First applications of a new evolution code based on the KADATH library

Rencontre des groupes de travail Formes d'onde et Tests de la Relativité Générale

IPhT, CEA Saclay – December 9, 2021

Jordan NICOULES

jordan.nicoules@obspm.fr

Laboratoire Univers et 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.
- In 2005, first successful simulation by Pretorius. (Pretorius (2005))
- In 2015, first successful detection of GW. (Abbott *et al.*, LIGO Scientific Collaboration and Virgo collaboration (2016))

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. (Brügmann (2018))

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).

Project

- Release a new independent, free and open-source evolution code.
- State-of-the-art codes for GW templates are the joint work of many people for many years, and still require huge numerical resources.
  $\Rightarrow$ Not competitive on this aspect, at least right from the beginning.
- **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).

Kadath library

- Numerical code (C++) developed at LUTH which implements **spectral methods** and a Newton-Raphson scheme to solve non-linear PDEs.
  website: https://kadath.obspm.fr/home/
- Can be used to study stationary systems or generate initial data for evolution for example.
  $\Rightarrow$ See recent release of FUKA branch for Initial Data. (Papenfort *et al.* (2021))
- Very flexible in terms of geometry, equations to solve, designed with NR in mind (within the 3+1 formalism), but no evolution schemes.

What's new?

- Solve hyperbolic systems of equations with
  - a 4th-order Runge-Kutta scheme ;
  - an adaptive step Runge-Kutta scheme (Dormand-Prince method).

  $\Rightarrow$ Free evolution.

- 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 equation

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)

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

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

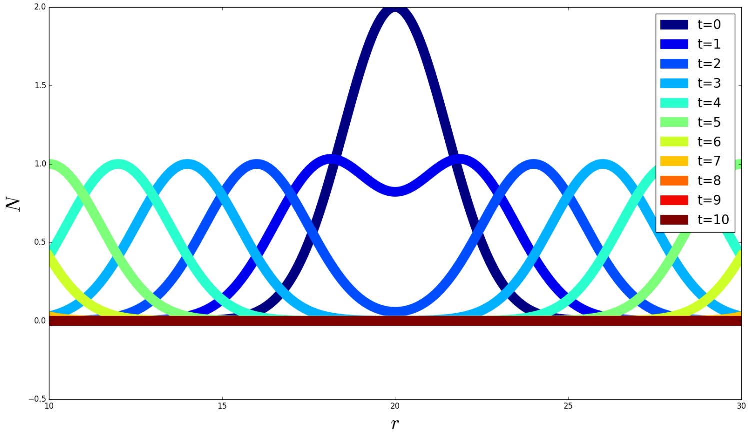
$$\begin{cases} \partial_t N = & cV \\ \partial_t G = & c\partial_r V \\ \partial_t V = & c\partial_r G \end{cases}$$

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

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

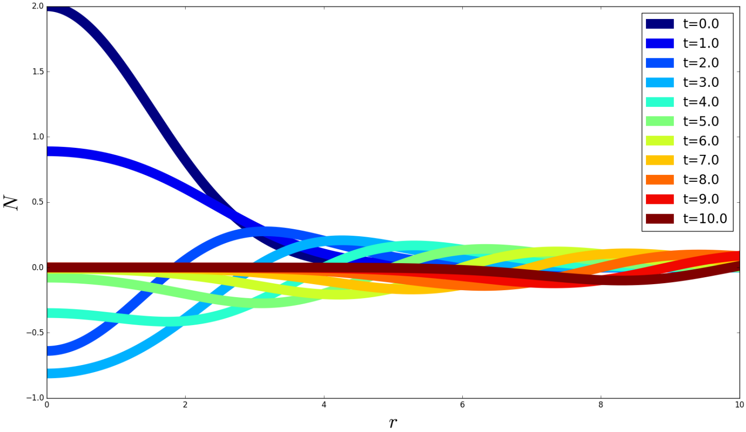
$\Rightarrow$ **It can be used as a measure of the numerical convergence.**

Illustration



Outgoing wave.

