Projekt Praktikum: Winter semester 2022/2023

Submission

Deadline for submission to the supervisors is the last week of January 2023. A digital version (PDF) should be submitted via email to the supervisors and the praktikum office (`physik.f-praktikum@uni-jena.de`). The results are presented in the form of a scientific poster the second week of February 2023.

Project: Solution of the initial boundary value problem with the wave equation

The wave equation in 1+1 dimensions for the scalar field $\phi(t,x)$

$$\partial_{tt}\phi - c^2\partial_{xx}\phi = 0\ , \tag{0.1}$$

is the main example for hyperbolic PDEs and it is usually associated to initial or initial-boundary value problems (IVP, IVBP). In the former, one specifies an initial profile $\phi(t=0,\,x)=\phi_0(x)$. In the latter, one additionally specifies conditions on the boundary $\partial\Omega$ of the spatial domain $\Omega=[a,\,b]$ (at all times). For example, periodic boundary conditions are $\phi(t,\,a)=\phi(t,\,b)$. IVPs and IBVPs with the wave equation are *well posed* problems, i.e. a unique solution exists and depends continuously on the initial data.

1. **Analytical part.** Write down the general analytical solution of the IVP: show that $\phi(x,t)$ can be decomposed in two "elementary waves"(or *characteristic waves*) propagating in opposite directions with speed c.

2. Rewrite the wave equation from the second order form above to a first-order in time and second-order in space system by defining the new variable $\Pi=\partial_t\phi$,

$$\partial_t\phi = \Pi\ , \tag{0.2a}$$
$$\partial_t\Pi = c^2\partial_{xx}\phi\ . \tag{0.2b}$$

   Then, derive d'Alembert's formula: find a general solution for $\phi(t,x)$ expressed in terms of the initial conditions $\phi(t=0,x)=\theta(x)$ and $\Pi(t=0,x)=\psi(x)$. Discuss the effect of two possible choices, (e.g. $\Pi(t=0,x)=0$ and $\Pi(t=0,x)=\pm c\theta'(x)$), for the initial conditions for $\Pi(t=0,\,x)$ in view of the analytical solution derived above.

3. Write the wave equation as a fully first order system by defining the new variables $\Pi=\partial_t\phi$ and $\chi=\partial_x\phi$. Rewrite the system as:

$$\partial_t u + A\partial_x u = S\ , \tag{0.3}$$

   where A is a matrix and $u=\{\phi,\,\Pi,\,\chi\}$ is the collection of the three fields or "state vector" and S is a source term. Discuss choices for the initial data for the fields $u(t=0,\,x)$.

   Then, observe that A is a block matrix. Focus on its non-degenerate 2×2 sub-matrix A' defined via

$$\partial_t\begin{pmatrix}\Pi\\ \chi\end{pmatrix} + A'\begin{pmatrix}\partial_x\Pi\\ \partial_x\chi\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix}\ . \tag{0.4}$$

   Identify the matrix A', then compute its eigenvalues λ_i and its eigenvectors. What happens to the system of equations (0.4) if you switch to new variables

$$\begin{pmatrix}\tilde\Pi\\ \tilde\chi\end{pmatrix} := R^{-1}\begin{pmatrix}\Pi\\ \chi\end{pmatrix} \tag{0.5}$$

   where $R=(v_1,\,v_2)$ is the matrix of eigenvectors $v_1,\,v_2$? To answer this, first show that $A'R=R\Lambda$, where $\Lambda=\mathrm{diag}\lambda_i$.

4. **Numerical part: Finite differencing approximation.** Introduce a uniform spatial grid, $x_i = ih$ with $i = 0, ..., n-1$ and $h = 1/(n-1)$ that discretizes the domain $\Omega = [0, 1]$. Call the values of a field $u(x)$ on the grid points $u(x_i) = u_i$. Show that an approximation of the first derivative at point x_i is given by

$$\frac{du}{dx} \approx \frac{u_{i+1} - u_{i-1}}{2h} + \mathcal{O}(h^2) \ . \tag{0.6}$$

Calculate the error term $\mathcal{O}(h^2)$ of the formula above.

Hint: consider the Taylor expansions of the function at $x_i \pm h$.

The formula above is called a second-order (because of the error term) finite-differencing approximation of the derivative with centered stencil (Note you use the same number of points around x_i).

Similarly, show that an approximation of the second derivative at point x_i is given by

$$\frac{d^2u}{dx^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \mathcal{O}(h^2) \ , \tag{0.7}$$

and compute the error term.

