Evolution Problem in Numerical Relativity

Jordan NICOULES

jordan.nicoules@obspm.fr

LUTH STUDENTS' DAY

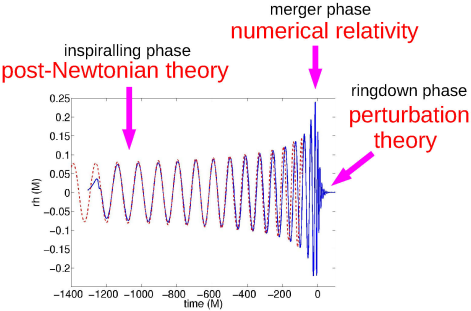
March 10th, 2021

Laboratoire de l'Univers et de ses Théories

Context

Context: Numerical Relativity

- Numerical Relativity: Solve Einstein's equations (or more) numerically.
- Black Hole Binary Grand Challenge (90s): achieve several orbits and merger to generate waveforms for Gravitational Waves emission.



Picture from http://www.iap.fr/actualites/laune/2016/OndesGr/forme_onde_an.jpg

Context: Numerical Relativity

- Numerical Relativity: Solve Einstein's equations (or more) numerically.
- Black Hole Binary Grand Challenge (90s): achieve several orbits and merger to generate waveforms for Gravitational Waves emission.
- In 2005, first successful simulation by Pretorius (https://doi.org/10.1103/PhysRevLett.95.121101)
- In 2015, first successful detection of GW (https://doi.org/10.1103/PhysRevLett.116.061102)

Context: A complex recipe

- To obtain a successful computation, you need a handful of ingredients, mixed together in a delicate and sometimes empirical manner.
- Put it in the oven of High Performance Computing for thousands of CPU hours.
- See for example Brügmann in *Science* for a short review https://science.sciencemag.org/content/361/6400/366

An overview of a few ingredients

- System of equations:
  - Choice of theory
  - Choice of dynamical variables
  - Choice of the order of equations (in time, in space)
  - Constraint damping...
- Discretization, integration scheme, numerical methods, parallelization.
- Gauge conditions, boundary conditions, initial data.
- Management of the physical objects (horizons, shocks) and extraction of relevant data (gravitational waveform).

This PhD

- **First:** Start with standard methods and implement them in Kadath, to have a working code.

- **Second:** Apply them to new physical systems (AADS spacetimes, scalar-tensor theories...) and/or explore less standard methods (constrained evolution schemes, time spectral methods).

Brief introduction to Kadath and my contribution

Kadath

- **Kadath library**: numerical code (C++) developed at LUTH which implements spectral methods and a Newton-Raphson scheme to solve non-linear PDEs.
- Can be used to study stationary systems or generate initial data for evolution for example.
- Very flexible in terms of geometry, equations to solve, designed with NR in mind, but no evolving systems yet (hence PhD).

What's new?

- Solve hyperbolic systems of equations with
  - a 4th-order Runge-Kutta scheme ;
  - an adaptive step Runge-Kutta scheme (Dormand-Prince method).
- Equations given as $\partial_t u = \dots$ with u being one of the dynamical variables, for bulk, boundary and matching equations.
- Implemented for spherical types of spaces (nucleus and shell domains) but easily transferable to other types of domains and spaces when needed.
- Save configurations with a custom frequency (e.g. every 10 time steps) or stop the time scheme with numerical or physical criteria.

A first application: the scalar wave

Why the scalar wave?

- Simple and controlled toy-model to proof test the code.
- The Einstein equations in Generalized Harmonic Gauge have a wave-like structure.
- Allows to test and familiarize with various aspects independently:
  - 1D/3D
  - various kinds of boundary conditions
  - constraint damping
  - penalty methods
  - self-interaction.
- Illustration of a few items here.