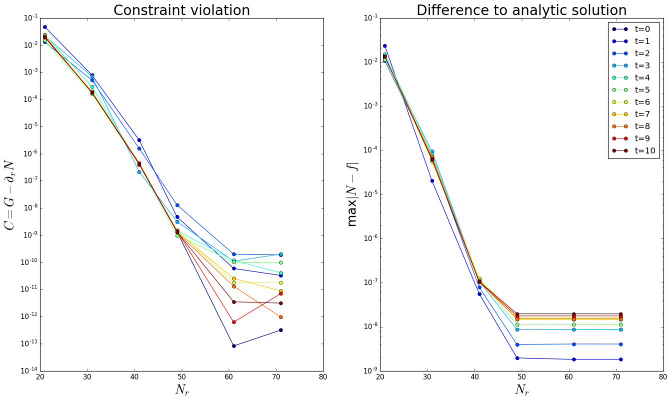
Illustration



3D, spherical symmetry, outgoing wave.

Some convergence results (1D)

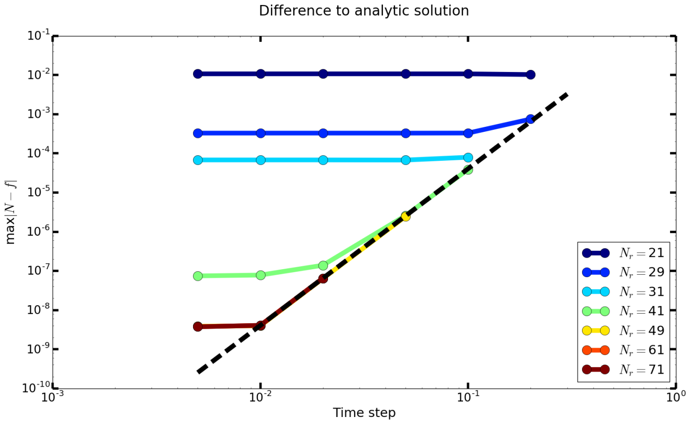
Spectral convergence



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 (2000)).
- Reduces the number of equations to compute in Kadath.
- Following Taylor *et al.*, way to go for second-order-in-space systems (Taylor *et al.* (2010)).
  $\Rightarrow$ Reduces the number of variables, equations and constraints.
- Works for boundary and matching conditions, 1D/3D, 1st and 2nd order in space.

Beyond spherical symmetry

- When introducing angular dependency (e.g. with initial data proportional to spherical harmonics Y_2, $Y_4, Y_{2,2}$), some instabilities develop at the origin and grow exponentially.

- **Consequence:** Fields become multivalued and diverge!
(there are multiple numerical points representing the physical origin and they don't agree anymore)

- I applied two main remedies:
  1. Regularization before computing some operations, such as the division by r.
  2. Filtering the angular coefficients of the RHS in the domain containing the origin.

A GR system:
Teukolsky wave

Initial data

- Goal: Proof test the code with a GR system similar to the wave equation.
  $\Rightarrow$ Propagation of a gravitational wave (without source).

- Initial Data obtained with the XCTS method (Extended Conformal Thin Sandwich) from an even-parity axisymmetric Teukolsky wave ($m = 0$), with a non-centered Gaussian-like profile.
  (Teukolsky (1982) , Hilditch *et al.* (2016))

- Starting from freely specifiable quantities, the constraints are solved with Kadath. This maintains qualitative properties.
  (Pfeiffer *et al.* (2005))

