Gravitational Waves: Assignment 3
Numerical simulation of the 1D wave equation

Numerical relativity codes are complicated and usually a few hundred thousand code lines long. Therefore, we restrict us to a much simpler non-relativistic problem: the wave equation.
Let us consider the one dimensional wave equation for a scalar field u

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \tag{1}$$

with $c = \text{const}$.

Considering that numerical relativity simulations are computational challenging and that high level languages as python are usually not used because of their performance, we would prefer that basic languages as C, C++, or Fortran are used. However, you can also use high level languages in case you do not have experience with other programming languages.

1. Simple discretization
   (a) Rewrite the wave equation assuming second order time and space derivatives.
   (b) Use a 1+1-dimensional grid and evolve the wave equation. Use as initial conditions $u(x, t = 0) = \exp(-(x - 10)^2)$ and $u(x, t = -\Delta t) = \exp(-(x - c\Delta t - 10)^2)$. Impose periodic boundary conditions. Make sure that the code does not save unnecessary time steps which would increase unnecessarily the memory footprint of the code.
   (c) Check the convergence of the numerical method by varying the resolution and performing a self-convergence test. Plot the convergence order over time.
   (d) Write a short documentation about your results interpreting the most important aspects.

2. Method of lines
   (a) Rewrite the equation with the help of the method of lines. Use second order spacial derivatives. Use the 4th order explicit Runge-Kutta Method for the time integration.
   (b) Use a 1+1-dimensional grid with periodic boundary conditions and evolve the wave equation. Use as initial conditions $u(x, t = 0) = \exp(-(x - 10)^2)$ and $u(x, t = -\Delta t) = \exp(-(x - c\Delta t - 10)^2)$ as initial condition.
   (c) Check the convergence of the numerical method by varying the resolution and performing a self-convergence test. Plot the convergence order over time by assuming at least 3 different resolutions.
   (d) Write a short documentation about your results interpreting the most important aspects.

Please hand in the answers in the next lecture or send the answers as pdf to kwtsang@nikhef.nl. We also expect you to send us the source code in addition with the compiler options you have used. For the task it is allowed to work in groups of 2-3 people.