System of equations (1D)

- $$\frac{\partial^2 N}{\partial t^2} = c^2 \frac{\partial^2 N}{\partial r^2}$$

- First-order reduction: Use the space and time derivatives (resp. G and V) as independent variables

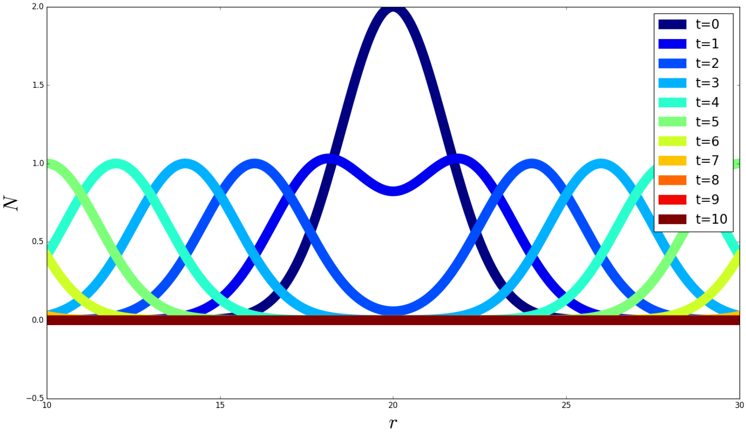
$$\begin{cases} \partial_t N = & cV \\ \partial_t G = & c\partial_r V \\ \partial_t V = & c\partial_r G \end{cases}$$

- Free evolution: the constraint $C = G - \partial_r N$ is not evolved.

(Rem: $C(t=0) = 0$ and $\partial_t C = \partial_t G - \partial_t(\partial_r N) = c\partial_r V - c\partial_r V = 0$)

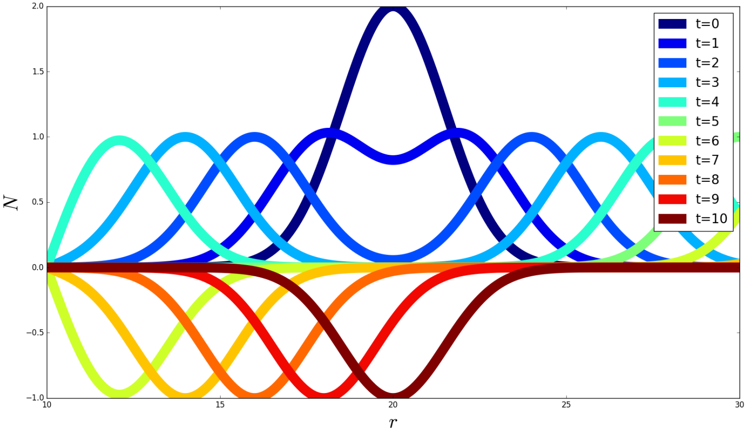
$\Rightarrow$ **It can be used as a measure of the numerical convergence.**

