

Introduction to the Einstein Toolkit

Roland Haas, Steven R. Brandt, Frank Löffler, Peter Diener, others

National Center for Supercomputing Applications,
University of Illinois Urbana-Champaign

June 13, 2022



Einstein Toolkit



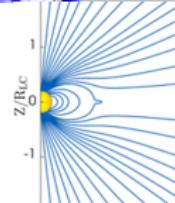
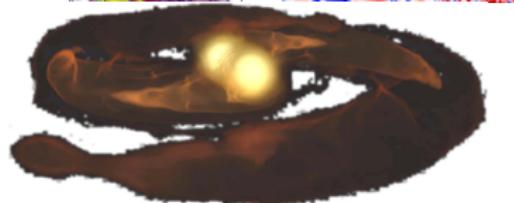
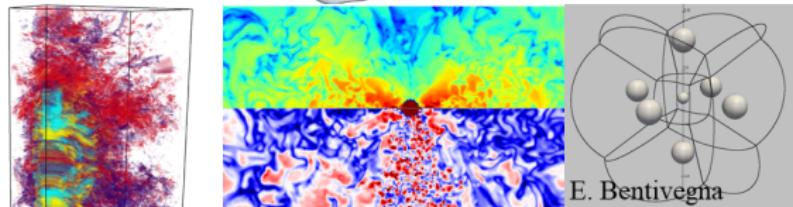
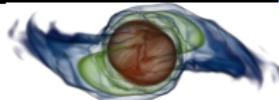
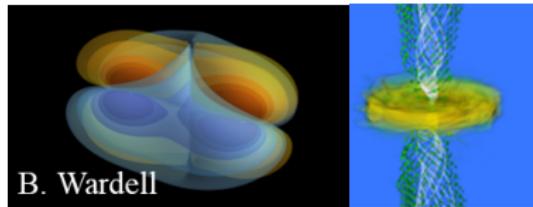
- Collection of scientific software components and tools to simulate and analyze general relativistic astrophysical systems
- Freely available as open source at <http://www.einsteintoolkit.org>
- Supported by NSF 1550551/1550461/1550436/1550514, NSF 1212401/1212426/1212433/1212460, NSF 0903973/0903782/0904015 (CIGR), 0701566/0855892 (XiRel), 0721915 (Alpaca), 0905046/0941653(PetaCactus/PRAC)
- State-of-the-art set of tools for numerical relativity, open source
- Currently 356 members from 249 sites and 43 countries
- > 396 publications, > 53 theses building on these components (as of June 2022)
- Regular, tested releases
- User support through various channels



Einstein Toolkit

Science

- Binary Black Hole Mergers
- Neutron Star Mergers
- Supernovae
- Accretion Disks
- Boson Stars
- Hairy Black Holes
- Cosmic Censorship



Community Effort!



Why?

Computational Challenges

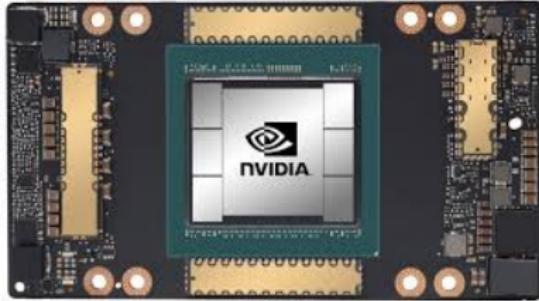


Computational Challenges





More and more diverse hardware





Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Cache



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Cache
 - Vector



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Cache
 - Vector
 - Accelerators



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Cache
 - Vector
 - Accelerators
 - Scale to many cores



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Cache
 - Vector
 - Accelerators
 - Scale to many cores
 - Scale to many nodes



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Cache
 - Vector
 - Accelerators
 - Scale to many cores
 - Scale to many nodes
 - Algorithms