5. Consider example functions, e.g.

$$f(x) = (x - \frac{1}{2})^2 + x \ , \tag{0.8}$$

$$f(x) = (x - \frac{1}{2})^3 + (x - \frac{1}{2})^2 + x \ , \tag{0.9}$$

$$f(x) = \sqrt{x} \ , \tag{0.10}$$

$$s(x) = \sin\left(12\pi x\right) \ , \tag{0.11}$$

$$s(x) = \sin^4\left(12\pi x\right) \ , \tag{0.12}$$

$$g_a(x) = \exp\left(-ax^2\right) \ , \tag{0.13}$$

and calculate the first and second derivative with the finite differencing formulas above. For each, plot the exact derivative and the finite difference approximation (top panel) and their differences (bottom panel) for a grid of say $n = 20$ points.

In some relevant cases, study the error term and the **convergence** of the numerical derivative. Proceed as follows. First, consider the difference

$$f'(x) - f'^{(h)}(x) = Ch^2 + \mathcal{O}(h^3) \ , \tag{0.14}$$

where $f'^{(h)}(x)$ denotes the numerical derivative of function f computed with a grid resolution h, and $f'(x)$ the analytical derivative. Then, compute the same quantity for a second grid resolution, $f'(x) - f'^{(h/2)}(x) = C(h/2)^2$. Note that the constant C is the same (why?). Finally, plot

$$\frac{f'(x) - f'^{(h)}(x)}{f'(x) - f'^{(h/2)}(x)} = \frac{Ch^2 + \mathcal{O}(h^3)}{C(h/2)^2 + \mathcal{O}((h/2)^3)} \sim 2^p \tag{0.15}$$

and extract the convergence order p, equal to 2 in this example. Note, that it will be a function of x, *i.e.*, the convergence depends on the grid position.

The convergence order for the entire spatial domain can be evaluated taking the (euclidean) norm as

$$\frac{|f'(x) - f'^{(h)}(x)|}{|f'(x) - f'^{(h/2)}(x)|} = Q \tag{0.16}$$

where $Q = 2^p$, and p being the convergence order.

In the case of oscillatory functions, it is useful to consider a different (but equivalent in meaning) plot in which one shows the difference $f'(x) - f'^{(h)}(x)$ superposed to the difference $Q(f(x) - f'^{(h/2)}(x))$ scaled by the proper expected factor $Q = 2^2 = 4$.

Note that in absence of a known analytical solution, one can always consider a third (or more) grid resolution, say $h/4$, and compare pairs of differences $(f'^{(h)}(x) - f'^{(h/2)}(x))$, $Q(f'^{(h/2)}(x) - f'^{(h/4)}(x))$ properly scaled. This second test is called *self-convergence test.* Perform this test for a couple of relevant cases and compare the results obtained with the two different convergence tests.

Bonus: Consider a two dimensional spatial grid $x_i = ih_x$, $y_j = jh_y$ with $i \in [0, n_x - 1]$, $j \in [0, n_y - 1]$ and $h_x = 1/(n_x - 1)$, $h_y = 1/(n_y - 1)$. Show that the second order mixed derivative takes the form:

$$\left(\frac{\partial^2 u}{\partial x \partial y}\right) = \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4h_x h_y} + \mathcal{O}(h_x^2, h_y^2) \tag{0.17}$$

Discuss the second-order derivatives ("pure" and mixed) in the three dimensional case. Consider the function $f(x,\, y,\, z) = 2xy + y^2 z$ and compute all its analytical derivatives. Then compute numerically $\partial_x \partial_y\, f(x,\, y,\, z)$ and $\partial_y \partial_z\, f(x,\, y,\, z)$ and compare them with the analytical case as in the previous point.

6. **Runge-Kutta time-integration.** There exist various algorithms to perform the time integration of (a single or a system of) ODEs

$$\frac{du}{dt} = F(u)\ , \tag{0.18}$$

given an initial value $u(t = 0) = u_0$ and a r.h.s. function F.

Runge-Kutta (RK) methods are robust schemes for integrating IVP. A general formulation can be given starting from the integral form of the IVP above and approximating the integrals by quadratures. The **S-stage RK** scheme reads:

$$u_{n+1} = u_n + \Delta t \sum_{i=0}^{S} b_i F(t_n + c_i \Delta t,\, u^{(i)}) \tag{0.19a}$$

$$u^{(i)} = \Delta t \sum_{j=0}^{S} a_{ij} F(t_n + c_j \Delta t,\, u^{(j)})\ , \tag{0.19b}$$

An equivalent form is

$$u_{n+1} = u_n + \Delta t \sum_{i=0}^{S} b_i K_i \tag{0.20a}$$

$$K_i = F\left(t_n + c_i \Delta t,\, u_n + \Delta t \sum_{j=1}^{S-1} a_{ij} K_j\right)\ . \tag{0.20b}$$

