ORCTM

1 Overview of ORCTM

An Oceanic Regional Circulation and Tide Model (ORCTM) including the nonhydrostatic dynamics can numerically reproduce the Internal Solitary Waves (ISWs) dynamics. The ORCTM has been developed from the basic framework of the Max-Planck-Institute ocean global model (Marsland et al., 2003). Based on the fractional step and the pressure correction methods (Press et al., 1988; Stansby and Zhou, 1998), the three-dimensional fully nonlinear momentum equations are involved and considered thoroughly under the Boussinesq fluid, and the nonhydrostatic dynamics simulation can be fulfilled economically comparatively in harmony with the original physical framework across the regional ocean.

2 Getting Started with ORCTM

The ORCTM can designed to run in the parallel computing with MPI (Message Passing Interface). The distribution of processors along the *x* and *y* directions can be defined with the variables nprocx and nprocy in the beginning of the namelist *OCECTL*, which namely means that the number of processors equals to nprocx × nprocy. The ORCTM code can be downloaded freely through the Github (https://github.com/HuangOCEAN02/ORCTM-v1.git).

Before the build of ORCTM, we must install the PETSc library (Portable, Extensible Toolkit for Scientific Computation, see https://petsc.org/release/) to employ the Krylov Subspace Methods to solve the Poisson equation. Here, we provide the petsc-3.10.2 zip file at the above link and the straightforward way to install the PETSs library. Firstly, we need compile the PETSc programs in **arch-linux2-c-debug** folder so that the ORCTM can call their functions freely. In the petsc-3.10.2 root directory we provide the users with a brief build shell script *petsc_conf_make.sh*. Secondly, the MPI libraries environment (Fortran and C) information including the blaslapack and compiler paths need to be set based on your MPI libraries environment in the beginning of this build script. Finally, run the *petsc_conf_make.sh* and compile the PETSc programs successfully in **arch-linux2-c-debug.**

Once installing the PETSc programs, we can build the ORCTM effortlessly via linking above PETSc program compiled. There are five subdirectories existing in every experiment working place: **orctm_src**, **BT_forc**, **obcs**, **diagnosis**, and **post_process**. In the **orctm_src** folder lies ORCTM source code, and we can use the **make** command to build the ORCTM. A file called *Makefile* allows the users to pro-process source files, specify the PETSc program pathes and optimization options. Please make sure that you define the environment variable PETSC_ARCH and PETSC_DIR correctly which represent the **arch-linux2-c-debug** location and PETSc root directory. ORCTM is installed by executing the ***make*** command and generate the executable file *orctm_nh.x***.**

The **BT_forc** provides the pre-processing file *mkgrid.f90* (Non-paralleled Fortran 90 program) which can generate the input binary files including the model grid files, initial fields and forcing fields. The **obcs** provides the open boundary conditions generated file *Gen_boundary.m* (MATLAB program) involved in the temperature, salinity, and amplitude and phase of the tidal current and elevation with frequency defined in the namelist file *OCECTL*, which can be controlled by the **OBC_TS_FORC**, **OBC_tide**, **OBC_UV_FORC**, and **OBC_ETA_FORC** options in *Makefile*. By default, the master MPI process can take charge of the I/O to produce the output "global" binary files. The code table for ORCTM output is shown in the file *code_output.txt*. In addition, the ORCTM can support to write output separately for individual processors, leaving it to the users to reassemble these into global files, which can be controlled by the **FRAGMENT** option in *Makefile*. If needed, the f90 file **Glue**/*Glue_new.f90* (Paralleled Fortran 90 program) in the **post_process** can do this to generate the global netcdf files, and requires the Pnetcdf libraries given in advance when compiling. The **diagnosis** folder can store the outputs for specified points in the model domain, and the coordinates and numbers about the specified points need to be prescribed in the namelist file *OCECTL* and the source code file *mo_diagnosis.f90* in advance, which above can be controlled by the **DIAGNOSIS** option in *Makefile*.

