ACSE-4.3-htmlview.ipynb

Hanran Ji hj19

Tianzong Yu ty616

Introduction

This programme is used to do post-processing about files output by:

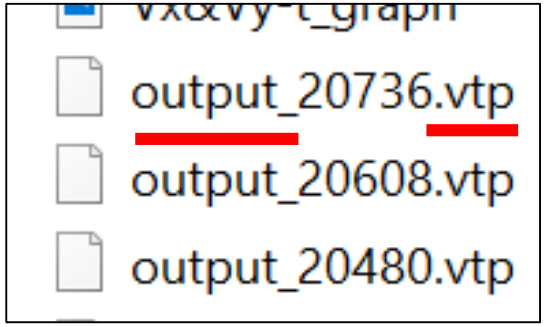
- `SPH_Snippet.cpp`,
- `SPH_Euler.cpp`,
- `file_writer.h & .cpp`,
- & SPH_2D.h & .cpp,

which are programmes to so calculation on smooth particle for ACSE-4.3 group assignment 2020.

Content

0. Prerequisite & constants
1. Work Streamline
2. Implementation
3. Peak-Time Plots
4. Animation & Water Parcel Distribution
5. Task 5 & Sloshing Interval

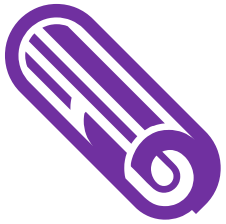
0 Prerequisite & Constants



e.g.
prefix_address = 'output-'
Suffix_address = '.vtp'

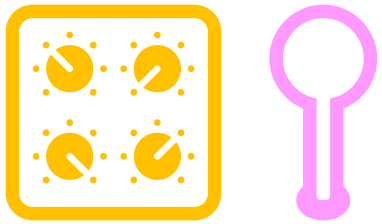
- Packages required;
- The .ipynb should be put in the same directory as output files.
- Other constants:
  - prefix_ & suffix_address: filename prefixes & suffixes (see top right corner)
  - no_max: maximum number of output file (see Pt. 2 for determination)
  - time_step: time step for each iteration (see Pt. 2 for determination)
  - total_t: total time, no_max*time_step
  - save_step: save per how many steps
  - items: total number of saved files, int(no_max/save_step)+1

1 –Streamline Overview



Read & split .vtp files

read_address



Get lists of physical quantities e.g. Vx, Vy, pressure

make_grid
fill_grid

1	2	3
[4	5	6]
7	…	…

Form and fill grids

find_peak

Get properties at peak

1 – read_address

Input	Output
Add: file address & name, str	0. length: no. of data in a file, int
	1. dim: dimension, int
	2. Pre_list: a list of pressure measured at each point, list
	3. V_list: a list of velocity measured at each point, including Vx & Vy as 2 sub-lists, list
	4. Point_list: a list of coordinates of each point, including x- & y-coordinate as 2 sub-lists, list

To read .vtp files and get pressure, velocities and co-ordinates.

1 – make_grid

Input	Output	
Point_list: Point_list from read_address, list of a lists	0. step: spatial step in grid, float	5. y_coord_grid: list of corresponding y indices on grid, np.array
	1. y_range: range of y coordinates, float	6. x_coord_grid: list of corresponding x indices on grid, np.array
	2. y_grid_size: no. of grids to be created on y-axis in grid, int	7. Vx_grid: initialised grid for velocities on x-axis, np.array
	3. x_range: range of x coordinates, float	8. Vy_grid: initialised grid for velocities on y-axis, np.array
	4. x_grid_size: no. of grids to be created on x-axis in grid, int	9. Pre_grid: initialised grid for pressure, np.array

To make grids based on coordinates from read_address.

1-fill_grid

Input		Output
y_coord_grid, list	V_list, list	Pre_grid, np.array
x_coord_grid, list	Pre_list, list	Vx_grid, np.array
Vx_grid, np.array	Length, int	Vy_grid, np.aray
Vy_grid, np.array		
Pre_grid, np.array	***all from previous functions***	***filled grids with corresponding quantities***

To fill the grid made from make_grid.