Observations: (i) a S-stage RK is defined by the set of coefficients $(a_{ij},\, b_i,\, c_i)$. These coefficients can be organized in a **Butcher table**

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & ... & a_{1S} \\ c_2 & a_{21} & a_{22} & ... & a_{2S} \\ ... & ... & ... & ... & ... \\ c_S & a_{S1} & a_{S2} & ... & a_{SS} \\ \hline & b_1 & b_2 & ... & b_S \end{array} \tag{0.21}$$

(ii) $a_{ij} = 0$ with $j \geq 1$, the method is *explicit*; (iii) if a_{ij} has nonzero elements in the upper triangular part, the method is *implicit*; (iv) An S-stage RK is an S$^{\text{th}}$ order accurate method, $\mathcal{O}(\Delta t^S)$.

Write a routine implementing the RK4 ($S = 4$) scheme

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \tag{0.22}$$

and test it using the *Hamilton equations for the harmonic oscillator* $H(q,p) = 1/2(p^2/m + q^2 m)$ (where $q = \omega x$). Note that for this specific set of equations the convergence is not $\mathcal{O}(\Delta t^4)$ but higher due to cancellations in the error term. Use $u = \{q, p\}$ and test convergence of the solution, say $q(t)$, against the exact solution and conservation of energy over long timescales (several periods).

Hint Do not store all the time steps in memory! For N timesteps that would require $8N$ doubles (or, in general, $4 \times m \times N$ with 4 stages and m variables). Memory usage should be kept constant ($= 4m$) by overwriting memory when updating the state vector, and by writing to file every given number of iterations.

Hint Try not to use too complex programming for these tasks. Sophisticated codes with classes, inheritance, etc., are difficult to navigate through and debug and are most useful in large projects.

Bonus In order to numerically integrate an ODE using RK4, 5 registers of memory are required to implement the algorithm as described above. This might lead to memory issues in case of big simulations with many equations to solve at the same time or when many variables (and consequently grid points) are involved. For this reason low-storages algorithm have been developed allowing to significantly decrease the memory needed. Discuss the feasibility of the method RK4()4[2S] by Ketcheson, in which only two storages are used to do the RK4 time-step and implement it to solve the Harmonic oscillator problem.

7. **Numerical solution to the Wave equation with periodic boundary condition** Consider now the wave equation in first-order in time form, say 0.2 (but the exercise can be repeated with 0.3). Discretize the space with a uniform grid and represent the derivatives on the r.h.s. at each point x_i with the finite differencing expressions derived above. Schematically, the r.h.s. discretization at each point can be indicated as $L[\{u_j\}]$ and collected in the array r_i. The wave equation 0.2 is then discretized in its *semi-discrete* form

$$\frac{du_i}{dt} = r_i = L[\{u_j\}] \ , \tag{0.23}$$

that can be integrated in time using e.g. a Runge-Kutta scheme. This approach is known as *method of lines*. Note that the spatial discretization can be done also with techniques other than finite-differencing.

Implement this method for 0.2. Use $c^2 = 1$ and a domain $\Omega = [0, 1]$ with **periodic boundary conditions**. Use as initial data any of the functions $s(x)$, $g(x)$ tested above.

Hint 1: The computation of finite differences always require the presence of grid points on the left and on the right. A general strategy to handle this situation is to add the necessary number of points ("ghosts") per direction (and dimension). These points are unphysical and should be filled before computing the finite difference by copy or extrapolating the physical values. In case of second-order finite differencing stencils and periodic boundary conditions, one has 2 ghost points and can simply fill them by copying the appropriate physical values.

Hint 2: Estimate how much memory do you need for a grid of $n+2$ points, m variables and 4 RK stage. In your code, input the grid size n, the time-step (α, see below how), and the final time. Output the spatial field at given time-steps or iterations as a 2 columns text file $\{x,\ \phi(x)\}$, with name labeled by time. Experiment with different ways to visualize the 1D wave. Keep in mind that the result must be printed *i.e.,* making a movie is a great visualization but is impossible to present on a paper. Try to find a combination of 1D or/and 2D or/and 3D plots, that are compact, efficient, simple to read and deliver the important information.

8. **Stability and convergence.**

A key aspect of the simulation is the choice of the timestep. For a given grid spacing h, set the timestep according to the following equation:

$$\Delta t = \frac{\alpha}{|c|} h \ , \tag{0.24}$$

and experiment with $\alpha = 1/2, 1, 2$. Verify experimentally that a necessary condition for stability is $\alpha \leq 1$, known as Courant-Friedrich-Lewy (CFL) condition. The solution of the hyperbolic IVP at a given point depends on the information in its *domain of dependency*; numerical stability is guaranteed if the time-step is sufficiently small such that it contains the domain of dependency for that point. In other terms, a scheme is CFL stable if the numerical domain of dependency is larger than that of the PDE. In general, the CFL condition depends on the equations and on the particular integration scheme.