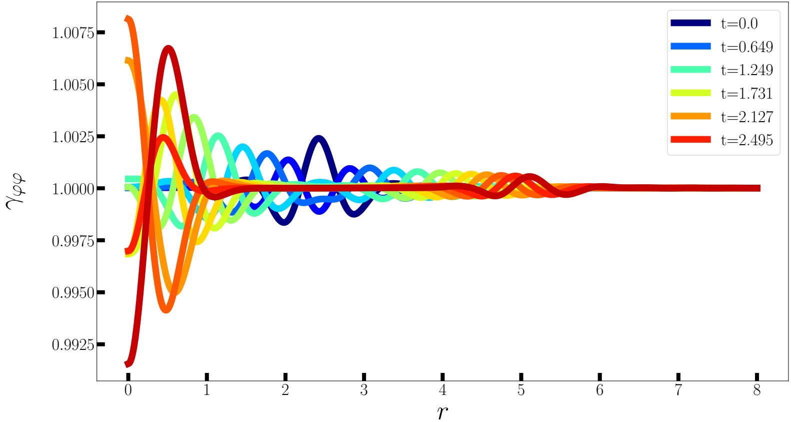
Evolution system

- BSSN formulation (Baumgarte-Shapiro-Shibata-Nakamura) with a first-order reduction for space derivatives and constraint damping. (Brown *et al.* (2012))
- Gauge conditions: 1+log slicing and Gamma-driver (*puncture gauge*).
  $\Rightarrow$ Evolution equations for the *lapse* and *shift*.
- Outer boundary conditions: simple radiative (Sommerfeld) conditions on all dynamic fields, imposed strongly.

Evolution system

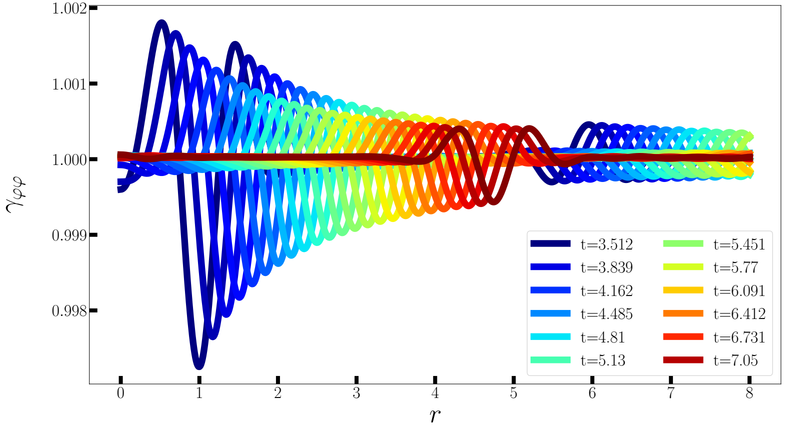
- The numerical domain consists in a nucleus and three spherical shells $\Rightarrow$ We need to patch the domains together.
- This is done with **penalties** on characteristic fields (depending on their velocities), which are given in the reference.
- We use spherical coordinates, and can use either a Cartesian or orthonormal spherical tensor basis.
- With the flexibility of Kadath, we could use other formulations as well! (provided they are in 3+1 form)

First results (preliminary)



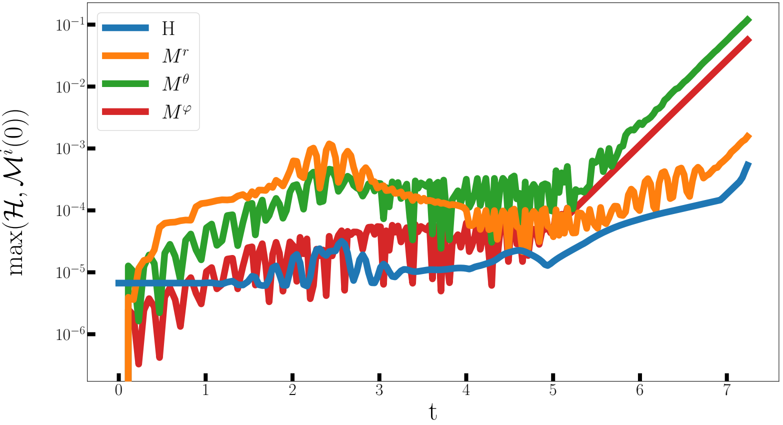
Evolution of the radial profile of $\gamma_{\varphi\varphi}$ (beginning).

