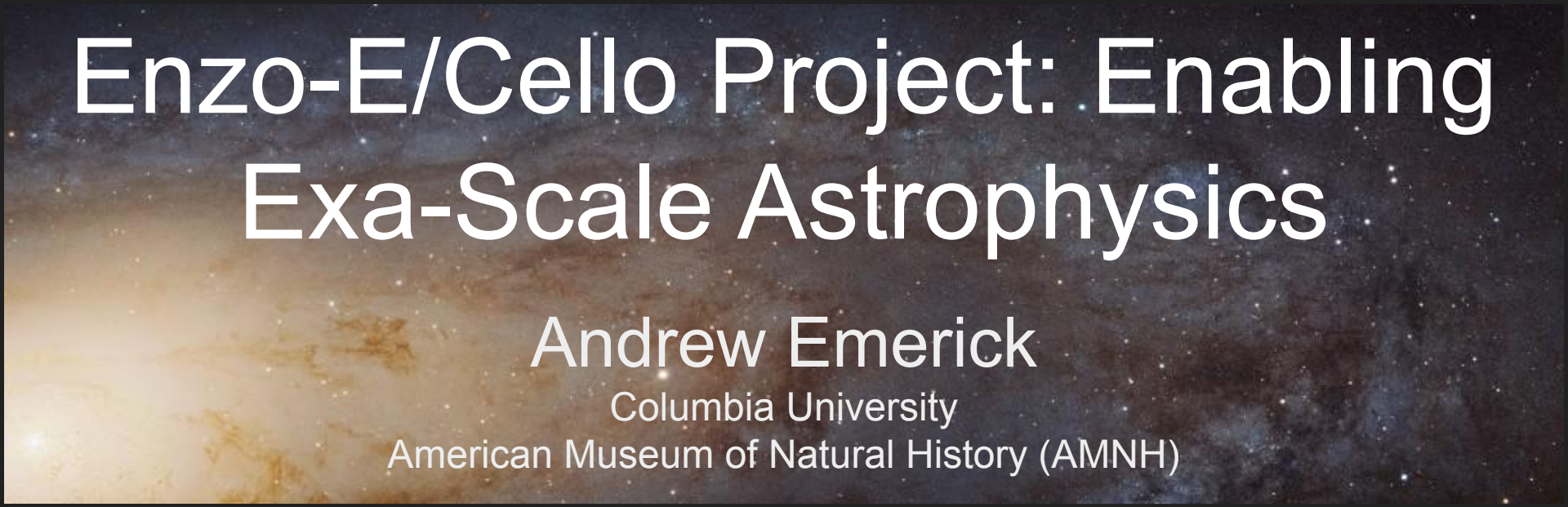


Enzo-E/Cello Project: Enabling Exa-Scale Astrophysics

Andrew Emerick

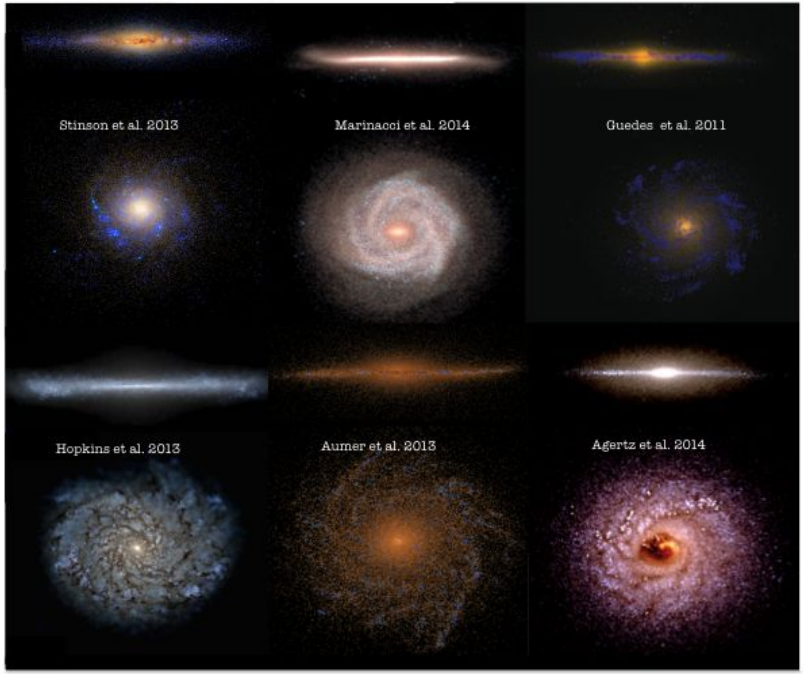
Columbia University

American Museum of Natural History (AMNH)

Greg L. Bryan (Columbia/Flatiron Institute)
Mordecai-Mark Mac Low (AMNH/Columbia/Flatiron)
Brian O'Shea (Michigan State)
John Wise (Georgia Tech)
Mike Norman (San Diego Supercomputing Center (SDSC))
James Bordner (SDSC)
Britton Smith (SDSC)
(and more….)