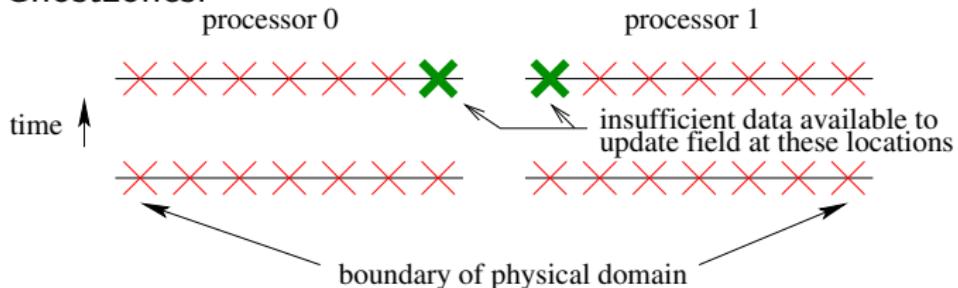


Computational Challenges

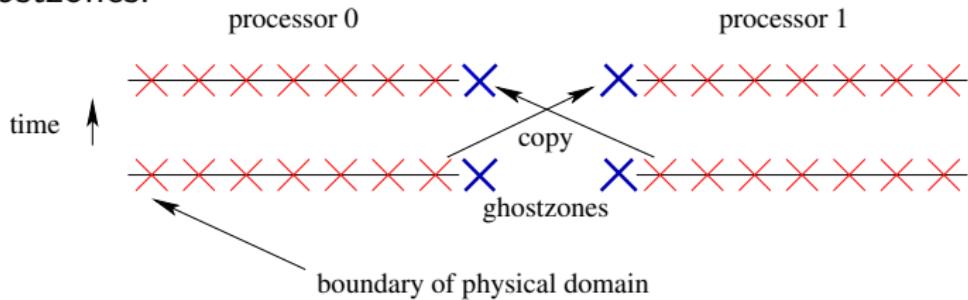
- Efficient use of all hardware is complex and tedious.
- Requires experts from different disciplines
- Requires good data layouts and APIs
- To ensure correctness, need good modularization on a number of levels and understanding of advanced programming concepts.
- Design and implementation needs to be carefully thought out in order to ensure extensibility and portability.

Domain Decomposition

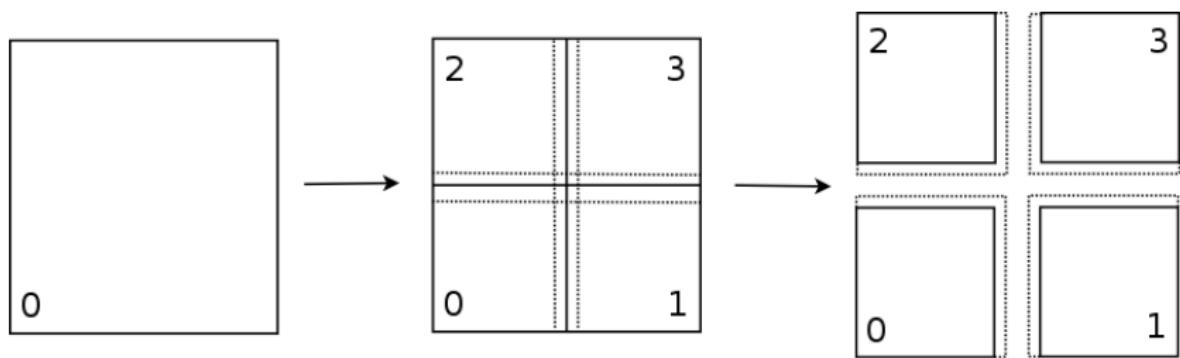
Without Ghostzones:



