SpECTRE: A Task-based Discontinuous Galerkin Code for Relativistic Astrophysics

Scott Field

Cornell Center for Astrophysics, Cornell University
Department of Mathematics, U. Mass Dartmouth

GR21, Relativistic Astrophysics, July 11, 2016

Collaborators: Lawrence Kidder, Francois Foucart, Erik Schnetter, Saul Teukolsky, Andy Bohn, Nils Deppe, Peter Diener, Francois Hebert, Jonas Lippuner, Jonah Miller, Christian Ott, Mark Scheel, Trevor Vincent

SIMULATING EXTREME SPACETIMES
Black holes, neutron stars, and beyond...

Outline

1. Motivation and goals
2. Tools
   - DG methods
   - Task-based parallelization
3. Results
   - Status
   - Tests

Outline

1. Motivation and goals
2. Tools
   - DG methods
   - Task-based parallelization
3. Results
   - Status
   - Tests

Challenges of relativistic hydrodynamics

- >50 coupled nonlinear PDEs with lengthly right-hand-side expressions
- Shocks in the hydrodynamic sector
- Turbulent flows
- Density profiles which quickly vary between unity and zero
- Multiscale and multiphysics
- Matter fields move throughout the computational grid

Challenges of relativistic hydrodynamics

- >50 coupled nonlinear PDEs with lengthly right-hand-side expressions
- Shocks in the hydrodynamic sector
- Turbulent flows
- Density profiles which quickly vary between unity and zero
- Multiscale and multiphysics
- Matter fields move throughout the computational grid

Goals

- Higher (than 2^{nd} order) accuracy for relativistic simulations with matter
- Efficient simulations with $\approx 100,000 - 1,000,000$ cores on next generation of supercomputers
- Provide predictions for multi-messenger astronomy

Discontinuous Galerkin (DG) method using task-based parallelism

Outline

1. Motivation and goals
2. Tools
   - DG methods
   - Task-based parallelization
3. Results
   - Status
   - Tests

Why discontinuous Galerkin methods?

- ✓ High-order where the solution is smooth
- ✓ Shock-capturing, robust
- ✓ Parallelizes well (only communicate surface data)
- ✓ Over 2 decades of experience in engineering and applied math

Why discontinuous Galerkin methods?

- ✓ High-order where the solution is smooth
- ✓ Shock-capturing, robust
- ✓ Parallelizes well (only communicate surface data)
- ✓ Over 2 decades of experience in engineering and applied math

Uses in computational relativity and astronomy

- Linearized Einstein equation (Zumbusch 2009, Field+ 2009)
- 1-dimensional Einstein equation (Field+ 2010, 2012)
- 1-dimensional general relativistic hydrodynamics (Radice+ 2011)
- Formulation of method for relativistic astrophysics (Teukolsky 2015)
- 3-dimensional (relativistic) hydrodynamics in an astrophysical context (Schaal+ 2015, Bugner+2015)
- 3-dimensional BSSN system (Miller+ 2016)

Significant opportunity for applications of discontinuous Galerkin methods

Discontinuous Galerkin methods (one slide summary)

1. **Domain decomposition**: Cover the physical domain by local subdomains
2. **Spectral methods**: On each subdomain approximate the solution by a set of basis functions
3. **Finite element**: Substitute the approximate solution into the PDE, integrate the residual against test functions
4. **Finite volume**: Couple neighboring subdomains with a *numerical* flux

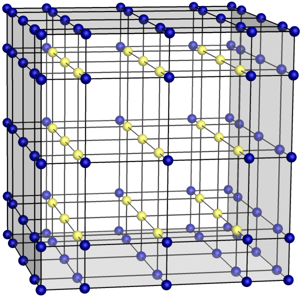


Figure : Typical subdomain with interpolation points. [Fig. from Huismann et al. 2015]

Overview of parallelization strategies

Data-based: Distribute data across cores

- Do exact same thing on each core, pass data with MPI
- Evolution proceeds according to a global simulation time
- Drawbacks: (i) idle cores due to synchronization/waiting and (ii) overlapping computation with communication is non-trivial

Overview of parallelization strategies

Data-based: Distribute data across cores

- Do exact same thing on each core, pass data with MPI
- Evolution proceeds according to a global simulation time
- Drawbacks: (i) idle cores due to synchronization/waiting and (ii) overlapping computation with communication is non-trivial

