Homework Assignment 2
SF2521, Spring 2016

Topics: Duhamel's principle and stability of numerical schemes; Conservation form; Relation between non-linear and linearized problems; The Shallow-water equations; The Lax-Friedrichs Scheme; Solution by characteristics; Boundary conditions.
Purpose: (1) Understand how inhomogeneous term in a numerical scheme affect teh solutions. (2) Get acquainted with elementary properties of and solution schemes for initial-boundary value problems for hyperbolic systems
Instructions: Write a short report with the plots and answers to the questions posed. Make sure the plots are annotated and there is explanation for what they illustrate.

1 Stability of numerical schemes (1p)

Consider a general scheme

$$\begin{aligned} U^{n+1} &= Q(t_n)U^n + \Delta t F^n \\ U^0 &= g \end{aligned}$$

where $U^n \in \mathbb{R}^d$.

1. Show that the following discrete Duhamel's Principle holds:

   $$U^n = S_h(t_n, 0)g + \Delta t \sum_{\nu=0}^{n-1} S_h(t_n, t_{\nu+1})F^\nu, \tag{1}$$

   where $t_n = n\Delta t$, and

   $$\begin{aligned} S_h(t,t) &= I, \;\; t \in \mathbb{R} \\ S_h(t_{n+1}, t_\mu) &= Q(t_n)S_h(t_n, t_\mu). \end{aligned}$$

   Explain (1) following the following statements: *(i) the inhomogeneous term F can be regarded as additional initial conditions for each new time level; and (ii) the solution is the sum of all the solutions that satisfy each of the "initial conditions".*

2. Show that if

   $$||S_h(t_{\nu+1}, t_\nu)||_h \leq Ke^{\alpha h},$$

   for some constant α and K independent of h, then

   $$||U^n||_h \leq K\left(e^{\alpha t_n}||g||_h + \int_0^{t_n} e^{\alpha(t_n - s)}ds \max_{0\leq\nu\leq n-1} ||F^\nu||_h\right). \tag{2}$$

   Here, $||\cdot||_h$ denotes either a norm for the grid functions U^n, g, and F^ν, or the associated operator norm for the solution operator $S_h(t,t)$. This means that *stability of the homogeneous problem* ($F \equiv 0$) *implies stability for the inhomogeneous problem.*

3. What would change if $\alpha = h^{-1/2}$ in (b)? Hint: in (2), the exponential funtions do not depend on the grid — no matter how one decreases h, the exponents remain the same.

2 The Shallow Water Model

In this exercise we shall investigate the relation between a non-linear problem and the corresponding linearized version. In particular we will see how well linear analysis predicts the behavior of the nonlinear problem.

Shallow water flow over a horizontal bottom is modeled by

$$\begin{cases} h_t + (hv)_x = 0, & \text{(conservation of volume)} \\ (hv)_t + (hv^2 + \frac{1}{2}gh^2)_x = 0, & \text{(force belance in } x\text{)} \\ \text{on } (x,t) \in [0,L] \times [0,\infty), & \end{cases} \tag{3}$$

where h is the water height (depth) and v the velocity of the water. In this model, the water velocity does not vary vertically. You may check Leveque's book for related discussion. We prescribe the boundary conditions $v(0) = v(L) = 0$ which says that water stay still at the boundaries, and the initial conditions that corresponds to a localized "water hill",

$$\begin{cases} h(x,0) = H + \varepsilon e^{-(x-L/2)^2/w^2} \\ v(x,0) = 0 \end{cases} \tag{4}$$

We shall take $L = 10m$, $H = 1m$, and $g = 9.61(m/s^2)$. The "width" w of the water hill is $0.4m$ and its height ε will be varied.

