{1}{M} \Delta t_{\text{compressible}}
\]

- Low Mach time steps are a factor of \( 1/M \) larger!
Low Mach Number Equation Set

- Derived from fully compressible equation set

\[
\begin{align*}
\frac{\partial (\rho X_k)}{\partial t} &= -\nabla \cdot (\rho X_k u) + \rho \dot{\omega}_k & \text{conservation of mass} \\
\frac{\partial (\rho u)}{\partial t} &= -\nabla \cdot (\rho uu) - \nabla \pi + \rho g & \text{conservation of momentum} \\
\frac{\partial (\rho h)}{\partial t} &= -\nabla \cdot (\rho h u) + \nabla \cdot \kappa \nabla T + \rho H & \text{conservation of energy}
\end{align*}
\]

- $\rho$: density
- $u$: velocity
- $X_k$: mass fraction of species “k”
- $\dot{\omega}_k$: reaction rate of species “k”
- $g$: gravity
- $h$: specific enthalpy
- $\kappa$: thermal diffusion coefficient
- $H$: reaction heating
- $\pi$: deviation from ambient pressure
- $T$: temperature
Low Mach Number Equation Set

- Our system is closed with an equation of state, which keeps system in thermodynamic equilibrium.
  - Differentiate equation of state along particle paths to represent as a divergence constraint:
    \[
    \nabla \cdot (\beta_0 \mathbf{u}) = \beta_0 S
    \]
    \[
    \beta_0 \rightarrow \text{captures expansion/contraction of fluid due to changes in altitude}
    \]
    \[
    S \rightarrow \text{captures local compressibility effects due to reactions and thermal diffusion}
    \]

- Numerical enforcement of divergence constraint analogous to solution methodology for incompressible flow
  - Pressure-projection method involving a variable-coefficient Poisson solve
Numerical Methodology

- Strang splitting couples advection/reaction/diffusion
  - **Advection** using Godunov approach
  - **Reactions** using stiff ODE solver
  - **Diffusion** semi-implicit (multigrid)
  - **Divergence-constraint** requires elliptic solve (multigrid)

\[
\begin{align*}
\frac{\partial (\rho X_k)}{\partial t} &= -\nabla \cdot (\rho X_k \mathbf{u}) + \rho \dot{\omega}_k \\
\frac{\partial (\rho \mathbf{u})}{\partial t} &= -\nabla \cdot (\rho \mathbf{u} \mathbf{u}) - \nabla \pi + \rho \mathbf{g} \\
\frac{\partial (\rho h)}{\partial t} &= -\nabla \cdot (\rho h \mathbf{u}) + \nabla \cdot \kappa \nabla T + \rho \mathbf{H} \\
\nabla \cdot (\beta_0 \mathbf{u}) &= \beta_0 S
\end{align*}
\]
Implementation Details

• Divide computational domain into cells.
• Divide the domain into grids
• In a pure MPI instantiation, we assign each grid to a core.
• Each core advances the solution on the grid(s) it owns
• Communication between grids to fill ghost cells
• We repeat this process for each level of refinement.
  – Integrate levels separately
  – Synchronize solution between levels
• Each core advances the solution on the grid(s) it owns
• Communication between grids to fill ghost cells
• Two-way communication between levels required to synchronize solution
Communication Requirements

• Advection and reaction steps require minimal communication – transfer of ghost cell data.

• Multigrid requires intensive communication – often 1000+ calls to transfer ghost cell data per linear solve.

• Also, there is a non-trivial amount of time spent determining communication patterns between grids.

• Our approach is to reduce communication overhead by reducing the number of MPI processes.
  – Hybrid MPI/OpenMP approach.
Parallel Implementation – Pure MPI

- Divide solution domain into grids

- Each grid is assigned to a core
- Cores communicate each other using MPI
  - In this example, we require 12 MPI processes.
Parallel Implementation – Hybrid MPI/OpenMP

• Divide solution domain into fewer, larger grids

• Each grid is assigned to a node
  – Spawn a thread on each core to work on the grids simultaneously

• Nodes communicate each other using MPI
  – In this example, we require 2 MPI processes.
Advantages of Hybrid MPI/OpenMP

• Fewer MPI processes lead to reduced communication time
  – Especially important in communication-intensive multigrid

• Fewer grids leads to reduced memory overhead requirements
  – Metadata contains a mapping of grids and their associated MPI processes.
Other Techniques

- As we approach the coarser levels of multigrid, we transfer the problem to fewer, larger grids, thus reducing the number of MPI processes in the bottom solve.