First results (preliminary)



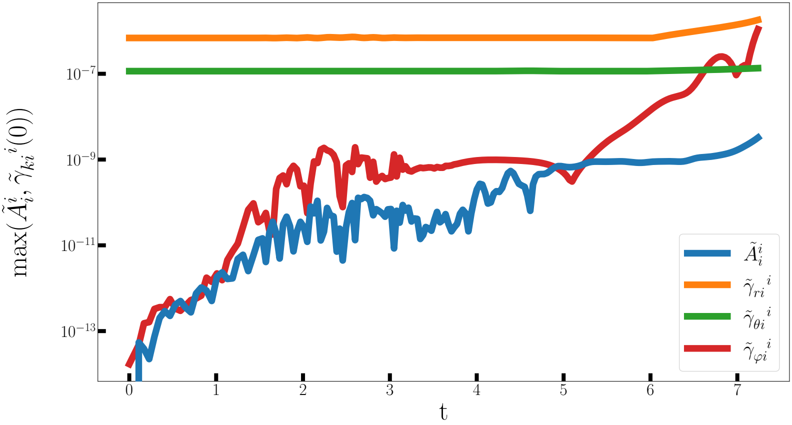
Evolution of the radial profile of $\gamma_{\varphi\varphi}$ (end).

Measure of the error: GR constraints



Evolution of the maximum of the Hamiltonian and momentum constraints in the central domain.

Measure of the error: Algebraic constraints



Evolution of the maximum of the traces $\tilde{A}_i^i$ and $\tilde{\gamma}^{ij}\tilde{\gamma}_{kij}$ in the central domain.

Challenges

- With the adaptive time-step, the integration stops because the time-step is reduced too much.
- A careful examination of all fields shows some unphysical behavior near the origin and the boundary with the first shell, as well exponentially growing instabilities on variables expected to vanish.
  $\Rightarrow$ The simulation works fine until they become dominant.
- Even though aforementioned regularization work significantly improved the stability and accuracy, further work is needed.
  $\Rightarrow$ Investigation in progress on the scalar wave equation.

Conclusion and future work

Work achieved

1. Implementation of time evolution schemes in Kadath, preserving its modularity.
2. Validation on a simple wave equation toy model.
3. Regularization techniques related to the numerical treatment of the physical origin.
4. Preliminary evolution of a Teukolsky wave in the context of GR.

Novelty

Combination of the ingredients:

- Free evolution of an axisymmetric **Teukolsky wave**
- with the **first-order BSSN** system in puncture gauge,
- in **spherical coordinates** including the origin, and a spherical tensor basis,
- using spatial **multi-domain pseudo-spectral methods** and matching through **penalties** on characteristics
- and an **adaptive time-step** time integration scheme.

Prospects

- Cure remaining instabilities and finalize the stable evolution of the Teukolsky wave, including higher amplitude to test the full non-linear regime.
- Apply the code to new physical systems, including in modified gravity.
- Optimize and possibly parallelize the code (work in progress).
- Longer term:
  - Take advantage of the flexibility of the library to investigate less standard methods (e.g. time spectral).
  - Enrich the library with an apparent horizon finder, more user-friendly and automatic treatment of domain matching, gravitational waveform extraction, various utilities...

Thank you!

Boundary conditions: Outgoing wave

- Characteristic structure:

$$\partial_t \begin{pmatrix} N \\ G \\ V \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -c \\ 0 & -c & 0 \end{pmatrix}}_{A_r} \partial_r \begin{pmatrix} N \\ G \\ V \end{pmatrix} = \begin{pmatrix} cV \\ 0 \\ 0 \end{pmatrix}$$

- What matters is the principal part (left-hand side), more precisely the characteristic fields, given by (left) eigenvectors of A_r:

$$u^0 = N, \qquad v^0 = 0$$
$$u^{\pm} = G \mp V, \qquad v^{\pm} = \pm c$$

- Impose boundary conditions only on incoming fields, namely u^+ at the inner boundary and u^- at the outer boundary:

$$\partial_t u^{\pm}(t, r_{\mp}) = 0$$

Boundary conditions: Outgoing wave

- $$\begin{cases} u^+ = & G - V \\ u^- = & G + V \end{cases}, \qquad \begin{cases} G = & \frac{u^- + u^+}{2} \\ V = & \frac{u^- - u^+}{2} \end{cases}$$

