# JAGURS Users Guide

ver. 2024.06.03

For JAGURS-D_V0600

# 1. Contents

## 2. Introduction

This document describes how to use JAGURS with sample data. JAGURS is a numerical code that computes tsunami propagation and inundation on the basis of the long waves, or the dispersive waves (Boussinesq-type). The code can take into account the effects of elastic deformation of the Earth caused by the weight of tsunami and variations in seawater density stratification in the vertical profile. These are solved on a finite difference scheme using a staggered grid and the leapfrog method. The calculations are performed in a spherical coordinate system or a Cartesian coordinate system. A nesting of terrain grids can be applied. A range of other functions are also available. The code is written in Fortran 90 with parallelization using OpenMP and MPI. References used in the development of JAGURS are listed below.

References

[Non-linear long waves]

Satake, K. (2002). Tsunamis, in International Handbook of Earthquake and Engineering Seismology, (eds. Lee, W.H.K., Kanamori, H., Jennings, P.C., and Kisslinger, C.) (Academic Press 2002) 81A, pp. 437–451.

[Nesting algorithm]

Jakeman, J.D., O.M., Nielsen, K., Vanputten, R., Mleczeko, D. Burbidge, and N. Horspool (2010). Towards spatially distributed quantitative assessment of tsunami inundation models, Ocean Dynamics. https://doi.org/10.1007/s10236-010-0312-4

[Parallel computing]

Baba, T., N. Takahashi, Y. Kaneda, Y. Inazawa and M. Kikkojin (2014). Tsunami Inundation Modeling of the 2011 Tohoku Earthquake using Three-Dimensional Building Data for Sendai, Miyagi Prefecture, Japan, Tsunami Events and Lessons Learned, Advances in Natural and Technological Hazards Research, 35, pp.89-98. https://doi.org/10. 1007/978-94-007-7269-4_3

Baba, T., K. Ando, D. Matsuoka, M. Hyodo, T. Hori, N. Takahashi, R. Obayashi, Y. Imato, D. Kitamura, H. Uehara, T. Kato, R. Saka (2015). Large-scale, high-speed tsunami prediction for the great Nankai trough earthquake on the K computer, Inter. Jour. of High Per. Comp. App. https://doi.org/10.1177/1094342015584090

[Dispersive waves]

Saito T., K. Satake, and T. Furumura (2010), Tsunami waveform inversion including dispersive waves: the 2004 earthquake off Kii Peninsula, Japan, J. Geophys. Res., 115, B06303, https://doi.org/10.1029/2009JB006884, 2010.

Baba, T., N. Takahashi, Y. Kaneda, K. Ando, D. Matsuoka, and T. Kato (2015). Parallel implementation of dispersive tsunami wave modeling with a nesting algorithm for the 2011 Tohoku tsunami”, Pure appl. Geophys. https://doi.org/10.1007/s00024-015-1049-2

[Effects of elastic loading and seawater density stratification]

Allgeyer, S., and P. Cummins (2014). Numerical tsunami simulation including elastic loading and seawater density stratification, Geophys. Res. Lett., 41, 2368–2375. https://doi.org/10.1002/2014GL059348

Baba, T., S. Allgeyer, J. Hossen, P.R. Cummins, H. Tsushima, K. Imai, K. Yamashita, and T. Kato (2017). Accurate numerical simulation of the far-field tsunami caused by the 2011 Tohoku earthquake, including the effects of Boussinesq dispersion, seawater density stratification, elastic loading, and gravitational potential change, Ocean Modelling, 111, 46-54, https://doi.org/10.1016/j.ocemod.2017.01.002

[Coupling with atmospheric pressure waves]
Mizutani, A., & Yomogida, K. (2023). Source estimation of the tsunami later phases associated with the 2022 Hunga Tonga volcanic eruption. Geophysical Journal International, ggad174. doi: 10.1093/gji/ggad174

[Others]
Kajiura, K. (1963). The leading wave of a tsunami, Bull. Earthq. Res. Inst., Vol. 41, pp. 535-571.
Tanioka, Y. and K. Satake (1996), Tsunami generation by horizontal displacement of ocean bottom, Geophys. Res. Lett., Vol. 23, pp. 861-864, 1996.

## 3. Installing libraries and compiling JAGURS

JAGURS uses three external libraries of NetCDF, PROJ4, FFTW3. We should install the libraries before compiling JAGURS. Please refer each “README” in the library package for installing the libraries. “README” in JAGURS package also includes some descriptions for installing the libraries.

### 3-1. NetCDF

This is a library to handle NetCDF formatted data. JAGURS after V0406 version needs NetCDF4 so that another library of HDF5 is required to be install.

Download site: http://www.unidata.ucar.edu/software/netcdf/

### 3-2. PROJ4

PROJ4 converts coordinates between spherical and Cartesian. This library is used in calculations for crustal deformation and Kajiura filter.

Download site: https://trac.osgeo.org/proj/

### 3-3. FFTW3

FFT3 computes discrete Fourier transform. This library is used in calculations for Kajiura filter and elastic loading effect.

Download site: http://www.fftw.org/

### 3-4. Compiling JAGURS

JAGURS-D_V0502/ includes all source files of JAGURS.