Illustration



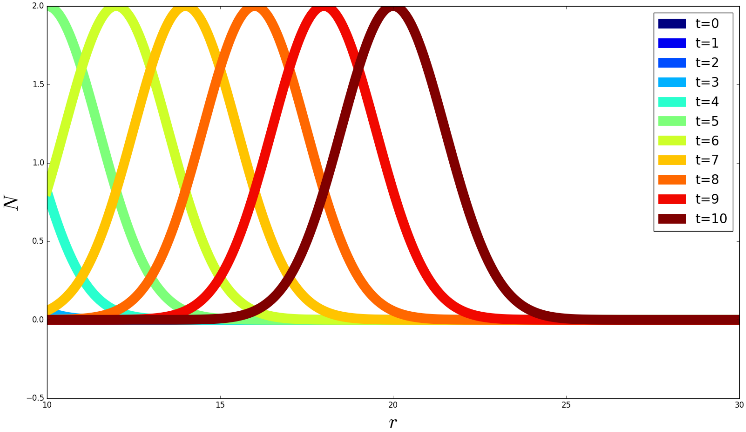
Outgoing wave

Illustration



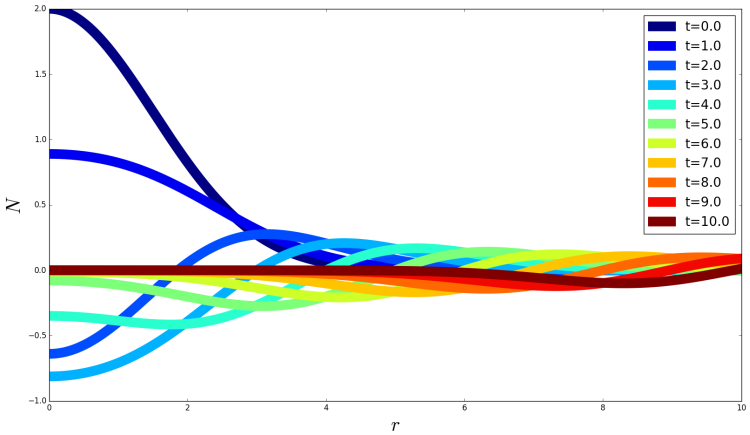
Wave reflected on the inner boundary

Illustration



Time-dependent source on the inner boundary

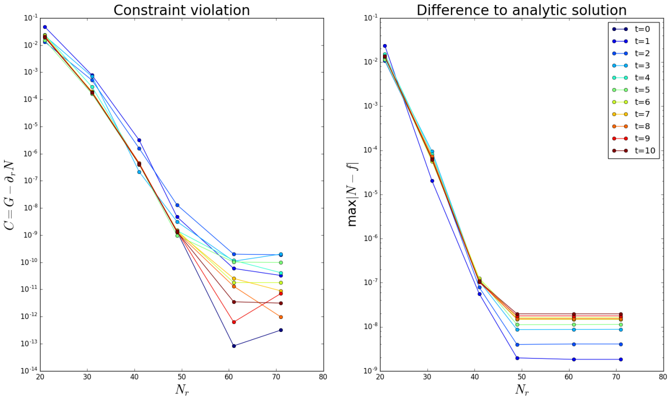
Illustration



3D, spherical symmetry, outgoing wave

Some convergence results (1D)

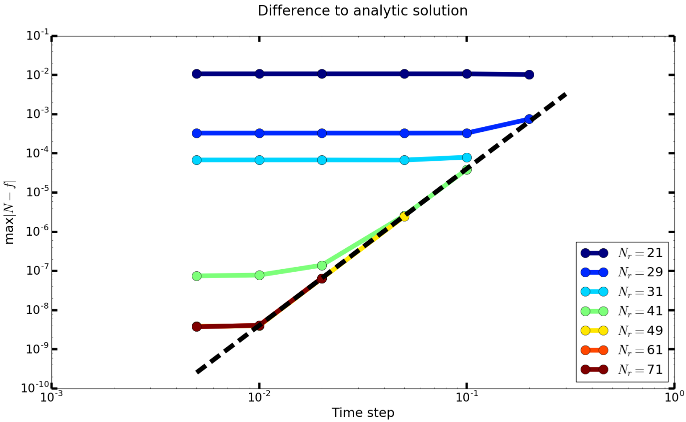
Spectral convergence



Outgoing wave with $h = 0.01$

Some convergence results (1D)

Time-step convergence



Convergence with respect to h ($t = 2$)

Penalty method

- Idea: No need to impose exact boundary conditions on the approximate (aka discretized) system.
  $\Rightarrow$ The boundary conditions are included in the bulk equations as a penalty term.

$$\mathrm{EOM}(u) + \kappa Q(x) \cdot \mathrm{BC}(u) = 0$$

- Schematically, $\kappa \xrightarrow[N \to \infty]{} +\infty$ and $Q(x) = \delta(x - x_{\mathrm{BC}})$

Penalty method

- Yields more stable schemes for spectral methods, allows more variety on boundary types and conditions (see for example Hesthaven https://doi.org/10.1016/S0168-9274(99)00068-9).
- Reduces the number of equations to compute in Kadath.
- Following Taylor *et al.*, way to go for second-order-in-space systems (https://doi.org/10.1103/PhysRevD.82.024037).
  $\Rightarrow$ Reduces the number of variables, equations and constraints.
- Works in Kadath for the scalar wave, for boundary and matching conditions, 1D/3D, 1st and 2nd order in space.

Current and future work

Conclusion

- Achieved work:
  - Implement a solver for evolution equations in Kadath
  - Validate it with the scalar wave
  - Use this toy-model to get familiar with various ingredients required for the evolution problem in GR.
- Current work: Compute the evolution of a GR system.
  Initial data consisting in gravitational waves (Teukolsky wave).
- Future work: Apply the code to new systems (e.g. stability of geons in AAdS spacetime, stability of black holes in modified gravity).

Thank you!