- For instance, at the inner boundary, this yields

$$\begin{cases} (\partial_t u^0 = & \partial_t N) \\ \partial_t u^+ = & 0 \\ \partial_t u^- = & \partial_t G + \partial_t V = c(\partial_r V + \partial_r G) \end{cases}$$

and so

$$\begin{cases} (\partial_t N = & cV) \\ \partial_t G = & \dfrac{\partial_t u^- + \cancel{\partial_t u^+}}{2} = \dfrac{c}{2}(\partial_r V + \partial_r G) \\ \partial_t V = & \dfrac{\partial_t u^- - \cancel{\partial_t u^+}}{2} = \dfrac{c}{2}(\partial_r V + \partial_r G) \end{cases}$$

Boundary conditions

- We can also choose other types of conditions, such as reflected wave/fixed end point ($\partial_t V = 0$), a time-dependent source...

- Characteristic fields are also relevant for domain matching, which is done with penalties, for example at the outer boundary of domain 1

$$\partial_t u^- = c\partial_r u^- + \kappa\delta(r - r_{12}) \left[u^-_{(2)}(r_{12}) - u^-_{(1)}(r_{12}) \right]$$

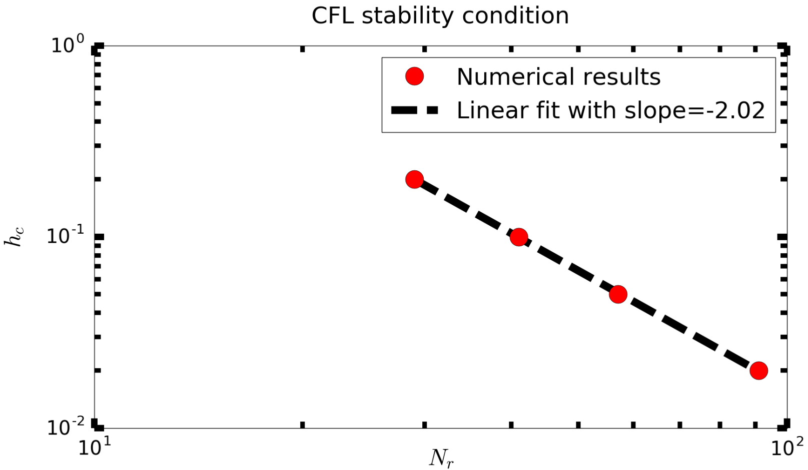
$$\partial_t u^+ = -c\partial_r u^+$$

and then we transform back to dynamic fields.

CFL stability condition

- CFL stability condition: $\Delta t \leq \mathrm{cst} \cdot \Delta x$
- For spectral methods especially Chebyshev collocation points, $x_i = \cos\left(\frac{\pi i}{N}\right)$ and $\Delta x \propto \frac{1}{N^2}$, hence $\Delta t_{crit} \propto \frac{1}{N^2}$.

