Numerical Relativity: an overview

Thanasis Giannakopoulos

November 18, 2022

University of Nottingham
UK | CHINA | MALAYSIA

1. Some history
2. A recipe for Numerical Relativity
3. Current and future challenges
4. A plan for the next lectures

The Einstein Field Equations (EFE) in geometric units ($G = c = 1$):

$$R_{ab} - \frac{1}{2} g_{ab} R + \Lambda g_{ab} = 8\pi T_{ab}$$

How to solve the EFE?

- Exact solutions (e.g. time independence, high symmetry)
- Perturbative solutions (e.g. close to exact solutions, non-linear terms not important)
- Numerical methods (e.g. highly dynamical fields, strong non-linearities, lack of symmetry)

Numerical Relativity (NR):

The use of numerical methods to find approximate solutions to the EFE, which converge to the continuum solution with more computational resources.

Some history

- Late 1950s: Arnowitt-Deser-Misner (ADM) 3+1 decomposition
- 1960s: first numerics by Hahn and Lindquist
- 1970s: axisymmetric numerics by Čadež, Smarr, and Eppley (PhD students of DeWitt); York's 3+1 decomposition
- 1990s: binary black hole grand challenge project (3+1 codes, single black holes, mesh refinement, CACTUS), critical phenomena (Choptuik), stable characteristic codes
- 2000s: compact binaries
  -neutron star merger in BSSN[1] formulation (2003-2004)[2]
  -BBH merger (Pretorius using GHG, 2005),
  -more BBH mergers with different methods (Brownsville/Rochester and NASA Goddard, 2006)

See e.g. Sperhake 2015 for a review

[1] Baumgarte-Shapiro-Shibata-Nakamura (BSSN)

[2] Shibata, Taniguchi, Uryū; Marronetti, Duez, Shapiro, Baumgarte; Miller, Gressman, Suen

A recipe* for NR

Can we break down a successful NR simulation into its ingredients?

1. Physical problem
2. Formulation
3. Analysis of partial differential equations (PDEs)
4. Numerical methods
5. Implementation
6. Evaluate errors
7. Physical interpretation

* courtesy of David Hilditch

1. Physical problem

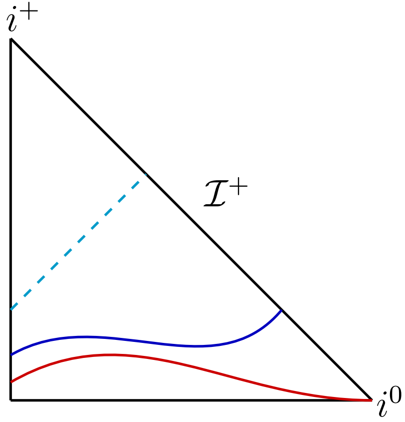
What do we want to study?

- Principle: gravitational collapse, black string instabilities
- GW astronomy: binaries of compact objects
- Astrophysics: accretion disks, core collapse
- Holography: heavy ion collisions, phase transitions

It is important to understand the limits of the problem described by other methods to cross-check, e.g. quasinormal modes of BH resulting from a binary merger, NR vs BH perturbations.

2. Formulation

Write the EFE in a way that can be used for numerical simulations: there is great freedom in this.

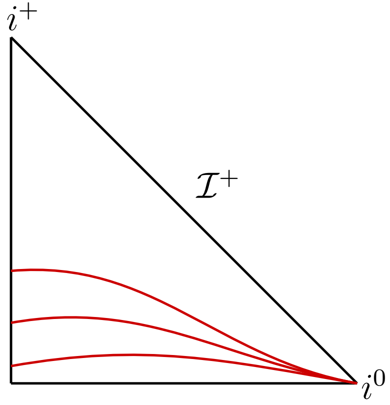


Spacetime slicing: spacelike, null, hyperboloid
Conformally compactified spacetime (CEFE)

2. Formulation: spacelike

More freedom:

- Free evolution
- Constrained evolution
- Constraint damping
- Evolved variables
  (e.g. ADM, BSSN)
- Gauge choice
  (e.g. maximal slicing, 1+log, harmonic)



