A very brief introduction to the Einstein Toolkit

(Open-source codes)

3rd HEL.A.S. Summer School and DAAD school for
Neutron Stars and Gravitational Waves

Theodoros Soultanis

IMPRS PhD student
at
Max Planck Institute for Astronomy, Heidelberg (MPIA)
Heidelberg Institute for Theoretical Studies (HITS)

Email:*theodoros.soultanis@h-its.org*

October 9, 2018

Outline

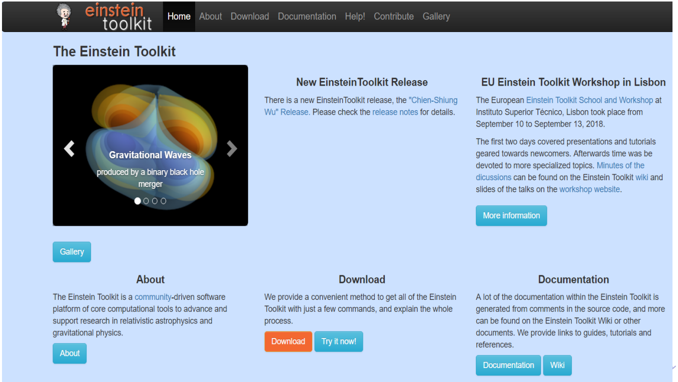
1. Einstein Toolkit: General information

2. Setting up a simulation: Jupyter notebook

Einstein Toolkit: What is it?

Einstein Toolkit, Link: https://einsteintoolkit.org/

- A three-dimensional code for the dynamical evolution of spacetimes in full General Relativity (BBH, BNS, Isolated stars)
- Open source, community-driven, and freely accessible infrastructure with highly active users
- Consists of diverse software components contributed by many different people and used by many more (210 members from 129 different groups and 36 countries)



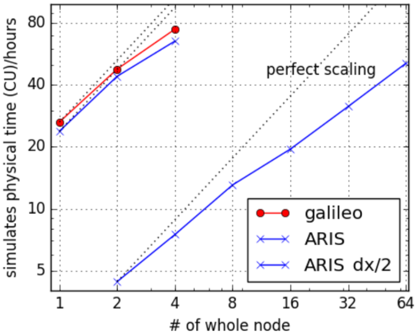
Einstein Toolkit: Structure

- Highly modular code which supports C, C++, FORTRAN, OpenCL, CUDA.
- Built on the Cactus framework, which provides the basic infrastructure for the numerical simulations (the modules in Einstein Toolkit's language are called Thorns)
- Thorns are developed independently (interchangeable with others)
  - Usually do not directly interact with each other
  - Each of them interacts with the Cactus framework
- Documentation is tricky

- Carpet is the driver module which is responsible for tasks such as Memory management, parallelization (OpenMP, MPI), input/output (IO)
- Implements an Adaptive Mesh Refinement

Cactus code: http://cactuscode.org/

Einstein Toolkit: Structure



Scalability tests for the Einstein Toolkit

- Scalability tested on HPC infrastructure (Galileo, ARIS)

Einstein Toolkit: 3+1 decomposition

- The McLachlan module implements the dynamical evolution of spacetime in the BSSNOK formalism
- The code is auto-generated from tensor equations via Kranc

Definitions

$$\tilde{\gamma}_{ij} = e^{-4\phi}\gamma_{ij}$$
$$e^{4\phi} = \gamma^{1/3}$$
$$A_{ij} = K_{ij} - \frac{1}{3}\gamma_{ij}K$$
$$\tilde{A}_{ij} = e^{-4\phi}A_{ij}$$
$$\tilde{\Gamma}^i = \tilde{\gamma}^{jk}\tilde{\Gamma}^i{}_{jk} = -\partial_j\tilde{\gamma}^{ij}$$