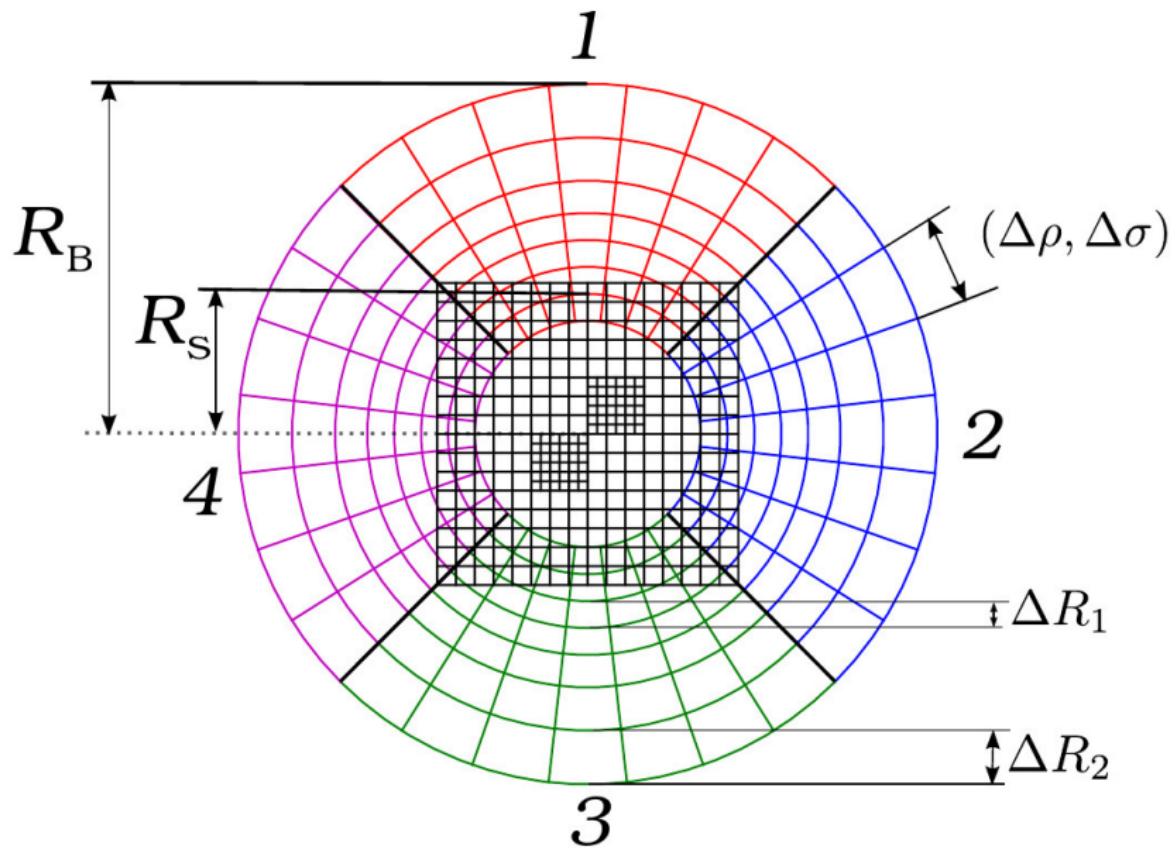
With Ghostzones:



Domain decomposition



Multiblock and refinement



ICSA

Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPy+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)
 - Algorithms / AMR (Adaptive Mesh Refinement, Carpet, CarpetX, MoL)



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)
 - Algorithms / AMR (Adaptive Mesh Refinement, Carpet, CarpetX, MoL)
 - GPU (CarpetX)



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)
 - Algorithms / AMR (Adaptive Mesh Refinement, Carpet, CarpetX, MoL)
 - GPU (CarpetX)
 - FPGA?



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)
 - Algorithms / AMR (Adaptive Mesh Refinement, Carpet, CarpetX, MoL)
 - GPU (CarpetX)
 - FPGA?
 - ASIC?



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)
 - Algorithms / AMR (Adaptive Mesh Refinement, Carpet, CarpetX, MoL)
 - GPU (CarpetX)
 - FPGA?
 - ASIC?
 - Neuromorphic processor?



Computational Challenges

- Simulate cutting edge science
- Use latest numerical methods
- Make use of latest hardware
 - Vector (Kranc, NRPY+)
 - Scale to many cores (OpenMP)
 - Scale to many nodes (MPI, Carpet, CarpetX)
 - Algorithms / AMR (Adaptive Mesh Refinement, Carpet, CarpetX, MoL)
 - GPU (CarpetX)
 - FPGA?
 - ASIC?
 - Neuromorphic processor?
 - Q-bits?



Computational Challenges

More Mundane Challenges



Computational Challenges

More Mundane Challenges

- Efficient I/O



Computational Challenges

More Mundane Challenges

- Efficient I/O
- HDF5



Computational Challenges

More Mundane Challenges

- Efficient I/O
- HDF5
- Checkpoint/Restart



Computational Challenges

More Mundane Challenges

- Efficient I/O
- HDF5
- Checkpoint/Restart
- Parameter Parsing



Computational Challenges

More Mundane Challenges

- Efficient I/O
- HDF5
- Checkpoint/Restart
- Parameter Parsing
- Visualization



Computational Challenges

More Mundane Challenges

- Efficient I/O
- HDF5
- Checkpoint/Restart
- Parameter Parsing
- Visualization
- Analysis



Computational Challenges

More Mundane Challenges

- Efficient I/O
- HDF5
- Checkpoint/Restart
- Parameter Parsing
- Visualization
- Analysis
- Steering



