

# User Manual of the Finite Volume WAVE Modeling (FVWAM)

## 1. Structure of directory

### 1.1. **code** (Directory of source codes)

1.1.1. **graph\_topology\_false** (the FVWAM model using the distributed graph communication topology without reordered processes)

1.1.1.1. **lib** (the library used by the FVWAM model)

1.1.1.2. **source** (the source code of the FVWAM model)

1.1.2. **graph\_topology\_true** (the FVWAM model using the distributed graph communication topology with reordered processes)

1.1.2.1. **lib** (the library used by the FVWAM model)

1.1.2.2. **source** (the source code of the FVWAM model)

1.1.3. **point\_to\_point** (the FVWAM model using the point-to-point communication method)

1.1.3.1. **lib** (the library used by the FVWAM model)

1.1.3.2. **source** (the source code of the FVWAM model)

### 1.2. **Example** (Directory of the test case)

1.2.1. **glo\_10km** ( the test case with the global 10km resolution)

1.2.1.1. **Partition** (the files of the grid partitioning results)

1.2.1.2. **TC\_parameter\_nc** ( the input files of wind forcing)

## 2. Operation procedure of the FVWAM

### 2.1. Operating environment requirements

#### 2.1.1.1. Pre-processing environment requirements

Install METIS version 5.0 or later. (For downloading and installing METIS, please refer to:  
<http://glaros.dtc.umn.edu/gkhome/metis/metis/download> )

#### 2.1.1.2. Runtime environment requirements

Install the MPI library (recommended MPI version 3.0 or later which must support MPI\_NEIGHBOR\_ALLTOALLV function interface), and NetCDF Fortran version 4.0 or later. If using the Intel compiler, Intel compiler version 2019 or later is required.

#### 2.1.1.3. Pre-processing workflow

This workflow generates the files of grid partitioning results for the FV-WAM model. The grid partitioning files for 512, 1024, 2048, 4096, 8192, 16384, and 32768 processes in the test case of the paper have been already generated, this step can be skipped. If you would like to generate a specified number of processes, please follow this step.

It's using the installed METIS software and the adjacency graph file "mesh\_global\_10km.tmppartition.graph" in the directory of "./example/glo\_10km/partition" to generate the grid partitioning result files. In a Linux Shell environment, the command is as follows, where "-ubvec=1.001" indicates that the grid quantity difference among processes does not exceed 0.1% (This parameter can be adjusted. Based on the preliminary tests, 1.001 is recommended), and 256 (it is the number of processes, and should be specified by user) represents the specified number of parallel computation processes. After executing this command, the grid partitioning result file "mesh\_global\_10km.tmppartition.graph.part.256" will be generated):

```
gpmmetis -ubvec=1.001 mesh_global_10km.tmppartition.graph 256
```

### **3. Generate the executable file**

#### **3.1. Configure the Makefile**

In the code directory, please edit the "Makefile" to set up the compiler, compiling options, and NETCDF library path (automatically configured via "nf-config") according to the operating environment. When using the Intel compiler with the "O3" optimization option. If using the cluster of the National Supercomputing Center of China in Jinan as described in the paper, the "Makefile" configuration is as follows. The file of "depend.mk" includes dependencies of compilation.

```
FC = mpiifort  
FFLAGS = -O3  
NC_LIB = $(shell nf-config --flibs)  
NC_INC = -I$(shell nf-config --includedir)  
LDFLAGS = $(NC_INC)  
LIBS = $(NC_LIB)  
.SUFFIXES:  
.SUFFIXES: .F90 .o
```

*.F90.o:*

*\$(FC) -c \$(FFLAGS) \$(LDFLAGS) \$<*

*include depend.mk*

*clean :*

*rm -f \*.o \*.mod \*.exe*

*depend depend.mk:*

*makedepf90 -o wave.exe \*.F90 > depend.mk*

### **3.2. Generate the executable file**

Execute the “make” command to generate the model executable file of “wave.exe”.

### **3.3. Run the FVWAM model**

#### **4.1. Link the FVWAM executable file**

Switch to the model example directory and link the compiled executable file “wave.exe” to the directory where the model example is located as the following command.

*ln -s <path\_to\_wave.exe> ./wave.exe*

#### **4.2. Modify model running parameters**

According to forecast requirements, modify the model parameter configuration file “namelist.wave” as needed. The example of “namelist.wave” in the test case of the paper is as following.

*&time\_control*

*start\_time = 2017, 08, 21, 00      (starting time for wave forecasting)*

*end\_time      = 2017, 08, 21, 01    (ending time for wave forecasting)*

*dt                = 15                      (time for each iteration, the unit of dt is second)*

*dt\_src\_ratio = 5                      (ratio for computing source term and advection term)*

*/*

*&io\_control*

*history\_interval      = "120 minutes"                      (output interval)*

*output\_file\_prefix    = "wave\_global\_10km"                      (prefix name of the output file)*

*frames\_per\_file        = "1 days"                      (forecasting period per an output file)*

*mesh\_file\_path        = "mesh\_global\_10km\_init.nc" (mesh file name)*

*/*

```

&wave_setting
case_name              = "wave" (case name)
tc_forcing_file_path=
"/TC_parameter_nc/CMA-STI_1713_2017-08-21_00:00:00_2017-08-24_00:00:00.nc" (input file
of wind forcing)

atm_forcing_interval   = "2 hours" (wind forcing interval)
atm_forcing_option     = 2   (1 is numerical weather forecasting result, 2 is tropical
                             cyclone forecasting result)
nDir                   = 36 (number of wave directions)
nFre                   = 35 (number of wave frequencies)
freMin                 = 0.03715 (minimum value of wave frequency)
fetch                  = 30000.0 (length of wind zone)

```

#### 4.3. Copy grid partition file

Based on the number of computing processes, the FVWAM model reads the *"tmppartition"* file as the input for parallel partitioning results in default. Before running the model, update this file according to the number of computing processes. For example, if the model needs to run on 512 computing processes, execute the following command in a Linux Shell environment, noting that the first parameter is the grid partitioning file that meets the number of computing processes, and the second parameter is the fixed file name.

```
cp ./partition/mesh_global_10km.tmppartition.graph.part.512 tmppartition
```

#### 4.4. Submit job

Submit a job with the same number of processes as specified in the grid partition file *"tmppartition"* to run the model file *"wave.exe"*. For example, the job submission command on the computing cluster in the paper is as follows, where *"-n 512"* specifies the total number of computation processes (512), *"-q production"* specifies the job queue, *"-J nmeffc\_model"* specifies the job name, and *"./wave.exe"* specifies the executable file name.

```
bsub -q production -n 512 -J nmeffc_model ./wave.exe
```