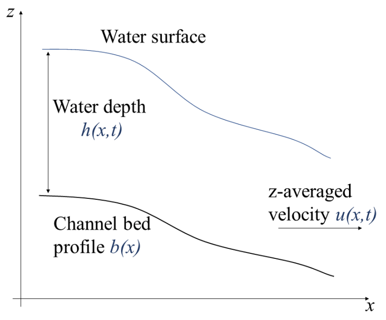
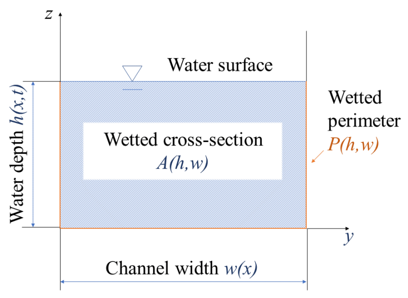
Home Assignment : Numerical 1D resolution of a flood wave propagation through an open-channel

2018



(a) Profile of the channel



(b) Channel cross-section

Figure 1: Schema of an open channel

Introduction

Floods on rivers can be caused many factors including heavy rainfall, accelerated snow melting, failure of dams and other. It is of crucial importance to predict the effects of these factors on the probability and extent of a flood, in order to ensure safety measures are applied when needed. At the same time, modelling of the performance of different flood prevention systems makes their design process more efficient. These tasks can with good accuracy be approached using the Saint-Venant equations.

The Saint-Venant equations were developed to model one-dimensional flows in channels with various altitude profiles and cross-sections. In this project we will use their simplified version, applicable for rectangular cross-section of the channel. Schema of such channel is provided in figure 1.

The system of 1D Saint-Venant equations consists of one linear transport equation (continuity) and one non-linear transport equation (streamwise momentum):

$$\begin{gathered}\partial_t h + u\partial_x h + h\partial_x u = 0 \\ \partial_t u + u\partial_x u = -g\left(\partial_x h + \partial_x b + \frac{u^2 n^2}{R^{4/3}}\right)\end{gathered} \tag{1}$$

where

$$R(h,w) = \frac{A(h,w)}{P(h,w)} = \frac{hw}{w+2h} \tag{2}$$

is the hydraulic radius of the channel. The idea of the proposed work is, step by step, to solve these equations.

This project is divided into three parts: the two first parts can be treated separately, while for the third part, one will need the codes produced in the first two parts.

1 Water depth transport (continuity equation)

At first we will focus on the equation modelling the transport of the water depth h by the velocity u:

$$\partial_t h + u\partial_x h + h\partial_x u = 0 \tag{3}$$

1.1 Discretization & stability assessment

Let us assume that u is a positive constant ($u = \mathit{const.} > 0$), thus simplifying equation (3) to:

$$\partial_t h + u\partial_x h = 0 \tag{4}$$

a) Discretize equation (4) with finite difference method using the explicit Euler scheme for temporal derivative and the second-order centered scheme for spatial derivative. Assume a uniform spatial distribution of computational nodes ($\Delta x = \mathit{const.}$). Show that the value of h at position i and time step $n+1$ can be approximated as

$$h_i^{n+1} \approx h_i^n - \frac{u\Delta t}{2\Delta x}(h_{i+1}^n - h_{i-1}^n) \tag{5}$$

Prove that the centered scheme for first spatial derivative has accuracy of $O((\Delta x)^2)$.

[2 point]

b) Assess the stability of the discretization (5) using von Neumann method. Is there a condition on $u, \Delta t$ and Δx that must be satisfied to avoid error amplification? (One can use the results of the exercise, or the course).

[2 points]

c) Prove that Lax scheme:

$$\frac{h_i^{n+1} - \frac{1}{2}(h_{i-1}^n + h_{i+1}^n)}{\Delta t} \approx -u\frac{(h_{i+1}^n - h_{i-1}^n)}{2\Delta x} \tag{6}$$

is conditionally stable and provide the condition of stability.

[2 points]

d) Using Taylor expansion, show that:

$$h_i^n = \frac{1}{2}(h_{i-1}^n + h_{i+1}^n) - \frac{(\Delta x)^2}{2}\frac{\partial^2 h}{\partial x^2} + O((\Delta x)^4) \tag{7}$$

[2 points]

e) Name at least one unconditionally stable scheme.
[1 point]

f) Another technique to ensure stability is to add a small diffusion term to the equation that one wants to solve:

$$\partial_t h \approx -u\partial_x h + \epsilon_h \partial_x^2 h, \tag{8}$$

where the artificial difusivity ϵ_h needs to satisfy the condition

$$\Delta x \to 0 \implies \epsilon_h \to 0 \tag{9}$$

in order to maintain consistency.