Task-based: Parallelize by which tasks are to be performed

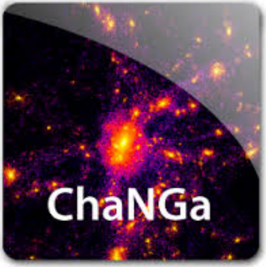
- Computation is naturally broken up into tasks
- As long as task-queues remain non-empty, cores will be hard at work
- No global synchronization
- Overlap of communication and computation is trivial
- Load balancing, task-stealing

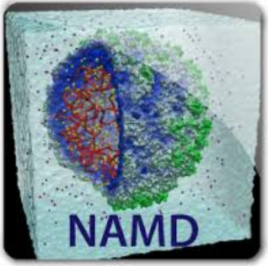
Task-based parallelism with Charm++

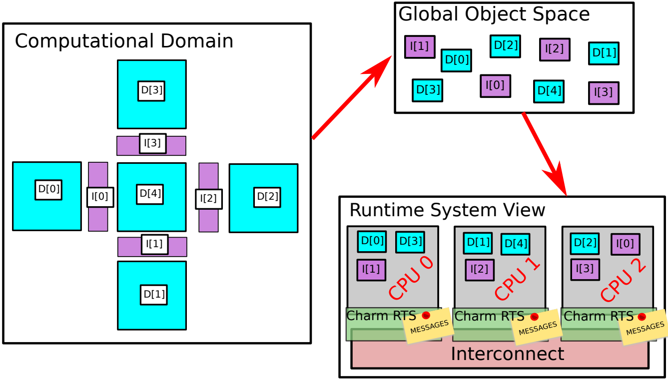
Charm++ (and related tools) is a library maintained by the Parallel Programming Laboratory at UIUC

Project started around 1992

Large community, 10s of code developers and reasonable documentation







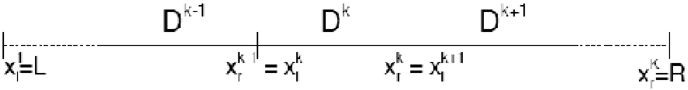
Runtime system maintains a “pool” consisting of tasks to be completed

The runtime system decides which messages from the work-pool to execute

Control switches back-and-forth between the scheduler and SpECTRE’s code

Tasks sent asynchronously (no waiting)

Charm++-ification of a DG scheme



Equation on subdomain D^k (interface between neighbors D^k and $D^{k\pm1}$)

$$\int_{D^k} (\ell_i \partial_t \Psi)\, dx = \int_{D^k} \ell_i \partial_x f(\Psi)\, dx + \text{Numerical Flux}\left(D^{k-1}, D^{k+1}\right)$$

Tasks for each subdomain D^k

- Send data to interfaces
- Compute volume RHS
- Slope limit the solution
- Send data to observers

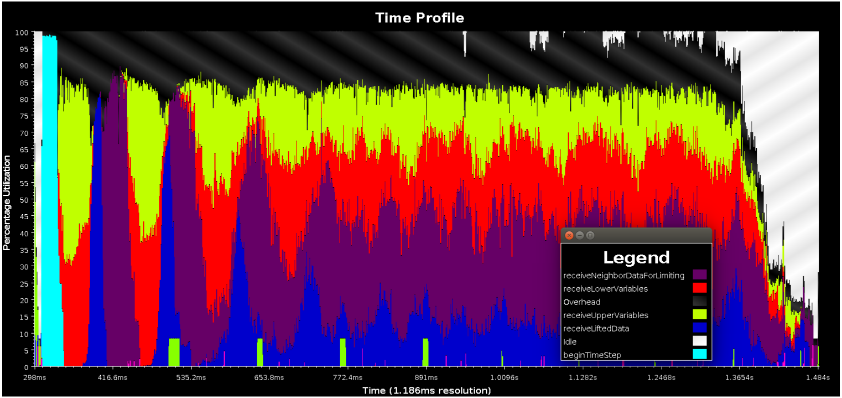
Tasks for each interface

- Compute numerical fluxes

Tasks for Observers

- When data arrives, process
- Write files

Time profile (10 steps of a relativistic MHD system)



