ISWFoam

1 Overview

ISWFoam is a transient solver for simulating internal solitary wave in continuously stratified, incompressible, viscous fluids, based on the open source code OpenFOAM-v1906.

2 Installation of ISWFoam

ISWFoam is straightforward to install. ISWFoam is available at https://github.com/Mr-trekking/ISW.git, which can be downloaded freely. The downloaded code includes the main program (ISWFoam-master), pre-processing (setRhoFields, setUFields), post-processing (postSensDensity.py) and verification tutorial.

Before installing ISWFoam, you should ensure that you have successfully installed OpenFOAM-v1906, which installation can refer to https://www.openfoam.com/releases/openfoam-v1906/.

ISWFoam is installed by executing the **wmake** command in the ISWFoam-master, setRhoFields and setUFields files, or directly executing the script prepared Allwmake through the **./Allwmake** command.

3 Runing tutorials

3.1 FlatBottom instructions

FlatBottom is a tutorial for the propagation and evolution of internal solitary waves along a flat bottom in continuously stratified fluid. The specific experimental settings are described in Hsieh et al. (2014).

3.2 wave generation

The method of initializing the field is selected to generate internal solitary waves specified according to weakly-nonlinear models that is the eKdV equation, which includes cubic nonlinearity. The pre-processing (setRhoFields, setUFields) program is used to generate internal solitary waves, by executing the **setRhoFields** and **setUFields** command, or directly executing the script prepared Allrun.pre through the **./Allrun.pre** command.

3.3 Parallel Computing

Perform single core calculation by command ISWFoam. The code has good parallel efficiency. It divides threads through **decomposePar** according to the file named decomposeParDict, and then executes parallel calculations through command:

mpirun -np *N* ISWFoam -parallel &>log

where ***N*** is the specified number of threads according to the file named decomposeParDict.

3.4 Post-processing (interface extraction)

In order to facilitate the application of the code, the post-processing script of the interface extraction has been provided, named **postSensDensity.py**. By executing the **./postSensDensity.py** command, the calculation data results in the *postProcessing* file are post-processed and stored in the *gaugesInterFace* file.

Reference

Hsieh, C. M., Hwang, R. R., Hsu, J. R. C., Cheng, M. H., 2014. Flow evolution of an internal solitary wave generated by gravity collapse. Applied Ocean Research. 48, 277-291. https://doi.org/10.1016/j.apor.2014.10.001.