1-find_peak

Input	Output
*length, length of lists	0. k, maximum (peak) y-coordinate, float
*Pre_list, lists of pressure	2. peak_x_coord, x-coordinate at peak, float
*V_list, lists of velocities	3. Vy_max, v-direction velocity at peak, float
*Point_list, lists of coordinates	4. Vx_max, x-direction velocity at peak, float
**tol, a value that is considered to be out of the main water	5. Pre_max, pressure at peak, float
min_y=2, minimum y (in metres) to be considered where peak could possibly occur, default set as 2(m)	6. peak_index, index of peak in list, int
output from read_address	
****reserved as a filter, commented out in code***	

To find physical quantities and location at peak of the wave.

1- ensemble

Input	Output
no, file number to read, int or str	Output[0, 1, 2] from fill_grid
*tol, float	Output[6, 5] from make_grid
	Output[:] from peak
	Output[0] from make_grid
***same as find_peak input**	***All are outputs from other functions**

To join functions written previously together.

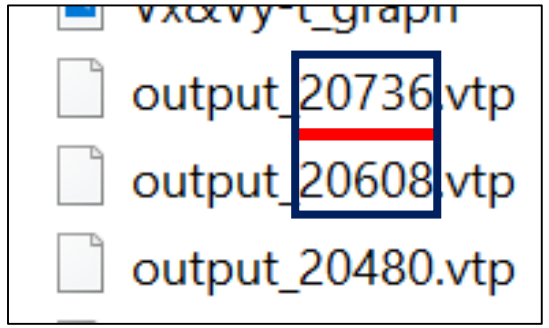
2-Implementation

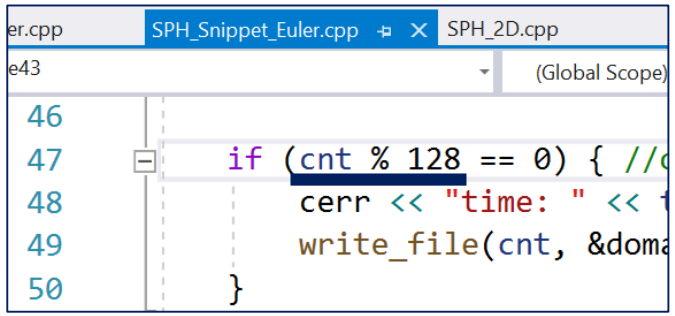
- After Pt.1's building functions, now they needed to be played with.
- A no_max is required along with save_step to find how many files to be read, also a final call on no_max is needed to obtain the last file output.

Clue to determine no_max & save_step:

open the directory of data -> find the file with maximum output number;

save_step = difference between two consecutive file number, or you can find in original c++ file (SPH_Snippet(_Euler).cpp)