Collaborative Challenges



Collaborative Challenges



?
problem



group



group



group



group



group



group



group

?
computation



group



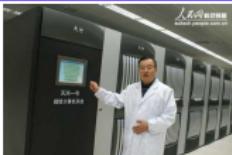
group



group



group



group



group



group



group











Workshop





group



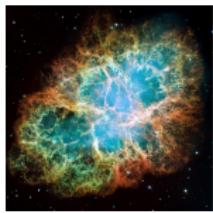
group



group



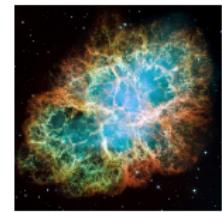
group



group

group

! competition



group

group



group



group

! standards



group



group



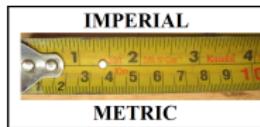
group



group



group



group



group



group



group



group



group



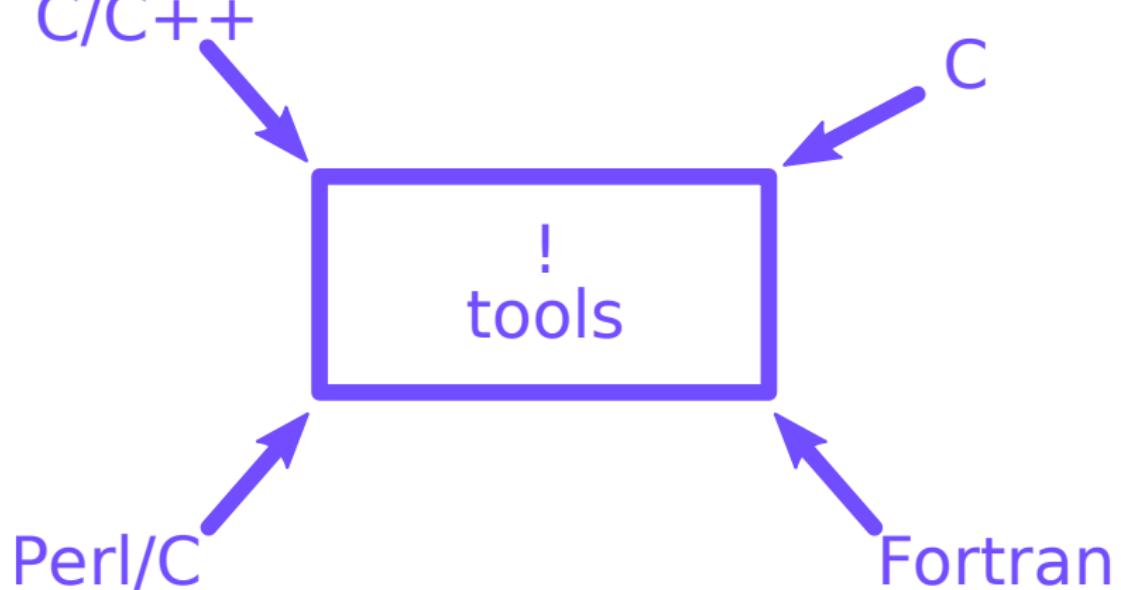
group

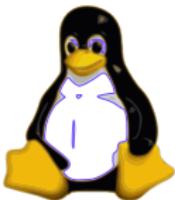


group



group





C/C++

C

! tools



Perl/C

Fortran



group



group



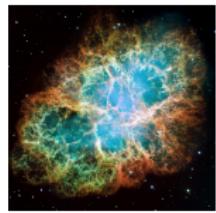
group



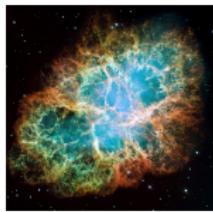
group



group



group



group



group

Collaborative Challenges

How can we work together?

- Researchers in the USA

- Arizona
- Florida
- Georgia
- Louisiana
- Illinois
- Indiana
- New York
- Tennessee
- Texas
- Pennsylvania
- California



- In other countries

- Canada
- Germany
- Italy
- Ireland
- Mexico
- Portugal
- Spain
- Turkey
- United Kingdom
- and many more





Goals:

- Community Driven
- Core computational tool for numerical astrophysics
- General purpose tool!