Show that Lax scheme also adds diffusion to the original equation, corresponding to $\epsilon_h = \frac{(\Delta x)^2}{2\Delta t}$.

Hint: Consider the semi-discrete equation

$$\frac{h_i^{n+1} - \frac{1}{2}(h_{i-1}^n + h_{i+1}^n)}{\Delta t} = -u\partial_x h \tag{10}$$

and the result of example 1.1 d).

[2 points]

1.2 Staggered grid arrangement

Adding an artificial diffusion term to equation (3) leads to:

$$\partial_t h + u\partial_x h + h\partial_x u - \epsilon_h \partial_{x^2} h = 0 \tag{11}$$

From this point onward we will assume velocity is stored in a so called ”staggered grid” as follows:

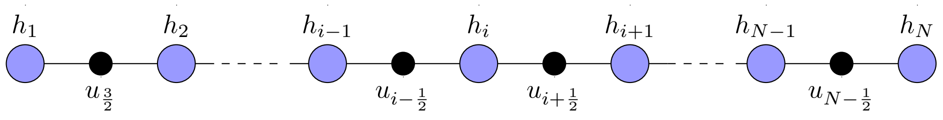


Figure 2: Scheme of the staggered grid

a) Show, using Taylor expansion, that the second order centered difference approximations of $\partial_x u$, $\partial_x h$ and $\partial_x^2 h$ in equation 11 are given by:

$$\begin{aligned} \partial_x u &= \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} + O\left((\Delta x/2)^2\right) \\ \partial_x h &= \frac{h_{i+1} - h_{i-1}}{2\Delta x} + O\left((\Delta x)^2\right) \\ \partial_x^2 h &= \frac{h_{i+1} - 2h_i + h_{i-1}}{(\Delta x)^2} + O\left((\Delta x)^2\right) \end{aligned} \tag{12}$$

[2 points]

b) Show that velocity at position x_i can be approximated by interpolation as:

$$u_i = \frac{u_{i+1/2} + u_{i-1/2}}{2} + O\left((\Delta x/2)^2\right) \qquad (13)$$

[1 point]

1.3 Numerical implementation

a) Using the results of section 1.2, discretize equation (11) applying explicit Euler scheme to temporal derivative and second order centered schemes for spatial derivatives. Assume that the values of u are known and present the result in the form:

$$h_i^{n+1} = a\ h_{i-1}^n + b\ h_i^n + c\ h_{i+1}^n \qquad (14)$$

where a, b and c have to be explicited in function of $u^n_{i+\frac{1}{2}}$, $u^n_{i-\frac{1}{2}}$, Δt, Δx and ϵ_h.

Writing equation (14) in matrix and vector form:

$$\mathbf{h}^{n+1} = \mathbf{M}_h \mathbf{h}^n\ , \qquad (15)$$

describe the generic structure of the matrix $\mathbf{M}_h$.

[2 points]

b) Note that solving the equation (14) for h_1^{n+1} and h_N^{n+1} requires the knowledge of $u_{1/2}$ and $u_{N+1/2}$ respectively, while these values are not stored in the vector $\mathbf{u}$ (see figure 2). We therefore have to use the boundary conditions of velocity. Here we take homogeneous Neumann condition at both boundaries:

$$\frac{u_{\frac{3}{2}} - u_{\frac{1}{2}}}{\Delta x} = 0 \qquad \frac{u_{N+\frac{1}{2}} - u_{N-\frac{1}{2}}}{\Delta x} = 0 \qquad (16)$$

How would you prescribe the values $u_{1/2}$ and $u_{N+1/2}$ to satisfy (16)?

[1 point]

c) Write a MATLAB function to build the matrix $\mathbf{M}_h$ from given $\Delta x, \Delta t, \mathbf{u}$, now taking $\epsilon_h = \frac{u_i \Delta x}{2}$.Use the skeleton function `continuity_build_matrix.m` .

Note: The vector $\mathbf{u}$ is one element shorter than the vector $\mathbf{h}$ (see figure 2). Use the boundary conditions (16) if needed.

[2 points]

Bonus: Show that $\epsilon_h = u_i \frac{\Delta x}{2}$ corresponds to applying backward uncentered scheme to $\partial_x h$ in equation (3). [1 point]

d) Implement the following boundary conditions for h:

- *Dirichlet boundary condition at* $x = x_{min}$ - Having a Dirichlet boundary condition there means that

$$h_1^{n+1} = h_1^n = h_{Dir} \tag{17}$$

  Modify the first line of the matrix $\mathbf{M}_h$ accordingly, using the skeleton function `continuity_apply_bc.m` .