- Hash sorting to efficiently compute communication patterns by more intelligently searching for neighboring grids.

- Cache commonly used communication patterns.
Type Ia Supernovae

- Full star dynamics
- 5000 km³ domain
- $576^3$ resolution
  - $1728 \cdot 48^3$ grids
Type Ia Supernovae

- Full star dynamics
- 5000 km$^3$ domain
- $576^3$ resolution
  - 1728 · 48$^3$ grids
- $1152^3$ resolution
  - 1831 grids
- $2304^3$ resolution
  - 2449 grids
- $4608^3$ resolution
  - 7072 grids
MAESTRO Strong Scaling

Strong Scaling Behavior of 768^3 MAESTRO Scientific Production Runs on jaguarpf.ccs.ornl.gov

- Pure MPI
- Ideal Scaling
- Hybrid MPI+OpenMP; 12 Threads
- Ideal Scaling

Average Time per Time Step (seconds)

Number of Processors

2K 4K 14K 33K 49K
MAESTRO Weak Scaling

![Graph showing MAESTRO Weak Scaling with data points at 768, 12k, and 96k processors showing an increasing trend in average time per time step (seconds) with the number of processors.]

- **Axes:**
  - Y-axis: Average Time per Time Step (seconds)
  - X-axis: Number of Processors

- **Points:**
  - 768 processors: Approximately 125 seconds
  - 12k processors: Approximately 150 seconds
  - 96k processors: Approximately 175 seconds

- **Trend:**
  - Linear increase with the number of processors.
Type Ia Supernovae with MAESTRO

- We have performed simulations of convection in a white dwarf preceding a Type Ia supernova on the jaguarpf XT5 at OLCF
  - 10K cores, 7M CPU-hours per simulation, \(1152^3\) effective resolution with 2 AMR levels.
  - Will perform more studies at (up to) \(4308^3\) effective resolution with 4 AMR levels.
CASTRO Overview

• Standard compressible equations of motion

\[
\frac{\partial (\rho X_k)}{\partial t} = -\nabla \cdot (\rho X_k \mathbf{u}) + \rho \dot{\omega}_k
\]

\[
\frac{\partial (\rho \mathbf{u})}{\partial t} = -\nabla \cdot (\rho \mathbf{u} \mathbf{u}) - \nabla p + \rho g
\]

\[
\frac{\partial (\rho E)}{\partial t} = -\nabla \cdot (\rho \mathbf{u} E + p \mathbf{u}) + \rho \mathbf{u} \cdot g + \nabla \cdot k_{th} \nabla T + \rho H
\]

• Advection (Godunov method) and reactions (stiff ODE solver) require little communication.

• Semi-implicit thermal diffusion and self-gravity (Poisson equation) are optional.
  
  – Using a monopole gravity approximation and explicit thermal diffusion, CASTRO scales to 200K+ cores.
CASTRO Weak Scaling

![Graph showing CASTRO Weak Scaling with two levels of threads and different processor counts. The graph plots Average Time per Time Step (seconds) against the Number of Processors.](image)
Type Ia Supernovae with CASTRO

- CASTRO has been used to perform simulations of the explosion phase of a Type Ia Supernova on jaguarpf (Haitao Ma, UCSC)
  - 12K cores, 2.5M CPU-hours, $8192^3$ effective resolution with 5 AMR levels.
Summary

• Low Mach number AMR code MAESTRO scales to 100K cores, performing science using O(10K) cores.
• Compressible AMR code CASTRO scales to 200K cores, performing science using O(10K) cores

• We are interested in testing the scalability of our codes on the next generation machines, which may have 24+ cores per node.