Progress in Astrophysical Hydrodynamics

Naab & Ostriker 2017

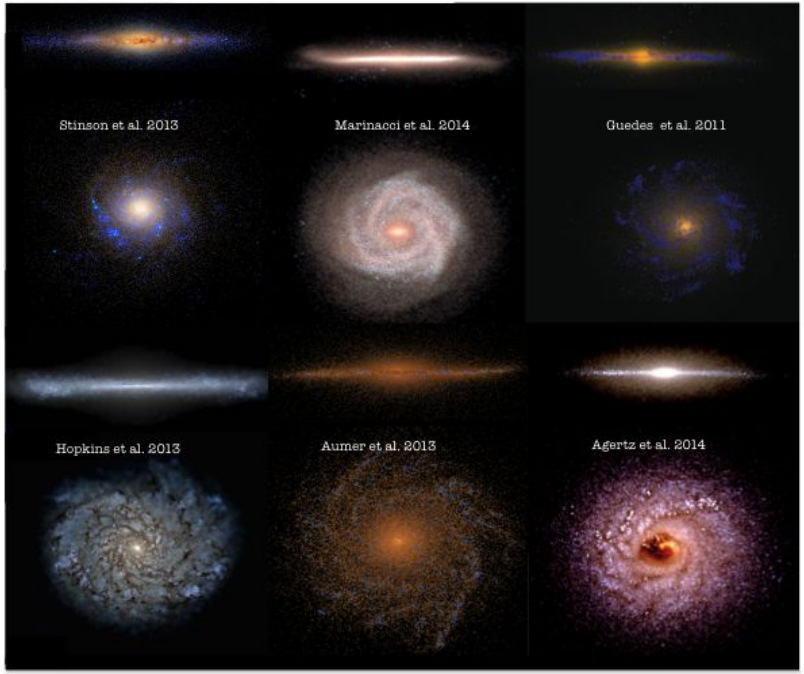


Progress in Astrophysical Hydrodynamics

Enabled by more powerful HPC systems

- Allow for greater dynamic range
- More detailed physics

Naab & Ostriker 2017



Progress in Astrophysical Hydrodynamics

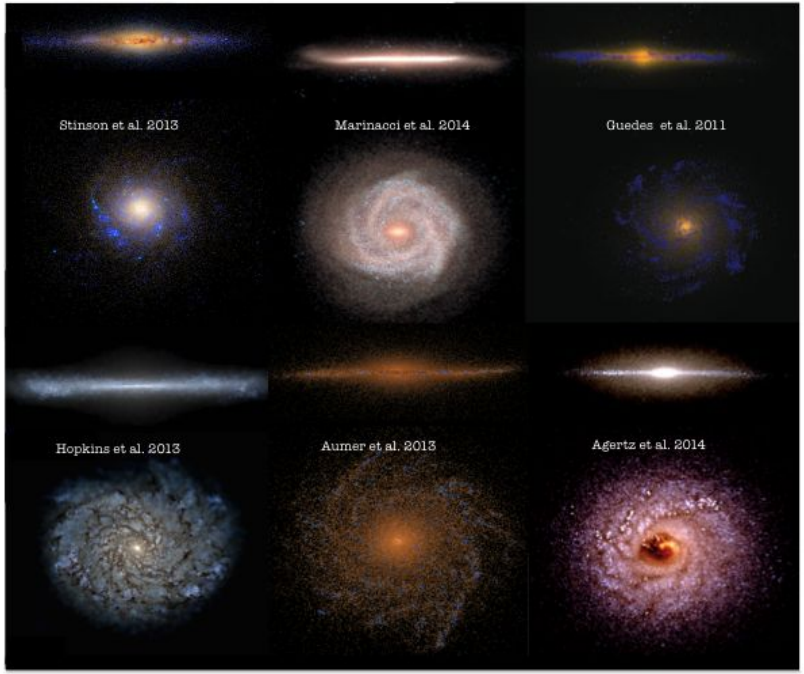
Enabled by more powerful HPC systems

- Allow for greater dynamic range
- More detailed physics

Variety of codes and methods:

- Lagrangian: SPH, moving mesh
- Eulerian: Grid-based codes
- Hybrid, meshless codes

Naab & Ostriker 2017



Enzo: enzo-project.org/

Adaptive mesh refinement (AMR), cosmological hydrodynamics

C/C++ and Fortran

Enzo: enzo-project.org/

Adaptive mesh refinement (AMR), cosmological hydrodynamics

C/C++ and Fortran

Physics:

- Multiple Hydro solvers
- MHD
- Cosmic Rays
- Star formation + stellar feedback
- Ray-tracing radiative transfer
- Cosmology
- Gravity
- Particles
- Radiative heating / cooling
- Chemistry

Enzo: enzo-project.org/

Adaptive mesh refinement (AMR), cosmological hydrodynamics

C/C++ and Fortran

Physics:

