ISWFoam

1 Overview

A numerical model, ISWFoam with a modified *k-ω SST* model, established by combining the density transport equation with a fully three-dimensional (3D) Navier-Stokes equation, is developed to simulate ISWs in continuously stratified, incompressible, viscous fluids based on the open source code OpenFOAM-v1906.

ISWFoam provides two initial wave generation methods to generate an ISW in continuously stratified fluids, including solving the weakly nonlinear models of the extended Korteweg-de Vries (eKdV) equation and the fully nonlinear models of the Dubreil-Jacotin-Long (DJL) equation. We use the DJLES open source package provided by Dunphy et al (2011) to solve the DJL equations. Then we input the initial field calculated by DJLES into OpenFOAM to obtain the initial field required for OpenFOAM numerical simulation.

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), the turbulence model (densityTurbulenceModels-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, densityTurbulenceModels-master, setRhoFields and setUFields files, or directly executing the script prepared Allwmake through the **./Allwmake** command.

3 Runing tutorials

3.1 FlatBottom-eKdV instructions

FlatBottom-eKdV is a tutorial for the propagation and evolution of ISWs generated by the eKdV equation along a flat bottom in continuously stratified fluid.

3.1.1 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.

3.1.2 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.

3.1.3 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.

3.2 FlatBottom-DJLES instructions

FlatBottom-eKdV is a tutorial for the propagation and evolution of ISWs generated by the DJLES equation along a flat bottom in continuously stratified fluid.

3.1.1 wave generation

A fully nonlinear models of the DJL equation is selected to generate ISWs. We use the DJLES open source package provided by Dunphy et al (2011) to solve the DJL equations.

3.1.2 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.

3.1.3 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

Dunphy, M., Subich, C., Stastna, M., 2011. Spectral methods for internal waves: indistinguishable density profiles and double-humped solitary waves. Nonlinear Processes in Geophysics, 18(3), 351-358. https://doi.org/10.5194/npg-18-351-2011.