  [1 point]

- *homogeneous Neumann Boundary condition at* $x = x_{max}$ - As in example 1.2b), one can consider a so called "ghost point" of index $N+1$ to create a homogeneous Neumann Boundary condition:

$$\frac{h_{N+1}^n - h_N^n}{\Delta x} = 0 \tag{18}$$

  Injecting this relation in equation (14), show that the boundary condition that one has to implement is

$$h_N^{n+1} = a^\star h_{N-1}^n + b^\star h_N^n$$

  where $a^\star, b^\star$ have to be explicited in terms of $u_{i+\frac{1}{2}}$, $u_{i-\frac{1}{2}}$, Δx, Δt, ϵ_h.

  Modify the last line of the matrix $\mathbf{M}_h$ accordingly, using the skeleton function `continuity_apply_bc.m` . Use the same ϵ_h as in 1.2 c) if needed.

  [1 points]

e) Describe what happens in the script `water_depth_transport.m`. To verify your implementation, compare the numerical and analytical solution. Comment on any disagreements you observe.

[2 points]

2 Momentum transport

2.1 Theoretical prerequisities

In the Saint-Venant model, the velocity satisfies the equation:

$$\partial_t u + u\partial_x u = -g\left(\partial_x h + \partial_x b + S_f(u^2, h)\right) \tag{19}$$

$$S_f = u^2 \frac{n^2}{R^{\frac{4}{3}}} \tag{20}$$

where g is the gravitational acceleration, $b(x)$ is the channel bed profile, $R(h) = \frac{wh}{w+2h}$ is the hydraulic radius, while $n = \textit{const.} > 0$ is the Manning's friction coefficient. Adding an artificial diffusivity to ensure numerical stability leads to:

$$\partial_t u + u\partial_x u - \epsilon\partial_x^2 u = -g\left(\partial_x h + \partial_x b + S_f(u^2, h)\right) \tag{21}$$

Discretization of (21) with Euler implicit scheme, assuming the staggered grid arrangement (Fig. 2), yields:

$$\begin{aligned}\frac{U_{i+\frac{1}{2}} - u^n_{i+\frac{1}{2}}}{\Delta t} + U_{i+\frac{1}{2}}\frac{U_{i+\frac{3}{2}} - U_{i-\frac{1}{2}}}{2\Delta x} - \epsilon^{n+1}_{i+\frac{1}{2}}\frac{U_{i+\frac{3}{2}} - 2U_{i+\frac{1}{2}} + U_{i-\frac{1}{2}}}{(\Delta x)^2} \\ = -g\left(\frac{H_{i+1} - H_i}{\Delta x} + \frac{b_{i+1} - b_i}{\Delta x} + (S_f)^{n+1}_{i+\frac{1}{2}}\right)\end{aligned} \tag{22}$$

where $u^{n+1} \equiv U,\ h^{n+1} \equiv H$ for readability.

To solve this non-linear system, several approaches exist. Among them are the Newton method, the fixed-point algorithm etc. Here, the fixed-point algorithm will be used.

2.2 Fixed-point algorithm

Principle: The method is based on finding a *fixed point* of a continuous function F, defined as

$$U = F(U)\ . \tag{23}$$

One therefore has to reformulate the non-linear problem to identify such function F. Then, starting from an initial guess U^0, one iterates $U^{k+1} = F(U^k)$ until one reaches convergence. Usually a convergence criterion that one can take is

$$\delta = \parallel U^k - U^{k-1} \parallel < tolerance \tag{24}$$

Application to eq. (22):

- Identify a function F from equation (22) such that

  $$U^{k+1}_{i+\frac{1}{2}} = F(U^k_{i-\frac{1}{2}},\ U^k_{i+\frac{1}{2}},\ U^k_{i+\frac{3}{2}})\ , \tag{25}$$

  There are several ways to obtain the required form (25), depending on which $U_{i+\frac{1}{2}}$ in equation (22) one chooses to isolate.

  Hint: Refer to Exercise 7 of the course.

  [2 points]

Boundary conditions: Recall that the velocity is only stored at the interior nodes $x_{3/2} \leq x \leq x_{N-1/2}$, while the values at the "ghost nodes" $x_{1/2}, x_{N+1/2}$ are defined by boundary conditions. In this section we want generic Neumann conditions to be satisfied at both boundaries:

$$\frac{U_{\frac{3}{2}} - U_{\frac{1}{2}}}{\Delta x} = \alpha_N \qquad \frac{U_{N+\frac{1}{2}} - U_{N-\frac{1}{2}}}{\Delta x} = \beta_N \tag{26}$$

One of the ways of enforcing these boundary conditions is to prescribe the values of $u_{1/2}, u_{N+1/2}$ at the beginning of every iteration.