- Multiple Hydro solvers
- MHD
- Cosmic Rays
- Star formation + stellar feedback
- Ray-tracing radiative transfer
- Cosmology
- Gravity
- Particles
- Radiative heating / cooling
- Chemistry

Open Source development and stable code: https://github.com/enzo-project

Significant Challenges for the Next Generation:

Scaling and memory management are major shortcoming of current codes

Current scaling to 10^3 - 10^4 cores (at best)

Significant Challenges for the Next Generation:

Scaling and memory management are major shortcoming of current codes

Current scaling to 10^3 - 10^4 cores (at best)

- Load balancing limitations

Significant Challenges for the Next Generation:

Scaling and memory management are major shortcoming of current codes

Current scaling to 10^3 - 10^4 cores (at best)

- Load balancing limitations
- Significant memory overhead

Significant Challenges for the Next Generation:

Scaling and memory management are major shortcoming of current codes

Current scaling to 10^3 - 10^4 cores (at best)

- Load balancing limitations
- Significant memory overhead

Significant Challenges for the Next Generation:

Scaling and memory management are major shortcoming of current codes

Current scaling to 10^3 - 10^4 cores (at best)

- Load balancing limitations
- Significant memory overhead
- Limited (if any) utilization of GPUs

Significant Challenges for the Next Generation:

Scaling and memory management are major shortcoming of current codes

Current scaling to 10^3 - 10^4 cores (at best)

- Load balancing limitations
- Significant memory overhead
- Limited (if any) utilization of GPUs

