Assignment (due online on Sunday, 2020/5/24)

不用寫報告，線上繳交要求結果圖和程式碼即可。這次作業的程式架構還滿複雜的喔！

For this assignment, you **do not** need to write a report. Just make sure your code works, generate the required plots, and attach your code.

1. Implement the second-order accurate in space (MUSCL-TVD) and third-order accurate in time (SSP-RK) numerical scheme to solve the NSWE (see the handout for 2020/5/11).

   We will simulate water wave propagation in a constant water depth of $h = 0.3$ m. The numerical domain to consider is

$$\begin{cases} -4 \text{ m} \leq x \leq 10 \text{ m} \\ 0 \text{ s} \leq t \leq 3.5 \text{ s} \end{cases}. \tag{1}$$

   We will impose the mirror boundary condition on both ends since it is easy to specify.

   The initial wave has the shape of a solitary wave:

$$\eta(x,0) = H_0 \text{sech}^2\Big(Kx\Big), \quad K = \frac{1}{h}\sqrt{\frac{3H_0}{4h}}, \tag{2}$$

   where $H_0 = 0.15$ m is the wave height.

   The initial flow velocity is specified as

$$U(x,0) = \frac{\eta(x,0)}{h}\sqrt{gh}. \tag{3}$$

2. Run the simulation using three different grid sizes. Compare the results at $t = 3.5$ s in one plot; your results should look similar to Figure 1.

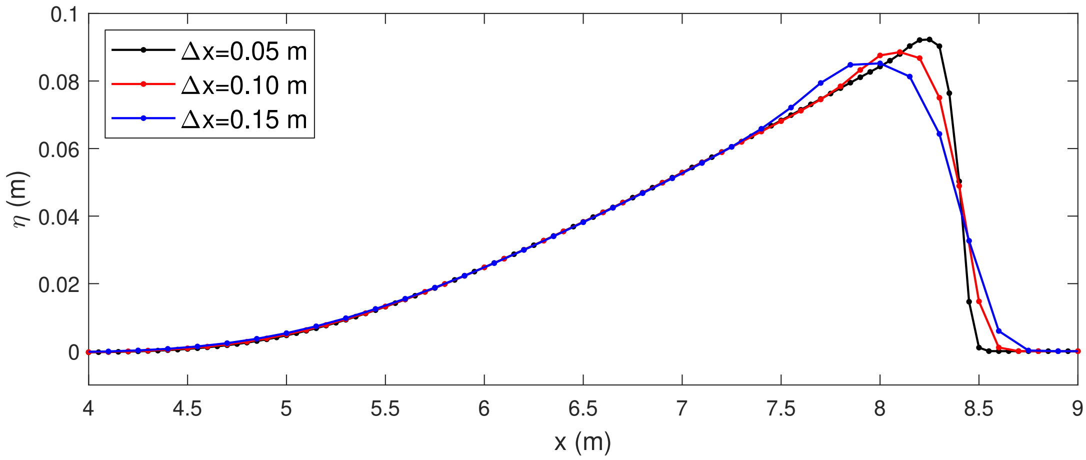

Figure 1: Numerical results at $t = 3.5$ s using different grid sizes.

3. Then, pick a grid size you like, and make a plot showing the evolution of this wave; for example, Figure 2 shows my results.

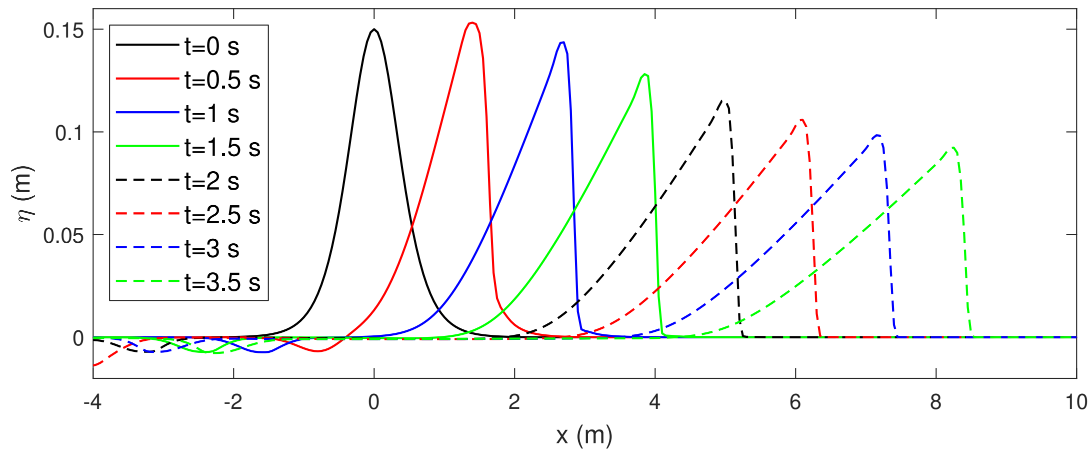

Figure 2: The evolution of a solitary wave with $H_0 = 0.15$ m in a constant water depth $h = 0.3$ m, computed based on NSWE with $\Delta x = 0.05$ m.

Tips

1. Remember, we now use a dynamic time step: Δt needs to be calculated at each time iteration, because it changes with U and H. Courant number $C_{\text{CFL}} = 0.9$ should be sufficient for this simulation.

2. The new MUSCL-TVD NSWE solver is computationally more expensive than our LSWE solver, so it is normal that the computation takes longer than what you are used to.

3. You can choose whether you want to work with η or H. The two are essentially the same thing because they are related by $H = \eta + h$. In more complicated problems, it is more advantageous to update H and $[HU]$ in time, but perform the MUSCL reconstruction on η and U. In this simple problem, you can do it any way you like; it's unlikely you will run into issues.

4. Make sure to impose the boundary condition on all the relevant variables. For example, in my code I have H, η, U, and $[HU]$ to set.

5. The overall structure of my code looks like this:

   [1] initialization

   [2] start a loop in time

   [3] — impose boundary conditions

   [4] — MUSCL reconstrucion of η and U

   [5] — HLLC Riemann solver to get numerical intercell fluxes F and G

   [6] — update H and $[HU]$ using SSP-RK

   [7] — recover η from updated H and U from updated $[HU]$

   [8] — (do three rounds of [3] to [7] as required by SSP-RK)

   [9] end the loop in time