2.1 Numerical Solution (1 p)

To begin with, let $\varepsilon = 0.1$ and solve numerically the conservation form equations, for h and $\rho = hv$, using the Lax-Friedrichs method: $u = (h, hv)^T$,

$$u_j^{n+1} - u_j^n + \frac{\Delta t}{\Delta x}\left(F^{LxF}(u_{j+1}, u_j) - F^{LxF}(u_j, u_{j-1})\right) = 0,$$

where the Lax-Friedrichs flux is defined as

$$F^{LxF}(u_{j+1}, u_j) := \frac{1}{2}\left[f(u_{j+1}) + f(u_j) - \alpha(u_{j+1} - u_j)\right], \quad \alpha = \max_u |f'(u)|,$$

for $j = 0, \pm 1, \pm 2, \cdots$. Use ghost cells at the boundaries. Prescribe values there by the procedure described in Leveque Ch 7 for solid walls. In this problem, $\max_u |f'(u)|$ may be taken to be the eigenvalue of f' with the largest magnitude, but you may use $\alpha = \frac{\Delta x}{\Delta t}$ for (1) and (2) below. Choose $\Delta t \leq C\Delta x$ and experiment with different constants C. Use $\Delta t/\Delta x$ as large as possible, without violating stability.

1. Make plots showing wave propagation and reflections at the boundaries. Compute at least until waves have been reflected at both boundaries and crossed each other, say until $t = 3$.

2. Run the program again with larger values of $\varepsilon(= 0.4, 0.8, 1.2, \dots)$. Describe how the solutions change with respect to

   (a) Wave shape, amplitude

   (b) Wave speed

   (c) Wave collisions?

   (d) You may have to adjust the time-step to ensure stability. Why?

3. Experiment with different values of α, say $\alpha = C_0 \frac{\Delta x}{\Delta t}$, $C_0 \in [0.5, 1.5]$. Compare the solutions that you computed with different values of α.

2.2 Linearization

1. (0.5p) Choose a constant state (h_0, v_0) which is consistent with the prescribed initial and boundary conditions, and derive the linearized problem at that state. Show that the linear problem is hyperbolic and compute wave speeds.

2. (1p) The linear constant coefficient problem that you derived above can be solved analytically by diagonalizing the system using eigenvectors. Determine the solution of the linear problem at $t = 1$. Discuss how information propagates, if the boundary conditions cause reflections and when reflected waves will appear. Compare with the numerical results for the non-linear case that you computed.

2.3 Non-reflecting Boundary Conditions (0.5p)

Derive boundary conditions that do not cause reflections for the linear problem. Formulate the corresponding conditions for the non–linear case. Implement the conditions in your program using either the technique of characteristic variables or by simply extrapolating all variables at the boundary (as described in Leveque Ch 7). How well does the method work? Try to measure the size of the reflection.

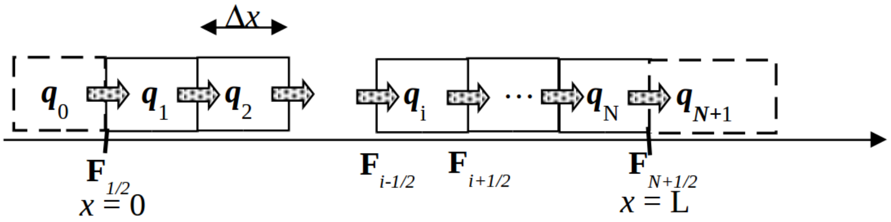

Figure 1: A diagram of the numerical setup

3 Suggestions for program structure

The homework assignments in this course require coding of several finite volume schemes for initial-boundary value problems for a number of different models, mostly explicit. You can of course code as you find practical; we have found the following structure useful. It is possible to separate the scheme from the equation system by defining the flux function $\boldsymbol{f}$ as the programming interface between scheme and equation. The state variables $\boldsymbol{q}$ for a system of s equations can be stored in an $N \times s$ array, say Q(1:N,1:s) .

- Fill ghost cells 0 and $N+1$ by the boundary conditions; and augment Q by rows for the ghost cells 0 and $N+1$.
- Compute the numerical fluxes $\boldsymbol{F}_{i+1/2}, i = 0, 1, \ldots, N$. This may entail much more computation than evaluating the flux function f which defines the differential equation, but for the Lax-Friedrichs scheme it is not much more.
- Compute the flux differences.
- Update the cells $1, \ldots, N$.
- Repeat the previous steps.

Two more suggestions:

- For debugging, put plotting into the code so you can inspect the solution and the fluxes and other ingredients at each time step; turn off the plotting and printing when the code works.
- Arrange – by saving solutions on file and implementing interpolation functions, if need be – so that solutions from different grids and with different parameter settings can be compared point-wise (in time and space), e.g. for showing convergence in L^2 – norm and showing several solutions in one plot.