Overhaul necessary to leverage exascale systems

Enzo - Technical Details

Patch-based, structured AMR

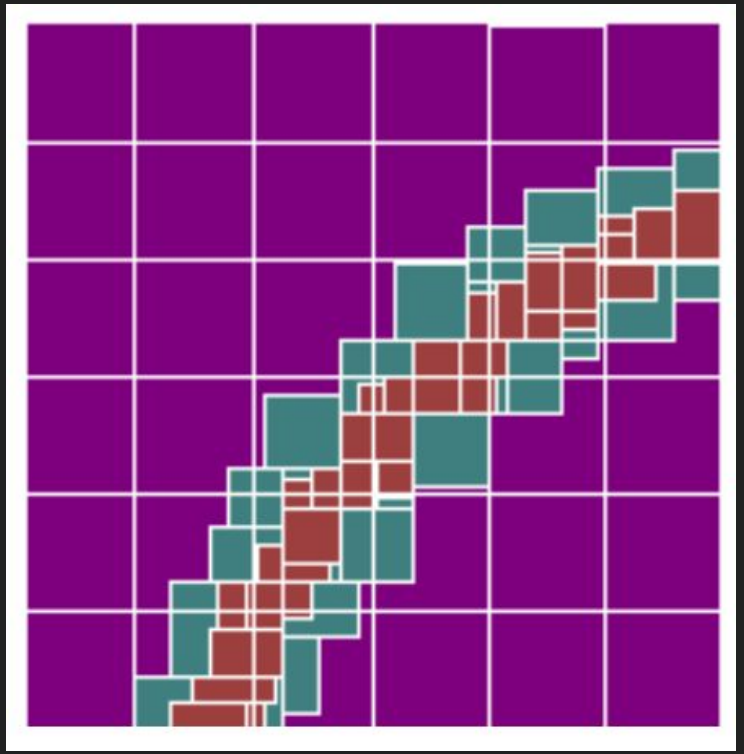


Image Credit: James Bordner

Enzo - Technical Details

Patch-based, structured AMR

Unbalanced mesh

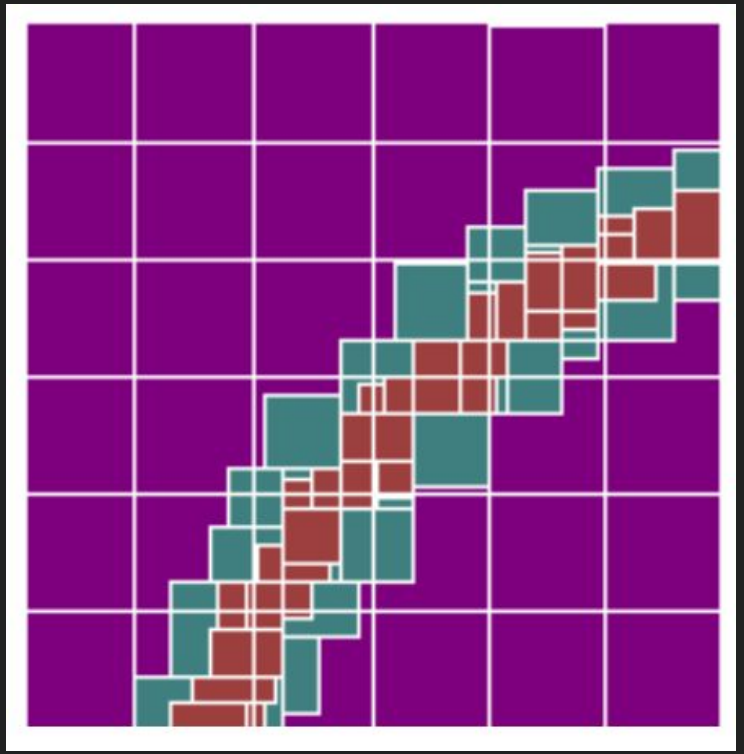


Image Credit: James Bordner

Enzo - Technical Details

Patch-based, structured AMR

Unbalanced mesh

MPI communication

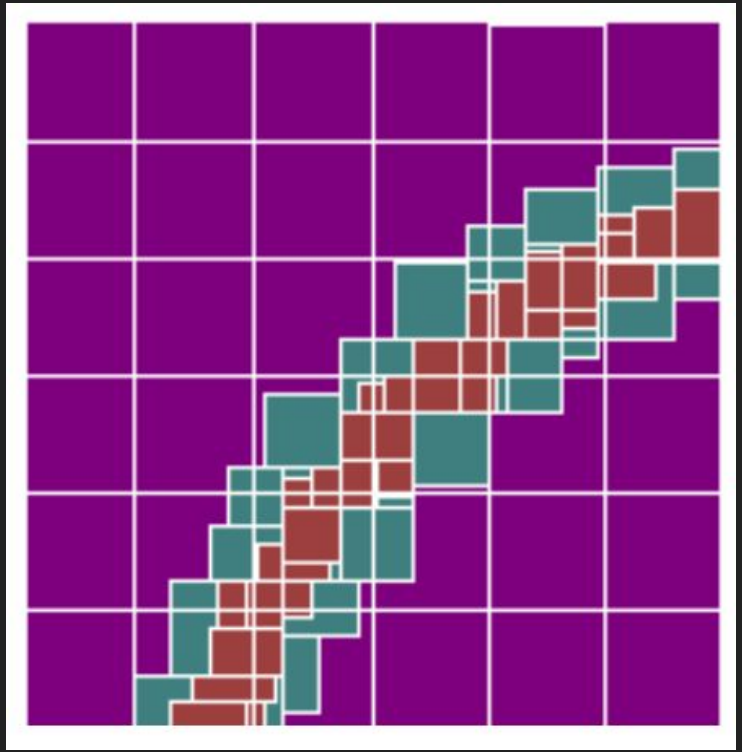


Image Credit: James Bordner

Enzo - Technical Details

Patch-based, structured AMR

Unbalanced mesh

MPI communication

Hybrid particle-mesh methods

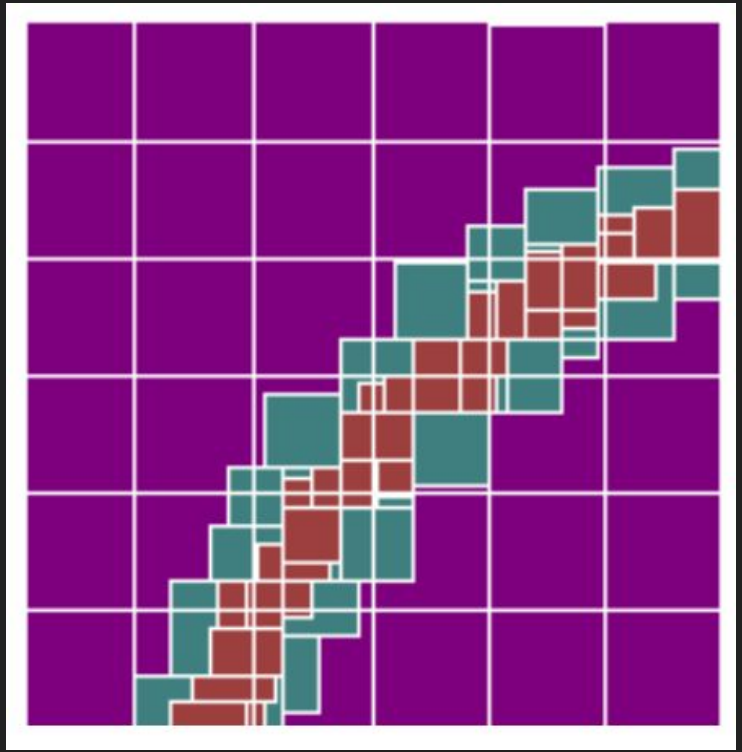


Image Credit: James Bordner

Shortcomings with Enzo in Exascale Era:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)

Shortcomings with Enzo in Exascale Era:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently

Shortcomings with Enzo in Exascale Era:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads

Shortcomings with Enzo in Exascale Era:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads
- Interpolation required from parent-to-child grids

Shortcomings with Enzo in Exascale Era:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads
- Interpolation required from parent-to-child grids
- Evolution occurs level-by-level across entire computational domain