CFL stability condition

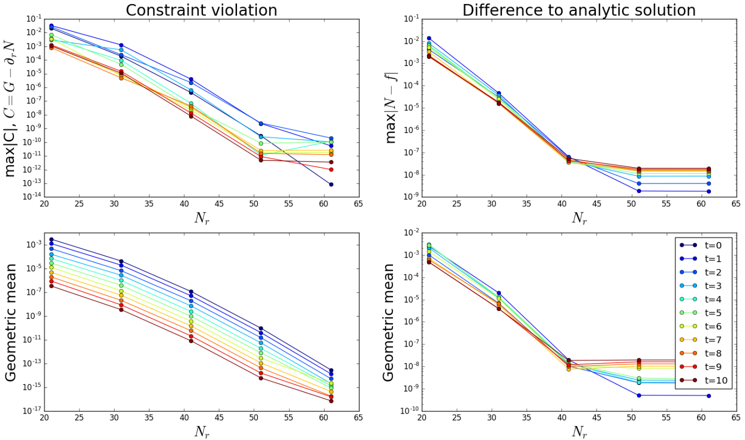


Critical time step versus number of points

Constraint damping

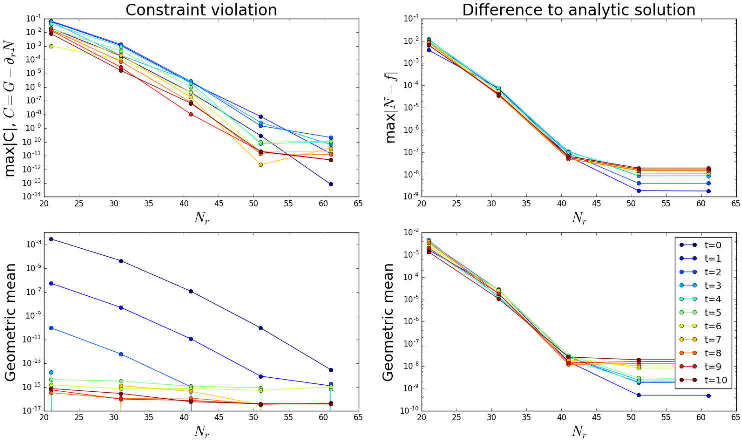
- Mathematically, if $C(t=0)=0$ then $\forall t, C=0$
- As we have seen, not true numerically
  $\Rightarrow$ Add terms proportional to the constrain in equations so that $\partial_t C = -C/\tau$
  $\Rightarrow$ If $C=0$ both systems have the same solution.
- In our case, use $\begin{cases} \partial_t N = & cV & \\ \partial_t G = & c\partial_r V & -C/\tau \\ \partial_t V = & c\partial_r G & \\ (C = & G - \partial_r N) & \end{cases}$
- This does not change the characteristic structure.

Constraint damping (convergence)



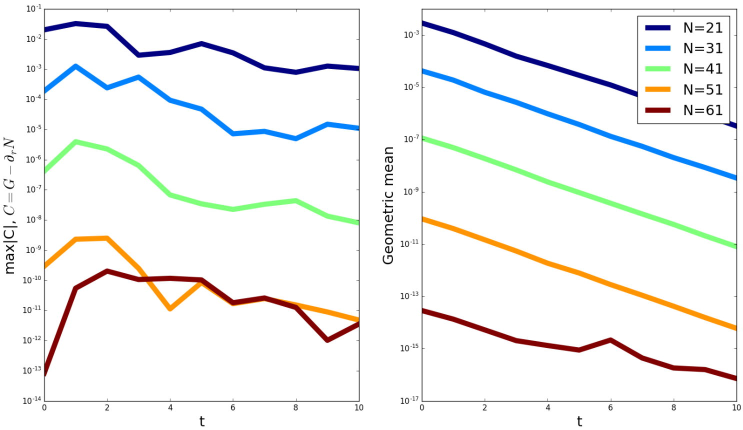
Convergence for $\tau = 1$

Constraint damping (convergence)



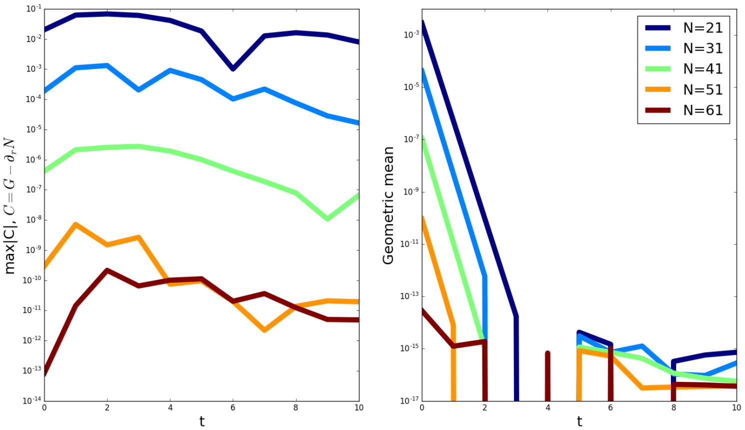
Convergence for $\tau = 0.1$

Constraint damping (time evolution)