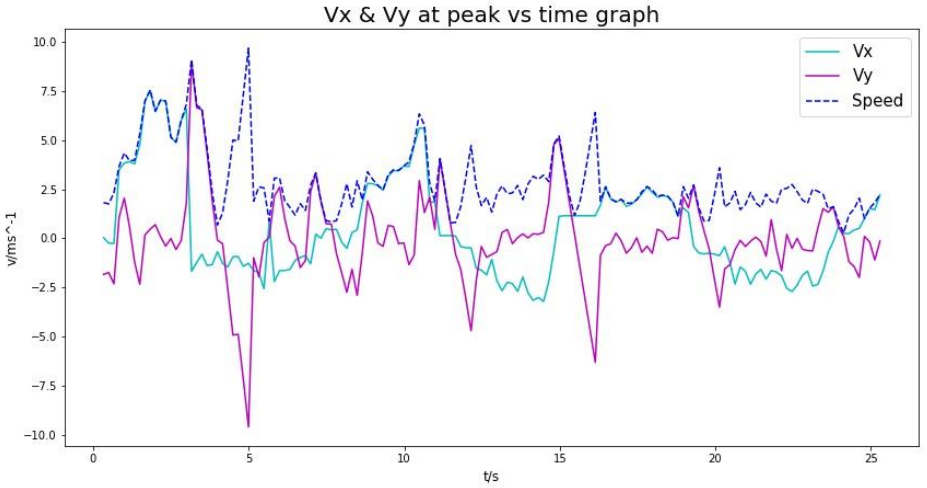


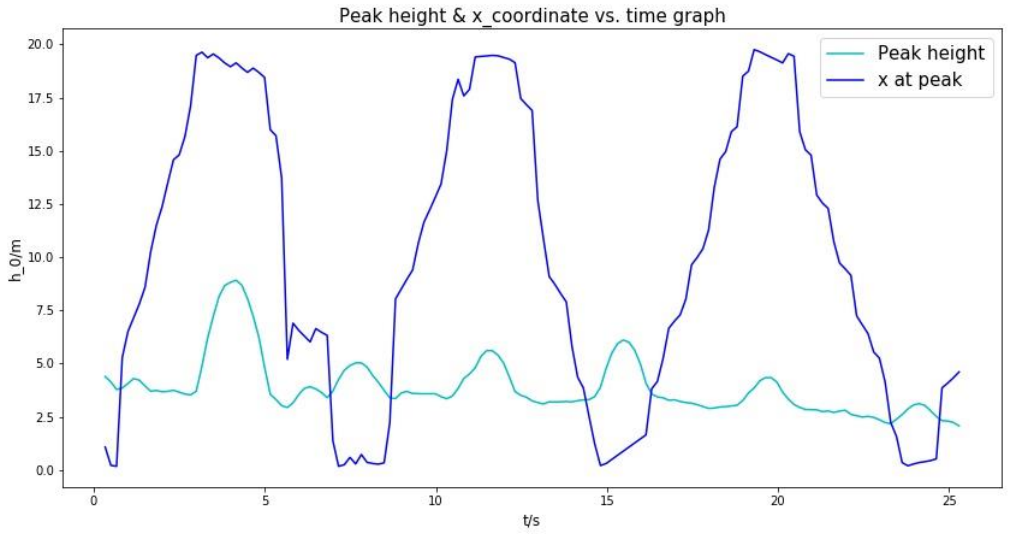
e.g.
no_max= 20736
Save_step=20736-20608=128

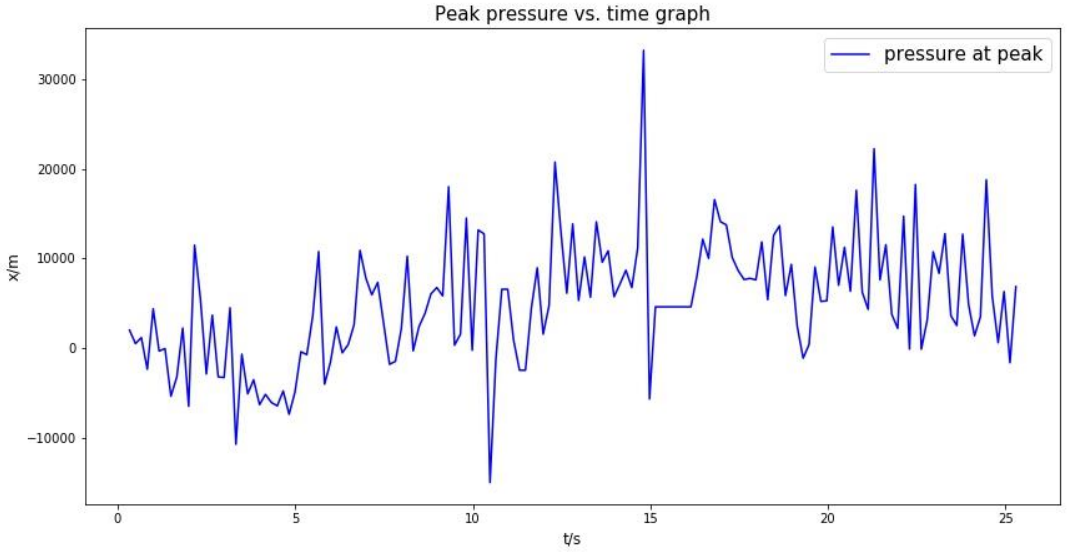
3 - Peak-time plot

- To plot coordinates & properties at peak against time, and save as .jpg.