Components:

- Cactus
- Simulation Factory
- Kranc
- NRPy+
- Science Modules



Guiding Principles

- Open
- Community Driven
- Good interfaces
- Separation of physics from computational infrastructure
- Production ready
- High quality code

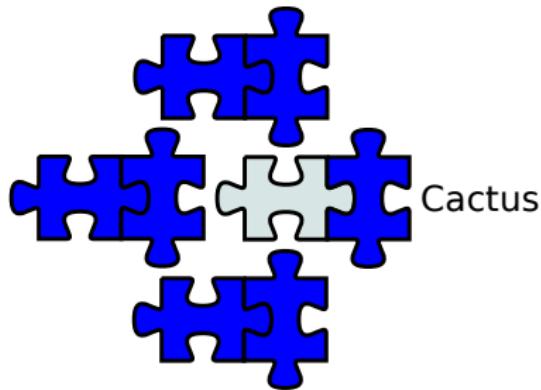
Einstein Toolkit as growing project

- Initially: some infrastructure, some application code



Einstein Toolkit as growing project

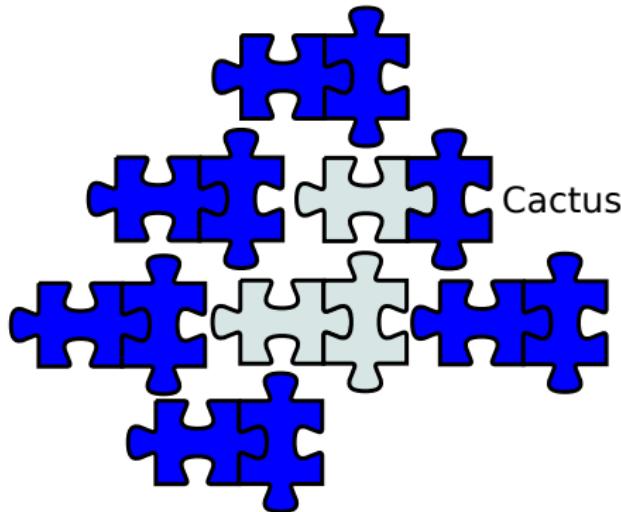
- Growing application suite



Cactus

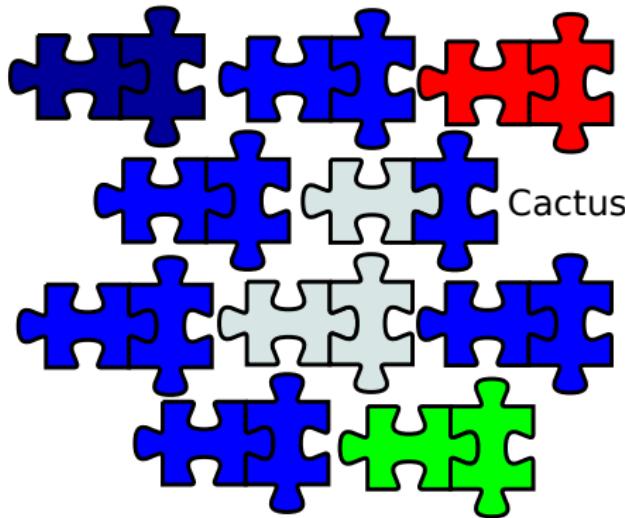
Einstein Toolkit as growing project

- Growing infrastructure “return”



