Introduction to Numerical Relativity: Exercise 1

Alex Vano-Vinuales

Autumn semester 2021

Write a simple program to numerically evolve the 1D wave equation,

$$\partial_t^2 \psi = \partial_x^2 \psi,$$

with initial conditions

$$\psi_0(x) = e^{-\frac{(x-x_0)^2}{2\sigma^2}} \quad \text{and} \quad \partial_t \psi_0(x) = k\, \partial_x \psi_0.$$

Steps to take:

1. introduce the variable $\pi = \partial_t \psi$ to write the system in first order in time and second order in space (FT2S) form (optionally you can also implement it in fully 1st order form);
2. discretise the spatial derivatives (using second order finite differences);
3. use the Method of Lines with a 4th order Runge-Kutta method to advance the system in time.

What to test?

1. Implement the above with periodic boundary conditions, and see what happens for $k = 0, 1, -1$. Interpret the results relating them to the mode propagation.
2. Also implement outgoing boundary conditions and see what happens.
3. Primordial in numerical codes: check convergence! Check pointwise convergence (at all i spatial gridpoints) at each time step and plot the convergence order over time using the L^2 norm ($p = \log_f \left(\frac{\sqrt{\sum_i (u_{f2_h} - u_{fh})^2}}{\sqrt{\sum_i (u_{fh} - u_h)^2}} \right)$).
4. Transform your wave equation implementation to the equations we labeled as "[Test code exercise 1/2]" ($t-x$ Laplace equation and weak hyperbolic model, respectively) in the first PDEs class and try running and checking convergence. What happens?

Notes:

- You can use any "reasonable" programming language. Please do not use Mathematica, Maple, Matlab, or other non-free tools for the code - it is fine to use them for the post-processing and creation of plots. Also please do not use any external libraries, as that may make it difficult for the instructor to test the code.
- Implement the Runge-Kutta method yourself - have fun!
- This code will be used to test other equations, so take into account when coding that the number of variables and RHSs will be increased later on.
- Comment the code in English while you write it.
- For the output, the easiest is to write it as ASCII file(s) and then load it with your software of choice (ygraph, GNUplot, Mathematica, python, ...) to visualize / make plots.
- Makefile: it is a good idea to have a Makefile to compile the code (it's actually necessary for larger codes).
- Make sure you use version control software (git, svn, ...) while developping your code.
- Ask the instructor if anything is unclear!