Enzo-E / Cello Project

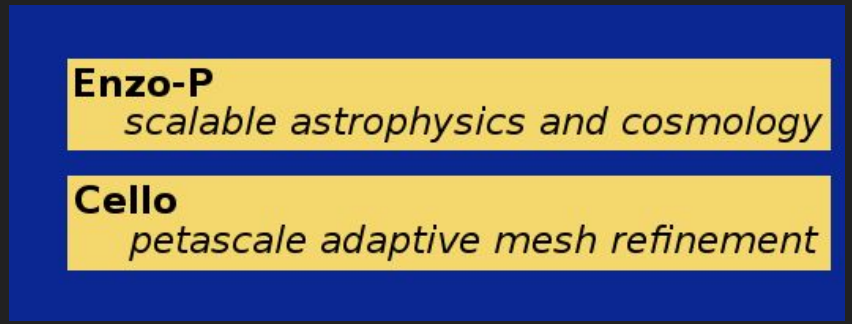


Image Credit: James Bordner

Exascale hydrodynamics from scratch

Open-source:

http://cello-project.org/

https://github.com/enzo-project/enzo-e

James Bordner (SDSC)

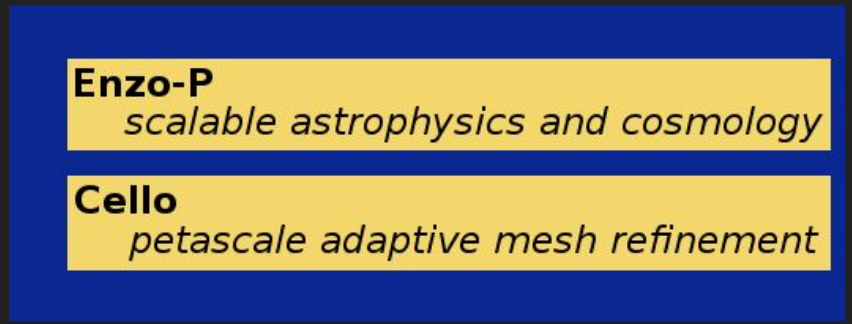
Mike Norman* (SDSC)

… and more:

Matthew Abruzzo (Columbia), Greg Bryan (Columbia), Forrest Glines* (MSU), Brian O'Shea (MSU), Britton Smith (Edinburgh), John Wise (Georgia Tech.), KwangHo Park (Georgia Tech.), David Collins (FSU)....

* = here at the Blue Waters Symposium

Enzo-E / Cello Project



Exascale hydrodynamics from scratch

“Cello” :

- Hierarchy, parallelization
- Charm++ interaction
- Easy APIs for use in Enzo-E layer

“Enzo-E” :

- Initial conditions generators
- Block-by-block methods (physics)

Image Credit: James Bordner

Enzo-E / Cello Project

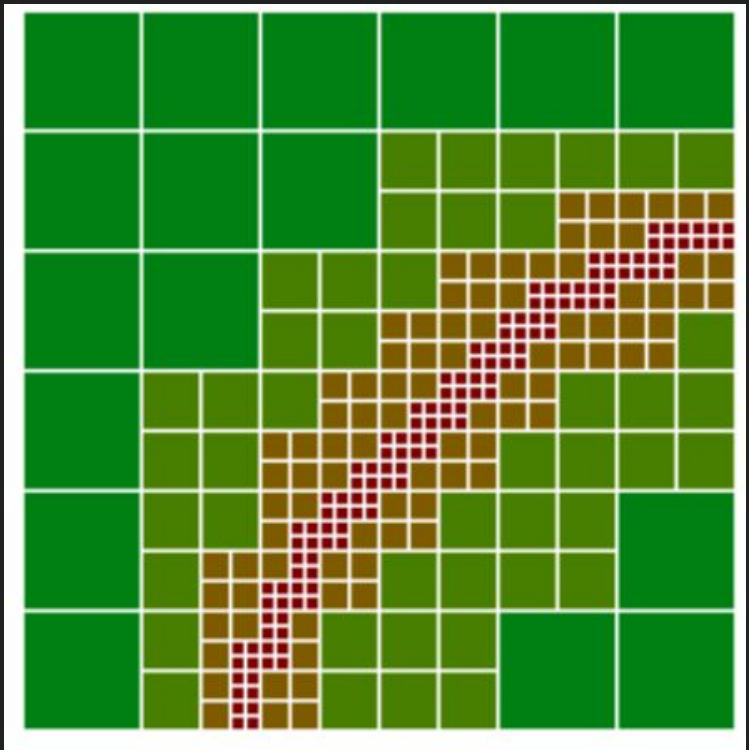


Image Credit: James Bordner

- Octree-based AMR
- Balanced Mesh
- More object oriented programming model
- Charm++ Parallelization