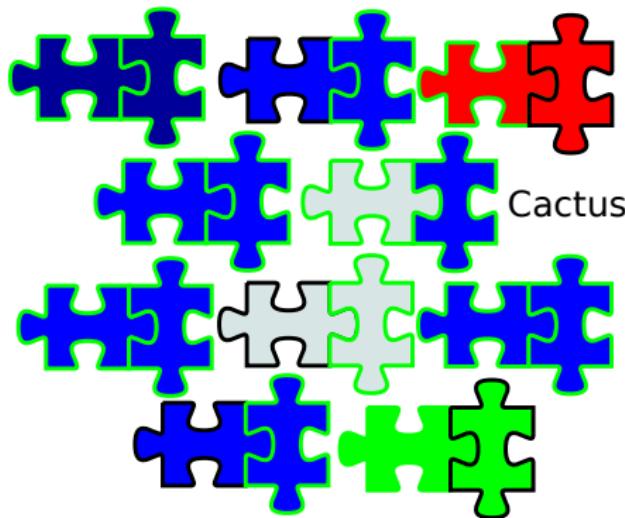
Einstein Toolkit as growing project

- Users from more fields of science



Einstein Toolkit as growing project

- Most modules open-source, but not necessarily all



Base Modules



The Einstein Equations

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

spacetime
curvature



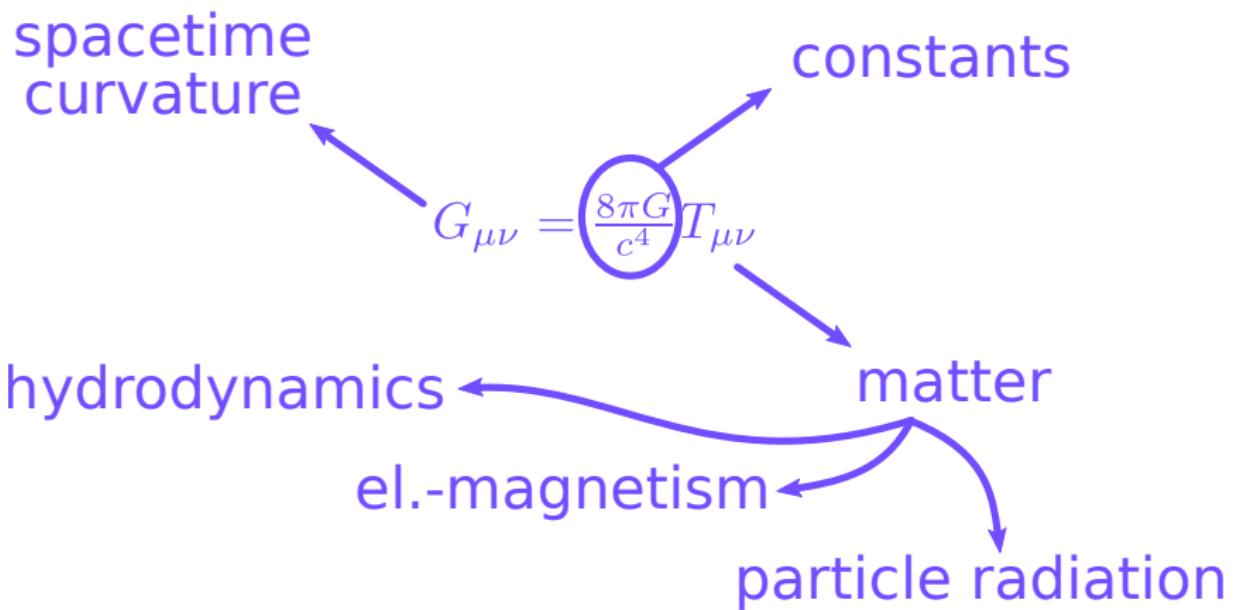
$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

spacetime
curvature constants

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

A diagram illustrating the components of the Einstein field equation. At the center is the equation $G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$. Three arrows point from the equation to three labels: "spacetime curvature" (top-left), "constants" (top-right), and "matter" (bottom-right).



$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

ADMBase

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

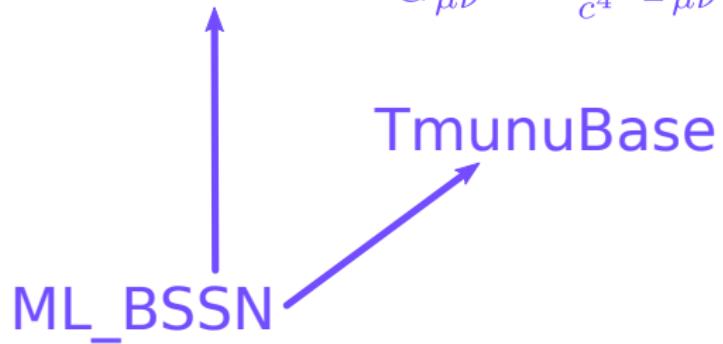
ADMBase

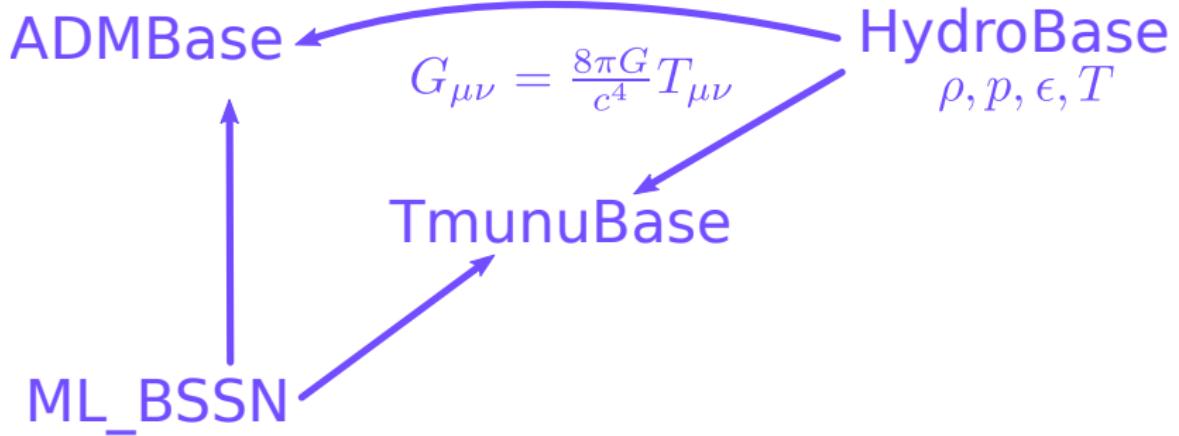
$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