Evolution of the constraint violation for $\tau = 1$

Constraint damping (time evolution)



Evolution of the constraint violation for $\tau = 0.1$

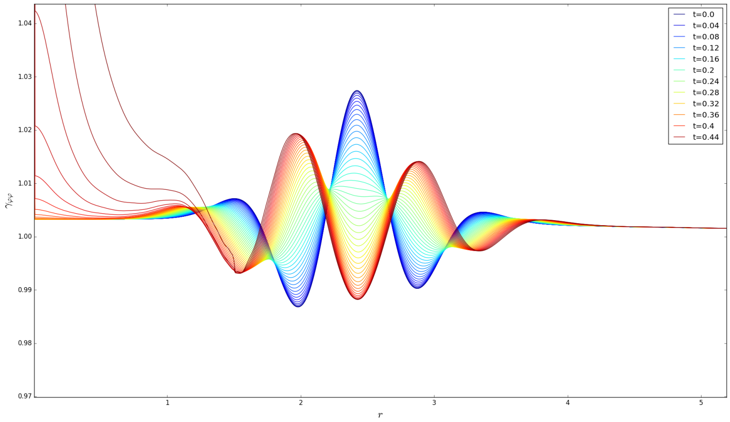
Initial data

XCTS yields a solution to the full non-linear system by solving the constraints

- $\tilde{\gamma}_{ij}$ given by the Teukolsky wave (solution to the linearized equations) with $F(r) = \frac{A}{2\sigma} r^9 \left(\exp\left(-\left[\frac{r - r_0}{\sigma} \right]^2 \right) + \exp\left(-\left[\frac{r + r_0}{\sigma} \right]^2 \right) \right)$ and $A = 10^{-8}$, $r_0 = 2$, $\sigma = 0.5$.
- $\partial_t \tilde{\gamma}^{ij} = 0$, $K = 0$, $\partial_t K = 0$, $\beta^i = 0$
- in vacuum.

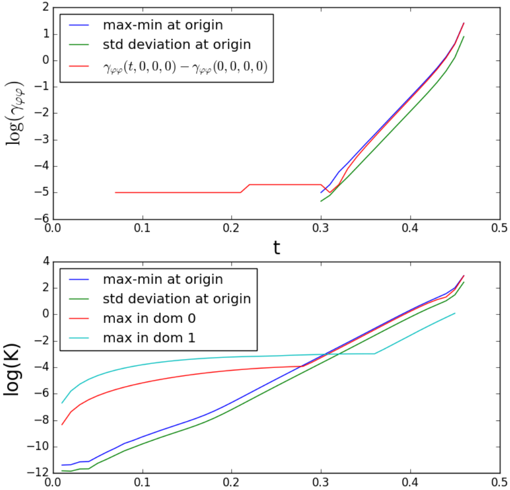
- Teukolsky (1982) , Hilditch *et al.* (2016)

Divergence at the origin in the FOBSSN system



Evolution of the radial profile of $\gamma_{\varphi\varphi}$ along (Ox)

Evidence of the instability



Evolution of the error at the origin for K and $\gamma_{\varphi\varphi}$

Regularization

The division by x in the nucleus requires to remove the finite part of the function:

$$f(x) = f(0) + xg(x) = a_0 T_0(x) + \cdots + a_N T_{2N}(x)$$

The usual operator computes the coefficients of $\dfrac{f(x) - f(0)}{x}$ from the a_i's (it's like acting on the first coefficient, as $T_0(x) = 1$).

We regularize by subtracting the value at 0 first, acting on the last coefficient (*tau*-method inspired):

$$f(x) \leftarrow f(x) - f(0)\frac{T_{2N}(x)}{T_{2N}(0)}$$