Relaxation: In practice, one needs to add a so-called relaxation to have convergence of this method. This is done by doing the following steps :

$$
\begin{aligned}
&U^* = F(U^k) && \text{Compute an intermediate velocity} \\
&U^{k+1} = \theta U^* + (1-\theta)U^k && \text{Compute the new velocity}
\end{aligned}
$$

where the relaxation factor θ is of the order of percent. We will use $\theta = 0.1$.

2.3 Numerical implementation

a) Implement the function `solve_momentum_transport.m` in which one applies the fixed-point algorithm to solve (22), including relaxation and enforcement of boundary conditions. Take $\epsilon = u\frac{\Delta x}{2}$ and apply the boundary conditions (26) with $\alpha_N = \beta_N = 0.01$.

   [6 points]

b) To check whether the implementation is valid or not, we can assess whether the code reproduces an exact solution of a semi-discrete equation:

$$
\frac{U - u^n}{\Delta t} + U\partial_x U = -g\left(\partial_x H + \partial_x b + U^2 n^2 \left(\frac{w+2H}{wH}\right)^{4/3}\right) \tag{27}
$$

   - Show that if $U_{th}(x) = 10 + 0.01x$, $b(x) = 100 - 0.1x$, $H(x) = 120 - 0.1x$, $n = 0.01$, $w = 1$, then to have equation (27) satisfied, $u^n(x)$ should be:

$$
\begin{aligned}
u^n(x) &= (10 + 0.01x)(1 + 0.01\Delta t) \\
&+ g\Delta t\left(-0.2 + 0.01^2(10 + 0.01x)^2\left(\frac{1 + 240 - 0.2x}{120 - 0.1x}\right)^{4/3}\right)
\end{aligned} \tag{28}
$$

     [1 point]

   - Take the initial condition (28) and compute the error between $U_{th}(x)$ and $U_{computed}$ (the output of your program), using the function `norm` of Matlab. The error should be small if the code is working properly. One may want to use the provided template `momentum_transport.m`. [1 point]

     Bonus: Vary θ in the range $\langle 0.00001; 0.9\rangle$ and describe what you observe. [1 point]

3 The Saint-Venant equations

Warning : *To do this section, you need the codes from the previous exercises !*

3.1 Theoretical prerequisites

You have developed, in the previous part, all the tools that are required to solve the Saint-Venant system of equations that we recall here :

$$\begin{cases} \partial_t h + u\partial_x h + h\partial_x u = 0 \\ \qquad\qquad \partial_t u + u\partial_x u = -g\left(\partial_x h + \partial_x b + u^2 n^2 \left(\dfrac{w+2H}{wH}\right)^{4/3}\right) \end{cases} \tag{29}$$

The set of equations (1) requires 4 boundary conditions. For the modelling of our physical conditions we choose:

$$\begin{array}{ll} h(x = x_{min}) = h_L^{\text{Dirichlet}} & \partial_x h(x = x_{max}) = 0 \\ \partial_x u(x = x_{min}) = 0 & \partial_x u(x = x_{max}) = 0 \end{array} \tag{30}$$

- Show that the system of equations (29) is hyperbolic

  [2 points]

3.2 Numerical resolution of Saint-Venant equations

The method that will be used to solve the problem (29) is called a splitting method, because each equation will be solved one after one other, and not both at the same time.

In the first section, one saw how to solve the first equation for h, if one has the velocity u , and in the second part, one saw how to solve the second equation for u, if one has the water depth h.

First, copy your functions `continuity_build_matrix.m`, `continuity_apply_BC.m` and `solve_momentum_transport.m` to the folder `Saint-Venant`. Set $\theta = 0.1$ in `solve_momentum_transport.m`. Set $\epsilon_h = \frac{u_i \Delta x}{2}$ in `continuity_build_matrix.m`.

a) Change your boundary conditions of velocity to **homogeneous** Neumann conditions at both boundaries. [1 point]

b) Describe the script `Saint-Venant.m`: the physical problem being modelled and its parameters; the workflow of the script with brief notes about each step. [2 points]

c) After what time does the water depth at $x = 10$ km exceed 9.5 m? (assuming the initial conditions provided in `Saint-Venant.m`) [1 point]

d) Based on which criteria can you say whether a steady state has been reached or not? [1 point]