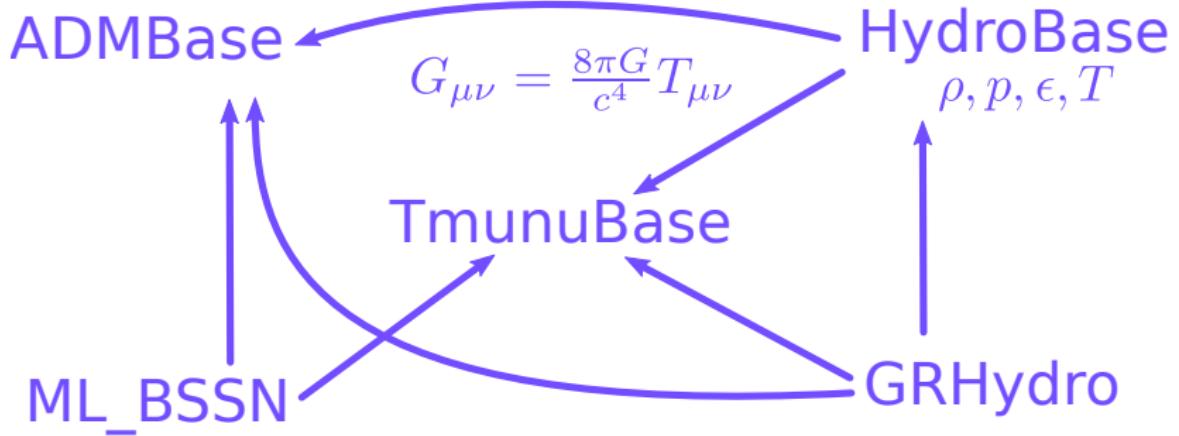
TmunuBase

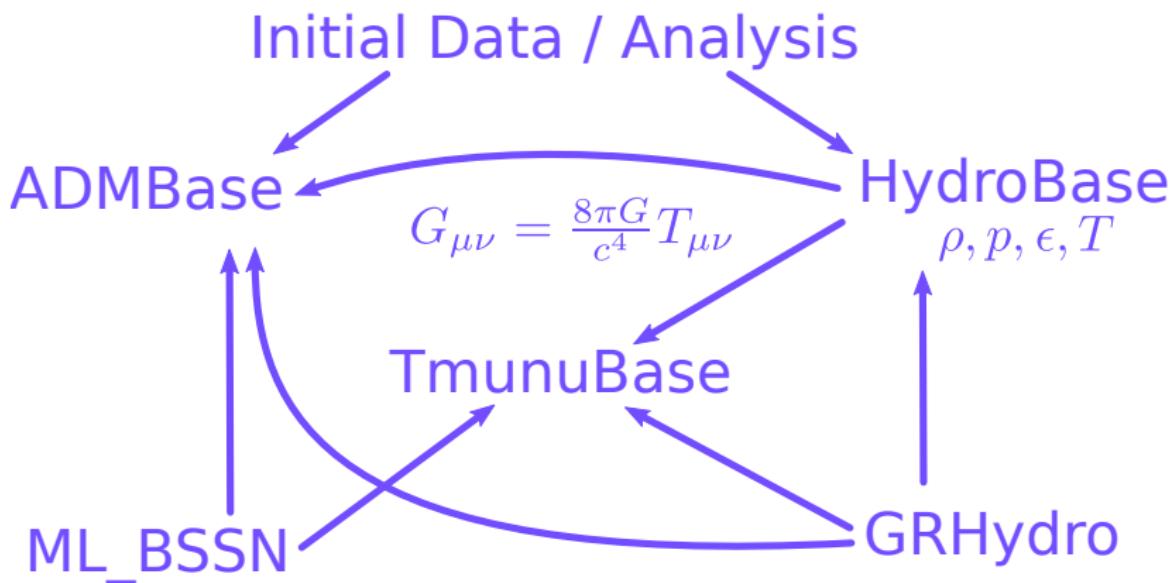
ADMBase

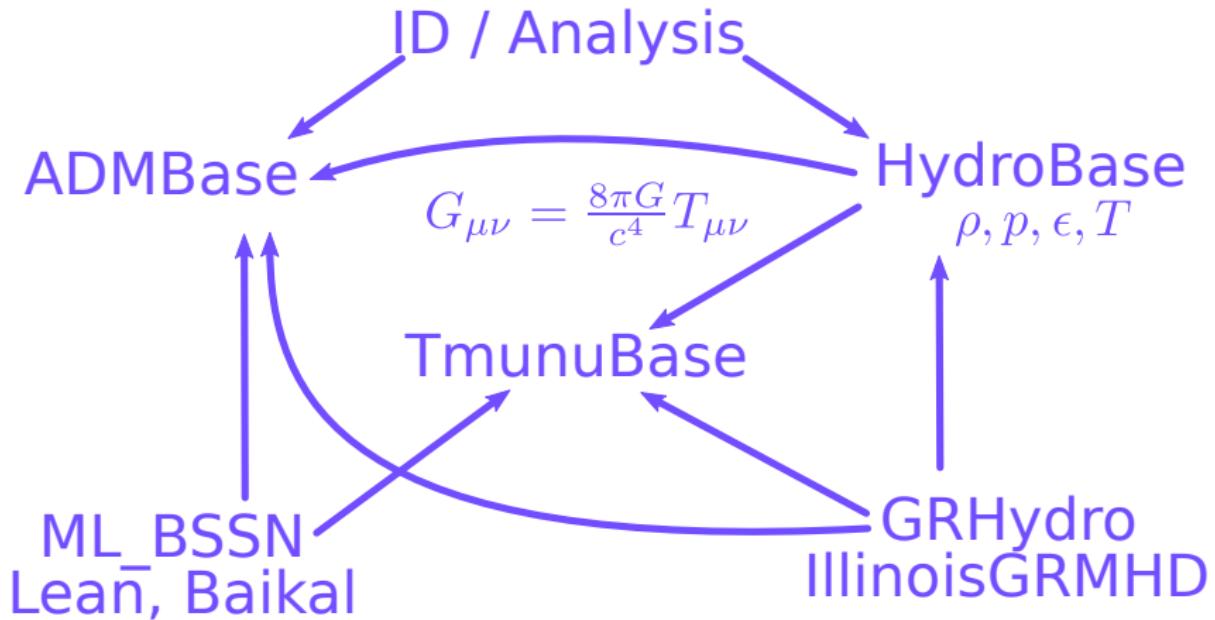
$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$











Guiding Principles

- Open, community-driven software development
- Separation of **physics** software and **computational** infrastructure
- Stable interfaces, allowing extensions
- Simplify usage where possible:
 - Doing science >> Running a simulation
 - Students need to know a lot about physics
(meaningful initial conditions, numerical stability, accuracy/resolution, have patience, have curiosity, develop a “gut feeling” for what is right ...)
 - Einstein Toolkit **cannot** give that, **however**:
Open codes that are easy to use allow to concentrate on these things!



Credits, Citations

In academics: citations, citations, citations!

For Einstein Toolkit:

- Open and free source
- No **requirement** to cite anything
- However: **requested** to cite
 - The DOI doi:10.5281/zenodo.3350841
 - Maybe the ET or Cactus papers
 - Some papers for the components list a few as well
 - List published on website and manage through publication database
- Soon: auto-generate list of citations during simulation run



Cutting Edge / Future

- New Driver Thorn: CarpetX
- New Spherical Coordinates Thorn (RIT)
- New Python Code Generator: Full thorn output from NRPy+
- Kerr background support in SelfForce1D



Recent

- PN based initial data and eccentricity reduction
- New Declarative Synchronization: Presync
- Python based simulation analysis: kuibit



Einstein Toolkit

- <http://www.einsteintoolkit.org/>
- Tools for high-performance computing in numerical relativity
- Open Source
- World-wide, open Community
- Used in high-end research

Supported By

The Einstein Toolkit is supported by
NSF 2004157/2004044/2004311/2004879/2003893, NSF
1550551/1550461/1550436/1550514 Any opinions, findings, and
conclusions or recommendations expressed in this material are those of the
author(s) and do not necessarily reflect the views of the National Science
Foundation.