X-axis	Y-axis
Time (sec)	X-coordinate
	Y-coordinate
	Velocity on x-&y-dir.
	Speed
	Pressure







4-Animation creating with HTML & plt.animation

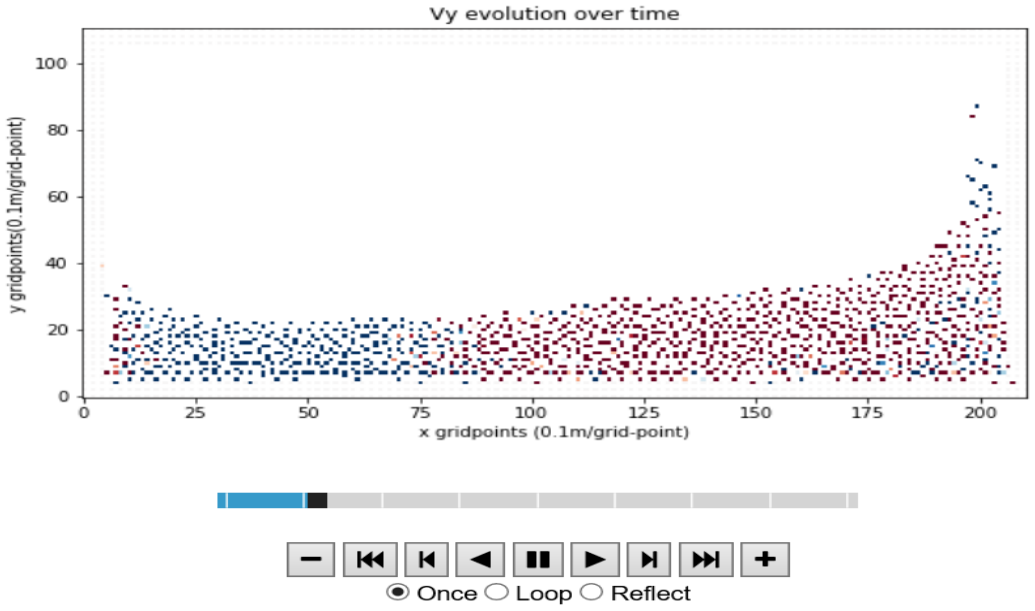
4.1 Shape of water distribution animation & peak tracker