Cyan: setup

Black: Charm++

White: idle

Red/Yellow: data to interfaces, local volume RHS

Blue: fluxes to elements

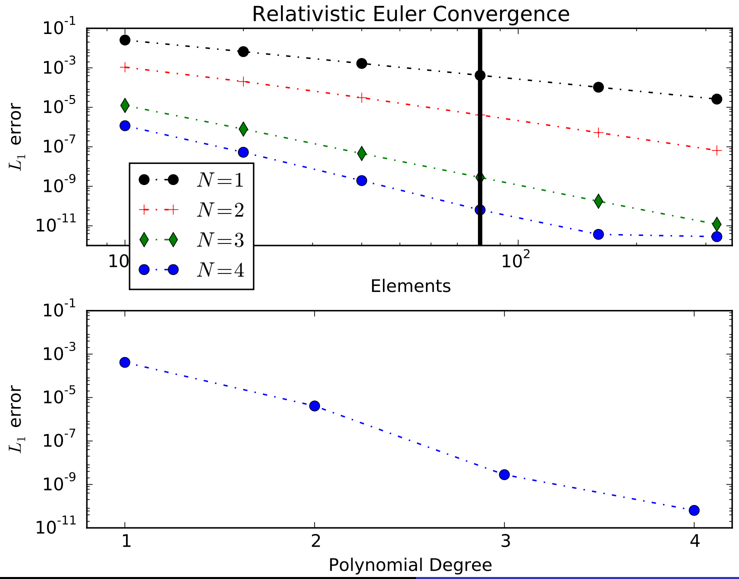
Purple: slope limiting

Outline

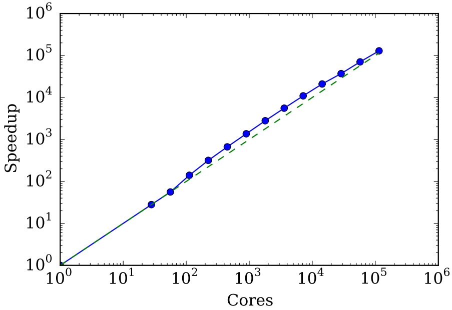
Summary of code

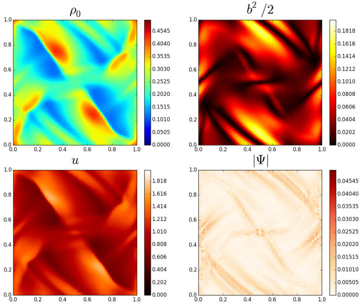
- Discontinuous Galkerkin (DG) method in 1D, 2D and 3D
- Elements are boxes, arbitrary order polynomial basis on each element
- Task-based parallelization carried out with Charm++
- Systems: Scalar wave, (non-)/Relativistic Euler, Relativistic MHD
- Numerical fluxes: Lax-Friedrichs, Roe, Marquina, HLL
- Variety of slope limiters
- Writes to HDF5, simple visualizations

Convergence test – relativistic hydro (smooth solution)



Strong scaling test on Blue Waters





Left: Strong-scaling on Blue Waters for Orszag-Tang vortex test.

Fixed size problem with 660 $\times$ 660 $\times$ 4 second order elements

✓ Similar results on Comet, Stampede

Right: Density, magnetic field strength, velocity

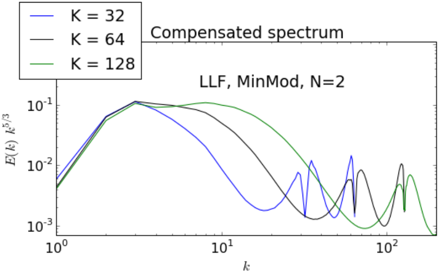
Turbulence

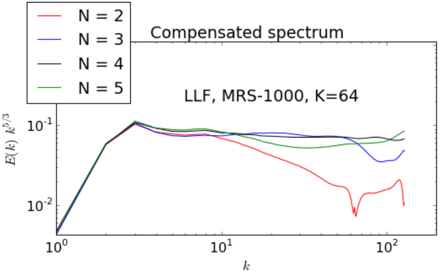
See e.g.
Radice, Couch, Ott (2015).

Kolmogorov inertial range:
$E(k) \sim k^{-5/3}$

- Can we perform high-order turbulence simulations?
- What are good choices for fluxes and limiters?

Additional collaborator:
David Radice (Caltech)





Highlights

- Large-scale simulation code for computational relativity/astronomy using discontinuous Galerkin methods
- First combination of DG methods with task-based parallelization
- Efficient usage up to $\mathcal{O}(100\,,000)$-cores (runs on all of Blue Waters)

Whats next

- h-p adaptivity
- Einstein Equation
- General grids (ie. mappings)
- Load balancing
- Code is almost ready to do science!