Test your code by varying the grid spacing h, $h/2$, $h/4$ and performing both convergence tests and self-convergence test. A simple way to perform a convergence test is to compare the solution at successive full periods $T = (b-a)/c = 1$ with the initial data $\{\phi(0,x), \phi(T,x), \phi(2T,x), \dots\}$.

Bonus **Von-Neumann stability**. Investigate and attempt to implement the von-Neumann stability analysis.

9. **Open boundary conditions**. Implement "transparent" boundary conditions at the computational domain boundary instead of periodic ones. Do this with 3 different methods:

(i) Simply fill the ghosts zones by linearly extrapolating the field from inside to the ghost zone. This is similar to what is done for period boundaries, but note here the ghosts are not filled with the exact information.

(ii) In addition to extrapolating linearly into the ghosts, let us try to impose an advection equation of the type

$$\partial_t \phi \pm \partial_x \phi = 0 \ , \tag{0.25}$$

at the last physical point. The idea here is to impose that the solution at the physical boundary (last physical points) "translates out" on the left and on the right. [Note that solution of the advection equation is a translation].

(iii) A third method is based on the characteristics analysis done at the beginning. From there one can argue that the characteristics encodes the "flow of information" and thus require that the incoming (outgoing) characteristics are zero

$$0 = w_{\pm} = \partial_x \phi + \Pi \ , \tag{0.26}$$

i.e. no waves enter the domain from the left/right. Using the fact that $\Pi = \partial_t \phi$ and $\partial_t \Pi = \partial_{xx}\phi$, it is possible to use second order and centered finite difference stencils to find an expression for ϕ^k_{n+1} and ϕ^{k+1}_n, where k and n are respectively discrete indices for time and space, in terms of ϕ^k_n, ϕ^k_{n-1}, ϕ^{k-1}_n, α (the latter being the C-L factor $\alpha \equiv c\Delta t/\Delta x$). One can thus fill the ghosts of each side using the expression found for ϕ^k_{n+1}.

Study the convergence for the three cases and discuss what happens.

10. **Regge-Wheeler (RW) equation**

Black holes respond to perturbations by resonating at characteristic complex frequencies determined by the hole's mass and spin. Similarly to the normal modes of a string, the imaginary part of these frequencies describes a proper oscillation frequency of the black hole spacetime. The black hole modes however are *quasi* normal modes, as the IVP has open boundaries and the oscillations are damped (dissipated) by the emission of gravitational waves.

Side note. Using the tortoise radial coordinate $r_* \in (-\infty, \infty)$ the RW equation reads

$$\partial_{tt}\phi - c^2 \partial_{r_* r_*}\phi + V(r)\phi = 0 \tag{0.27}$$

where V is a potential depending on r. It is possible to adopt a Regge-Wheeler-Zerilli potential, that is defined in schwarzschild coordinates. Thus, it would also be required to implement the root-finder to go from tortoise coordinates to schwarzschild ones. The RWZ potential has a maximum at $r_* \sim 3M$ and rapidly drops to zero towards the black hole horizon ($r_* \to -\infty$) and also falls to zero at ($r_* \to +\infty$). Hence at these boundaries the RW equation reduces to the 1+1 wave equation in Cartesian coordinates studied below. Note the IVP with the Regge-Wheeler equation has similarities with the IVP with the Schrödinger equation in quantum mechanics.

Effective approach. A simpler approach is to adopt a Pöschl–Teller potential, that allows to obtain qualitatively the same behavior of the wave. The potential can be set as

$$V(x) = \frac{V_0}{\cosh^2(\kappa x + \beta)} \tag{0.28}$$

with the particular choice of parameters: $V_0 = 0.15$, $\kappa = 0.18$, $\beta = -0.43$.
Replacing the r_* in Eq. 0.27 with x, and defining the potential as in Eq. 0.28 we obtain a full system.

Adapt the 1+1 wave equation code with open boundaries for the solution of the RW equation. Study how a Gaussian pulse is scatterred by the PT potential.

Hint 1 First, investigate the behaviour of the potential given. After implementing it into your wave equation, experiment with different initial conditions for the wave, its location and amplitude.

11. **2+1 Wave equation** *This is a quite time-consuming task, that if we have time we will try to work out. The result is interesting and beautiful. But a proper understanding of RW with PT potential task is more important and should come first.*

Bonus Estimate π as (i) twice the area under a semicircle and (ii) as $4\int_0^1 \frac{1}{(1-x^2)}dx$. Then, repeat the process using OpenMp to parallelize the computations.