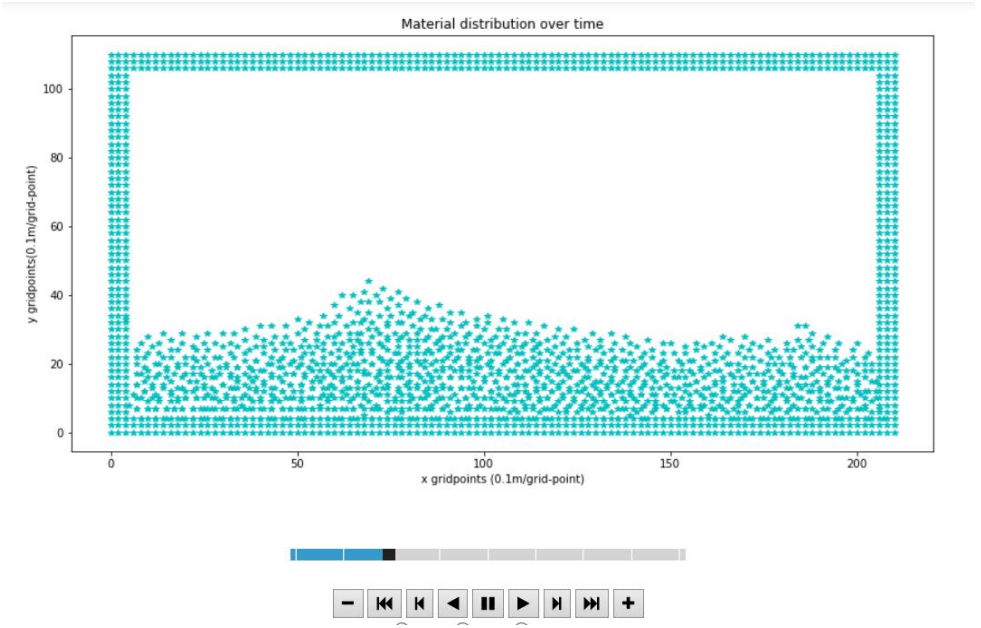
4.2 Vx&Vy meshgrid change over time

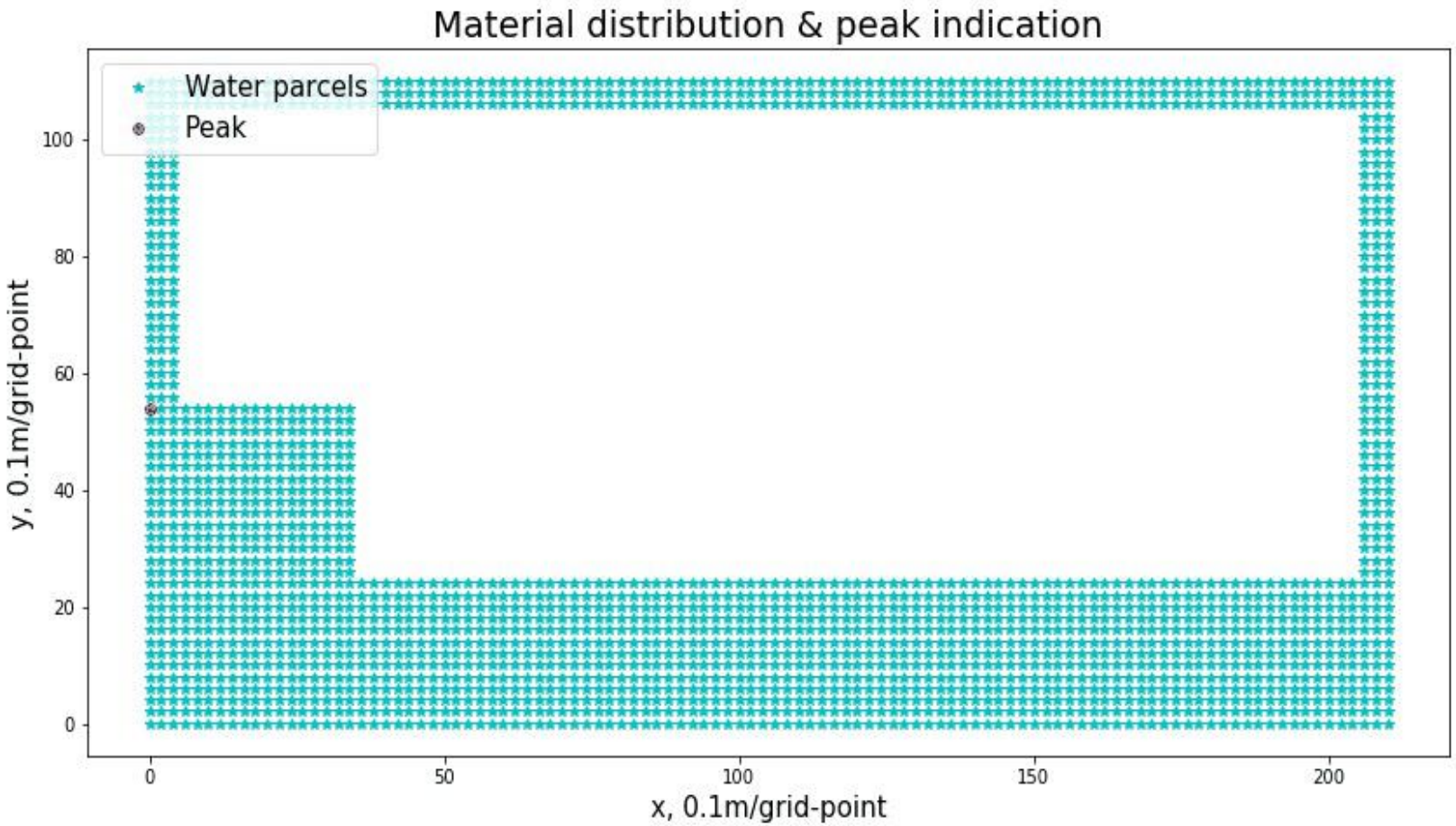
4.3 Speed meshgrid change over time

4.4 Pressure meshgrid change over time

4-Animation creating with HTML & plt.animation







need a folder /mdsave to save figures in the folder to save slices for making . gif

5 – Task 5

- Comparison between theoretical and experimental period between 2 adjacent sloshes.

Peak selection may have to be picked manually.