Enzo-E / Cello Project

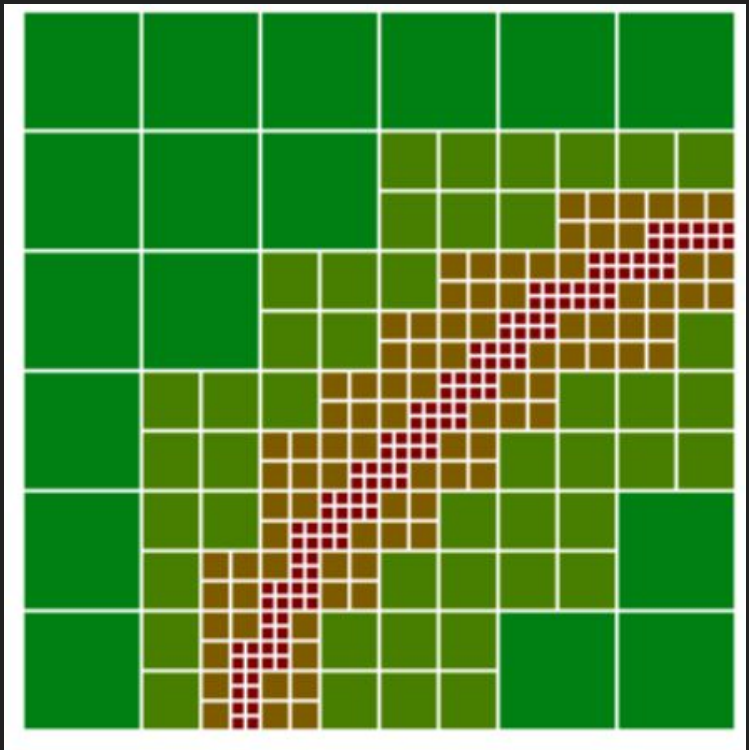


Image Credit: James Bordner

Octree-based AMR

Balanced Mesh

More object oriented programming model

Charm++ Parallelization

- Task-based parallelism

Enzo-E / Cello Project

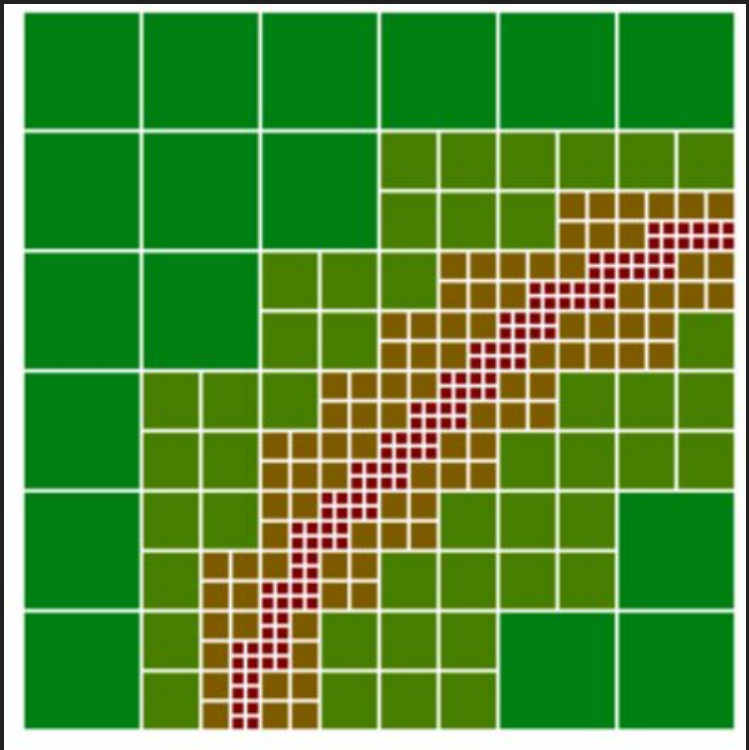


Image Credit: James Bordner

- Octree-based AMR
- Balanced Mesh
- More object oriented programming model
- Charm++ Parallelization
  - Task-based parallelism
  - Asynchronous execution