3 Nonhydrostatic example experiments

Here, the ORCTM provides the five numerical experiments in the **verification** directory about the nonhydrostatic dynamics from the laboratory-scale cases in a tank to field-scale one like the northern South China Sea. The laboratory-scale cases mainly focus on the Kelvin-Helmholtz instability and Internal Solitary Wave dynamics, and the field-scale experiment can reproduce the nonlinear tide-topography interactions via the tidal boundary forcing.

The first case (Sect. 3.1) is the lock-exchange problem. The second to fourth cases (Sect. 3.2–3.4) are designed to explore the nonlinear evolution of Internal Solitary Wave caused by its interactions with the different terrains. The last one (Sect. 3.5) is the generated nonlinear internal waves case in a double-ridge environment analogous to the Luzon Strait, which aims at the large-amplitude ISW packets disintegrated from the multi-modal baroclinic tides. As shown below, we provide the shell scripts ***restart_nonhydro.sh*** for the users to run these experiments at every working place.

3.1 The lock-exchange problem

The hydrostatic simulation needs to switch off the **NON_HYDROSTATIC** option in *Makefile*.

OCECTL:

namelist

BT_forc:

mkgrid.f90: generate the grid and topography files, initial fields files, and forcing fields files

src_orctm:

source code

post_process:

post_process.sh: generate the output netecdf files

3.2 Internal Solitary Wave in a tank

The case employs the **DIAGNOSIS** option.

OCECTL:

namelist

BT_forc:

mkgrid.f90: generate the grid and topography files, initial fields files, and forcing fields files

src_orctm:

source code

diagnosis:

High-frequency diagnostic outputs

post_process:

post_process.sh: generate the output netcdf files.

3.3 Internal Solitary Wave shoaling on a Gaussian terrain

The case needs to turn on the **DIAGNOSIS** option.

OCECTL:

namelist

BT_forc:

mkgrid.f90: generate the grid and topography files, initial fields files, and forcing fields files

src_orctm:

source code

diagnosis:

High-frequency diagnostic outputs

post_process:

post_process.sh: generate the output netecdf files.

3.4 Internal Solitary Wave breaking on a slope.

OCECTL:

namelist

BT_forc:

mkgrid.f90: generate the grid and topography files, initial fields files, and forcing fields files

src_orctm:

source code

post_process:

post_process.sh: generate the output netecdf files.

3.5 Nonlinear Internal Waves in a double-ridge system

The case needs to turn on the **OBC_TS_FORC**, **OBC_tide**, **OBC_UV_FORC, DIAGNOSIS** and **FRAGMENT** options. Therefore, the fragment reassembling can be fulfilled via *Glue_new.f90* to generate the output netcdf files.

OCECTL:

namelist

BT_forc:

mkgrid.f90: generate the grid files, initial fields, and forcing fields

obcs:

Gen_boundary.m: generate the boundary input files

src_orctm:

source code

diagnosis:

High-frequency diagnostic outputs

post_process:

Glue/*Glue_new.f90*: reassemble the outputs into the global files

post_process.sh: generate the output netecdf files.

Reference:

Marsland, S. J., Haak, H., Jungclaus, J. H., Latif, M. and Röske, F.: The Max-Planck-Institute global ocean/sea ice model with orthogonal curvilinear coordinates, Ocean. Model., 5, 91-127, https://doi.org/10.1016/S1463-5003(02)00015-X, 2003.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., Numerical Recipes in C. The Art of Scientific Computing, Cambridge University Press, 1988.

Stansby, P. K., and Zhou, J. G.: Shallow-water flow solver with non-hydrostatic pressure: 2D vertical plane problems, Int. J. Numer. Meth. Fluids, 28, 541-563, https://doi.org/10.1002/(SICI)1097-0363(19980915)28:33.0.CO;2-0, 1998.