Time evolution

$$\frac{d}{dt}\tilde{\gamma}_{ij} = -2\tilde{A}_{ij}, \quad \frac{d}{dt}\phi = -\frac{1}{6}\alpha K, \quad \frac{d}{dt} = \partial_t - \mathcal{L}_\beta$$
$$\frac{d}{dt}K = -\gamma^{ij}D_jD_i\alpha + \alpha(\tilde{A}_{ij}\tilde{A}^{ij} + \frac{1}{3}K^2) + 4\pi\alpha(\rho + S)$$
$$\frac{d}{dt}\tilde{A}_{ij} = e^{-4\phi}\left(-(D_jD_i\alpha)^{TF} + \alpha(R_{ij}^{TF} - 8\pi S_{ij}^{TF})\right) + \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{il}\tilde{A}^l_j)$$
$$\partial_t\tilde{\Gamma}^i = -2\tilde{A}^{ij}\partial_j\alpha + 2\alpha\left(\tilde{\Gamma}^i{}_{jk}\tilde{A}^{kj} - \frac{2}{3}\tilde{\gamma}^{ij}\partial_jK - 8\pi\tilde{\gamma}^{ij}S_j + 6\tilde{A}^{ij}\partial_j\phi\right)$$
$$+ \beta^k\partial_j\tilde{\Gamma}^i - \tilde{\Gamma}^j\partial_j\beta^i + \frac{2}{3}\tilde{\Gamma}^i\partial_j\beta^j + \frac{1}{3}\tilde{\gamma}^{li}\partial_l\partial_j\beta^j + \tilde{\gamma}^{lj}\partial_j\partial_l\beta^i$$

"1+log" slicing

$$\frac{d\alpha}{dt} = -2\alpha K$$

Gamma driver

$$\partial_t\beta^i = kB^i$$
$$\partial_tB^i = \partial_t\tilde{\Gamma}^i - \eta B^i$$

Einstein Toolkit: Hydrodynamics

- The GRHydro module implements a grid based High Resolution Shock Capturing Scheme (HRSC) for the evolution of matter
- It supports GRHD, GRMHD (Valencia formulation)

- The EOS_Omni module provides an interface for all the EOS calls
  - polytropes, gamma law, piecewise polytropes, hybrid

- Hydro_RNSID generates initial configurations of rapidly (uniformly or differentially) rotating stars for the GRHydro code.
- So far, two kinds of rotation law profiles are available: uniform and differential, described the j-law profile:

$$\Omega_c - \Omega = \frac{1}{A^2 r_e^2} \left[\frac{(\Omega - \omega) r^2 \sin^2 \theta e^{-2\nu}}{1 - (\Omega - \omega)^2 r^2 \sin^2 \theta e^{-2\nu}} \right]$$

- Initial data for BNS models are generated externally LORENE, and subsequently imported to the Einstein Toolkit

Jupyter notebook session

Clone the repository containing useful material such as thornlist files, parameter files, configuration files, EOS tables:

```
git clone https://thsoulta@bitbucket.org/thsoulta/et_thessaloniki.git
```

- Download and build the Einstein Toolkit
  - Personal computer
  - Clusters
- Set up a simulation
  - Isolated rotating (uniformly, differentially) or non-rotating stars with Hydro_RNSID
  - Binary neutron star mergers with LORENE

References

- Einstein Toolkit, https://einsteintoolkit.org/
- T. W. Baumgarte and S. L. Shapiro, Phys. Rev. D 59, 024007 (1998), gr-qc/9810065.
- M. Shibata and T. Nakamura, Phys. Rev. D 52, 5428 (1995).
- T. Nakamura, K. Oohara, and Y. Kojima, Prog. Theor. Phys. Suppl. 90, 1 (1987)
- Stergioulas N., 1999, Rapidly rotating neutron star, https://www.gravity.phys.uwm.edu/rns
- LORENE, http://www.lorene.obspm.fr/
- Kranc, http://kranccode.org/
- Mösta, P., Mundim, B. C., Faber, J. A., et al. 2014, Classical and Quantum Gravity, 31, 15005.