Enzo-E / Cello Project

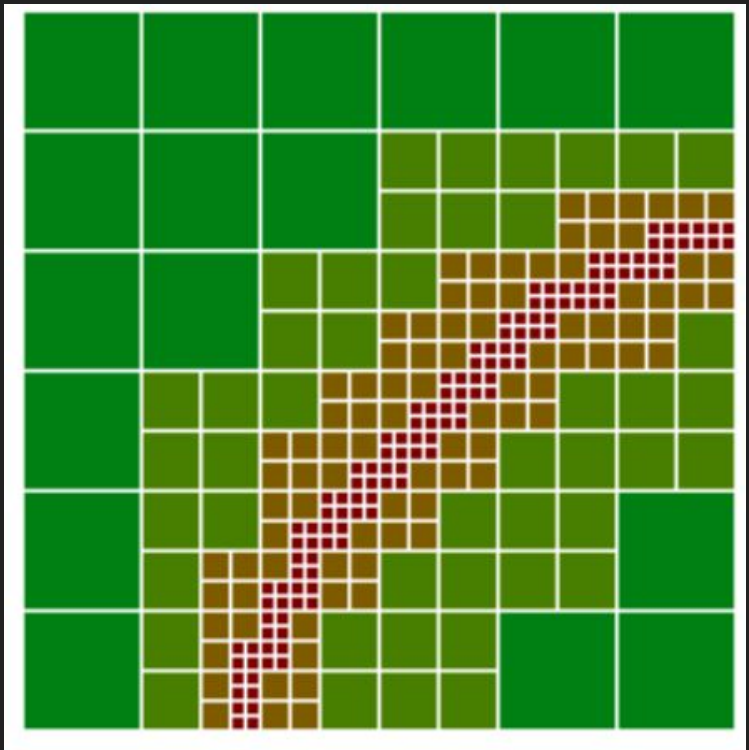


Image Credit: James Bordner

- Octree-based AMR
- Balanced Mesh
- More object oriented programming model
- Charm++ Parallelization
  - Task-based parallelism
  - Asynchronous execution
  - Automatic load balancing

Advancements with Enzo-P/Cello:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads
- Interpolation required from parent-to-child grids
- Evolution occurs level-by-level across entire computational domain

Enzo-E/Cello:

- Hierarchy is localized

Advancements with Enzo-P/Cello:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads
- Interpolation required from parent-to-child grids
- Evolution occurs level-by-level across entire computational domain

Enzo-E/Cello:

- Hierarchy is localized
- Each block is its own parallel task, independent of level

Advancements with Enzo-P/Cello:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads
- Interpolation required from parent-to-child grids
- Evolution occurs level-by-level across entire computational domain

Enzo-E/Cello:

- Hierarchy is localized
- Each block is its own parallel task, independent of level
- Charm++ provides significant load balancing and scheduling advantages

Advancements with Enzo-P/Cello:

Enzo:

- Replicates hierarchy across all MPI processes (memory intensive)
- Patch-AMR is difficult to load balance efficiently
- Parent-child communication overheads
- Interpolation required from parent-to-child grids
- Evolution occurs level-by-level across entire computational domain

Enzo-E/Cello:

- Hierarchy is localized
- Each block is its own parallel task, independent of level
- Charm++ provides significant load balancing and scheduling advantages
- Fixed block size allows for efficient, simplified load balancing

Pushing the limits of AMR Hydrodynamics

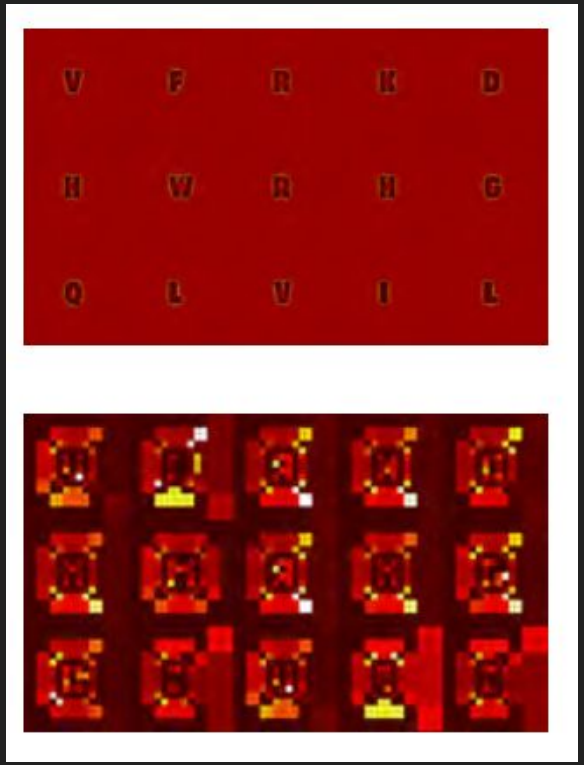


Image Credit: James Bordner

AMR Hydro Scaling: “Exploding Letters” Test

One of largest AMR simulations, run on Blue Waters:

- 256k cores
- 1.7×10^9 grid cells (32^3 cells per block)
- 50×10^6 blocks

Impossible to do with Enzo:

- Enzo’s hierarchy would require 72 GB / proc.!!!