3. PDE analysis

$$R_{ab} - \frac{1}{2} g_{ab} R + \Lambda g_{ab} = 8\pi T_{ab}$$

1. Choose a formulation
2. The PDE system we solve obtains a character (e.g. strongly hyperbolic, weakly hyperbolic)
3. Well-posedness of the PDE problem: there is a unique solution that depends continuously on the given data
4. If PDE problem is ill-posed GOTO 1 (e.g. change gauge)

3. PDE analysis

$$R_{ab} - \frac{1}{2} g_{ab} R + \Lambda g_{ab} = 8\pi T_{ab}$$

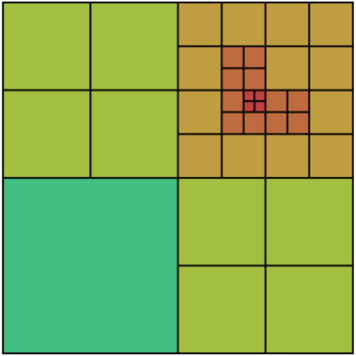
1. Choose a formulation
2. The PDE system we solve obtains a character (e.g. strongly hyperbolic, weakly hyperbolic)
3. Well-posedness of the PDE problem: there is a unique solution that depends continuously on the given data
4. If PDE problem is ill-posed GOTO 1 (e.g. change gauge)

Numerical Relativity (NR):

The use of numerical methods to find approximate solutions to the EFE, which converge to the continuum solution with more computational resources.

4. Numerical methods

- Method of lines
- Finite differencing
- Spectral methods
- Numerical boundary conditions
- High-resolution shock capturing
- Artificial dissipation
- Mesh refinement



Schematic example of a refined grid
Suárez Fernández et. al: arXiv:2205.04379v1

5. Implementation

We need a code that implements the chosen numerical method.

- Is there relevant open source software?
  (e.g. Einstein Toolkit, GRChombo, SpECTRE, NRPy+)
- If not, develop your own code
  (programming language, libraries, code design and modularity)
- Optimization (memory, speed, scalability and parallelization)
- **Test**, fix bugs, create documentation

6. Evaluate errors of the numerical simulation

Perform convergence tests: Solve the same PDE problem with increasing resolution, and inspect how the error behaves.

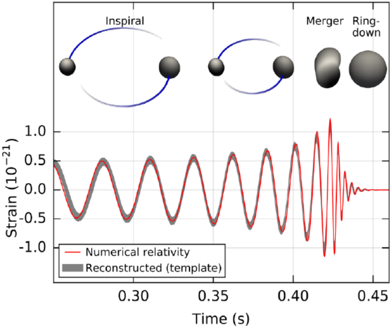
- Does the error converge to zero with increasing resolution?
- What is the convergence rate?
- If convergence is lost or the rate is not the expected one:
  - revisit formulation, numerical methods, implementation
  - increase resolution (maybe you need more computational resources)
- Compare against known solutions (e.g. exact, perturbative)

7. Physical interpretation of the results

We need to manipulate the data (visualization, postprocessing).

For example:

- Gravitational waveforms
- Event horizons
- Curvature invariants



NR waveform modeling GW150914
Abbott et al: arXiv:1602.03837v1

What's next?

- 2 body problem:
  - more accurate gravitational waveforms (code efficiency and accuracy, formulation that include $\mathcal{I}^+$)
  - add features (black hole spin, environmental effects, neutrino transfer in neutron stars)
- Critical phenomena in gravitational collapse (fully 3D simulation)
- Cosmological scenarios (e.g. early universe phase transitions via holography)
- Go beyond GR (e.g. single compact objects, collapse, 2 body problem)

Plan

Lectures 2 and 3

- PDE analysis: hyperbolicity and well-posedness
- Spacelike formulations (ADM, BSSN, GHG)
- Characteristic formulations

Lecture 4

- Numerical methods (method of lines, finite differences)
- Toy models: implementation and convergence

Plan

Lectures 2 and 3

- PDE analysis: hyperbolicity and well-posedness
- Spacelike formulations (ADM, BSSN, GHG)
- Characteristic formulations

Lecture 4

- Numerical methods (method of lines, finite differences)
- Toy models: implementation and convergence

Thank you!