Scaling Results

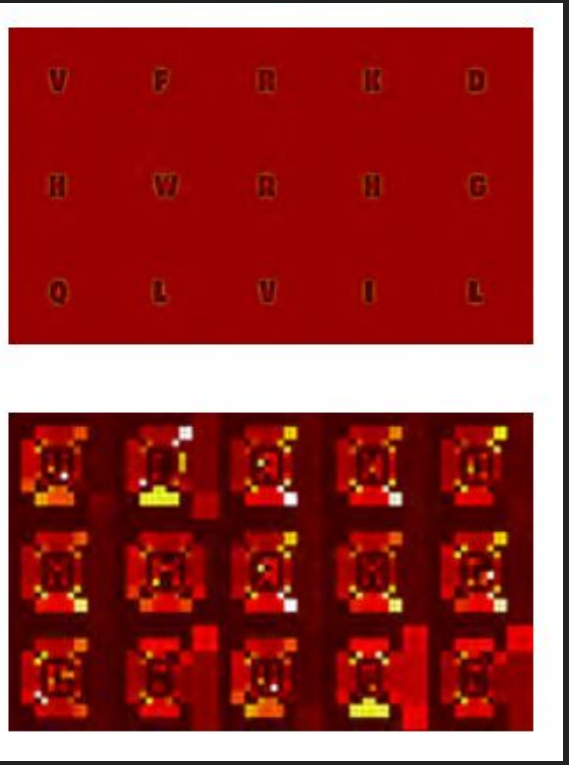
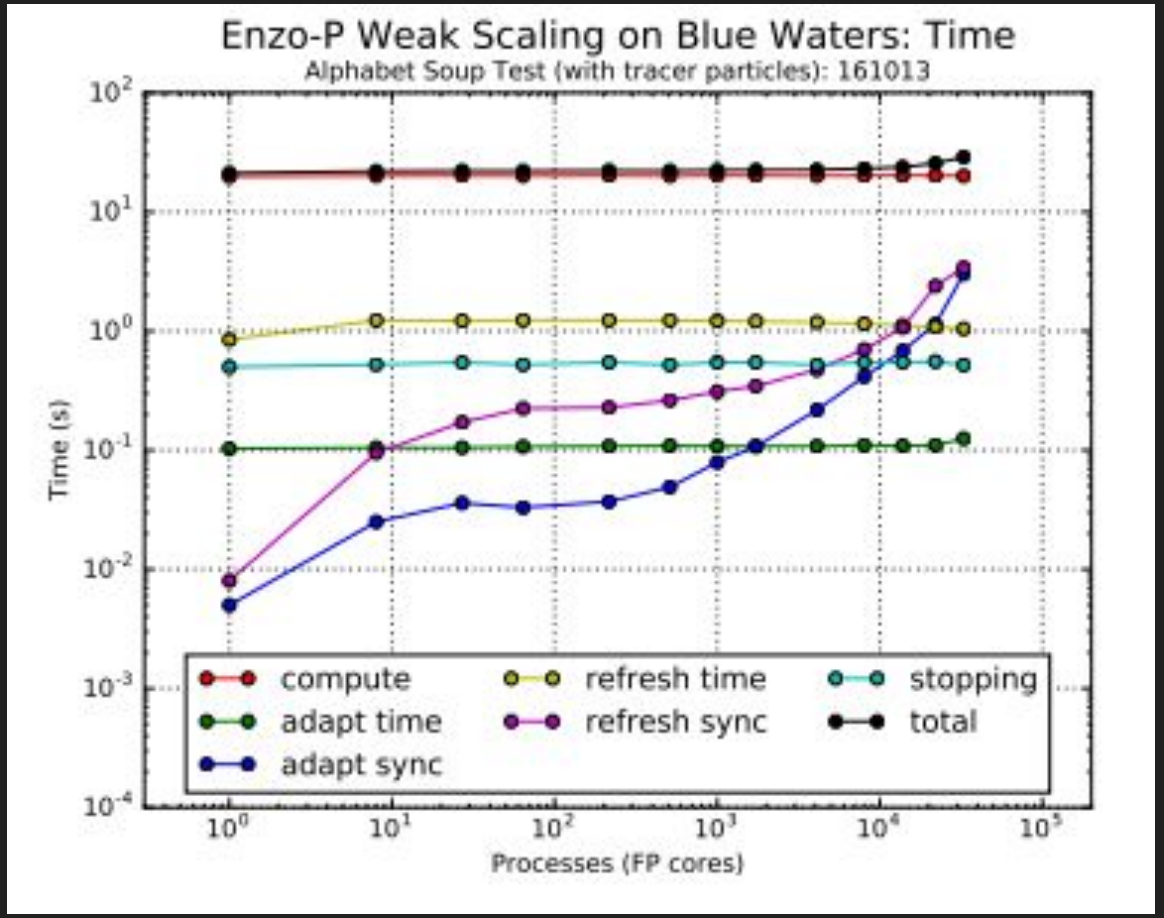


Image Credit: James Bordner



Goals as a Blue Waters Fellow

Implement physics methods to simulate an isolated, Milky Way galaxy

a) Gas cooling and chemistry (GRACKLE package)
b) Background acceleration /potential field
c) Star Formation
d) Stellar Feedback (supernova)
e) Isolated galaxy ICs (with particle support)

Stepping stone to full-physics cosmological simulations

Test-case for how to develop in the new Enzo-E / Cello framework

Defining Community Development in Enzo-E

Similar development structure to Enzo

Migrated code development to github, managed with git

Adopting a pull request development framework

- New additions pulled into master via a pull request
- Reviewed and accepted by 2-3 developers, with final PR-tsar approval

Development community growing (~5 - 10 people)

Future Work: Exascale Astrophysics

Flux correction

Modern stellar feedback algorithms

AMR Cosmology and isolated galaxy runs

MHD with cosmic rays

Ray-tracing radiative transfer

Block adaptive time stepping

Questions?