(1) In “Makefile”, the libraries paths are correctly mentioned for your environment, for example,

```
NETCDF=/opt/atlocal/netcdf/4.1.3                ←path for NetCDF library
PROJ4_DIR=/home/G10004/t-katou/JAGURS/local     ←path for PROJ4 library
FFTW3_INCLUDE_DIR=$(MKLROOT)/include/fftw       ←path for FFTW3 library
```

(2) Compiling JAGURS by following command.

```
$ make -f Makefile
```

Several functions of JAGURS such as changing coordinate systems (spherical or Cartesian) or degree of parallelization (MPI or non-MPI) are selected by compile options. For detail about the compile options, please refer table in 8-1. Compile options.

## 4. Sample Dataset

### 4-1. Sample data list

The sample dataset includes the following files.

[Source code]
JAGURS-D_V0502/ ... complete source code

[Terrain datafiles]
bathy.SD01.grd, bathy.SD02.grd, bathy.SD03.grd, bathy.SD04.grd, bathy.SD05.grd

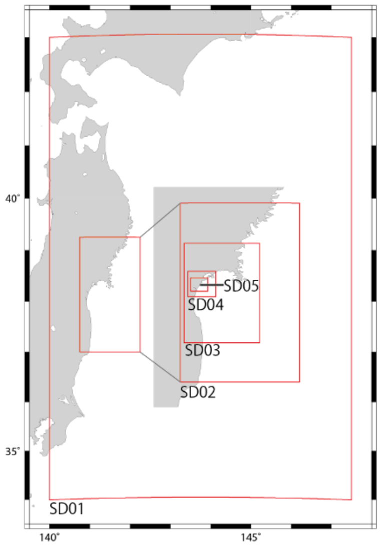
[Crustal displacement datafiles]
disp.SD01.grd, disp.SD02.grd, disp.SD03.grd, disp.SD04.grd, disp.SD05.grd

[WetOrDry file]
wetordry.SD05.grd ... a file that explicitly distinguishes between land and sea

[Elastic loading file]
PREM_Ggz.nc ... A green function representing elastic deformation of the crust by a point unit load (NetCDF4 format, See 7-2.C. Elastic loading and seawater compressibility)

[Parameter files]
gridfile.dat ... a parameter file describing parent-child relationships in nesting
test_tgs.txt ... a parameter file describing waveform output points
tsun.par ... a computing parameter file

[Script file]
qsub.sh ... a queue addition script
(written for the JAMSTEC system)




The sample dataset is constituted with five layers of terrain grid nesting centered on Sendai City, Tohoku, Japan. The grids are given sequential IDs starting from the domain with the largest area. The grid cells in the gridfiles of the different domains have size ratios of 3:1.

SD01 ... 18 arc-seconds interval
SD02 ... 6 arc-seconds interval
SD03 ... 2 arc-seconds interval
SD04 ... 2/3 arc-seconds interval
SD05 ... 2/9 arc-seconds interval

## 4-2. JAGURS-D_V0502/ – Complete source code

This directory contains the JAGURS source codes.

## 4-3. bathy.xxxx.grd – Terrain datafiles

These files contain topographic data of land and sea.

- Example of filenames ... bathy.SD01.grd—bathy.SD05.grd
- File format ... GMT gridfile [(#1)]
- Data orientation ... Vertical downward (negative values for land, positive values for sea depths), unit: m

## 4-4. disp.xxxx.grd – Crustal displacement datafiles

These files store data on crustal displacement (the initial sea surface condition). Grid regions and grid cell sizes of the crustal displacement datafiles have one-to-one correspondences with the terrain datafiles in all domains. If disp.grd is not used, "NO_DISPLACEMENT_FILE_GIVEN" is mentioned on the column instead of file name. In this case, the initial sea surface is not given, that is still zero. However, the initial sea surface distribution is interpolated from disp.grd for the root domain, when "init_disp_interpolation=1" is used in "tsun.par".

- Example of filenames ... disp.SD01.grd—disp.SD05.grd
- File format ... GMT gridfile [(#1)]
- Data orientation ... Vertical upward (positive values for lifting), unit: m

## 4-5. wetordry.xxxx.grd – WetOrDry files (optional)

These files are used for explicitly distinguishing land or sea at the start of computing. These are used giving an area of land below the sea level (0m) to be dry condition at the initial. When the WetOrDry file is not specified or "NO_WETORDRY_FILE_GIVEN" is mentioned, dry or wet condition is determined based on the elevation in the terrain datafiles.

Grid regions and grid cell sizes in the WetOrDry datafiles have one-to-one correspondences with the terrain datafiles in domains.

- Example of filename ... wetordry.SD05.grd
- File format ... GMT gridfile [(#1)]
- Data contents ... negative values for dry (land), positive values for wet (sea)

---

[(#1)] Note that the GMT default format is the nf format (#18). But input and output files in JAGURS-D_V0516 or earlier are written in the GMT cf format (#10). In JAGURS-D_V0520 or later, both of nf format (#18) and cf format (#10) are available. File format of each domain may be different from each other. But file format must be the identical on a domain.

## 4-6. Parameter files

### A. gridfile.dat – Parameter file defining terrain nesting

This file describes parent-child relationships in nesting and whether or not optional files are used. From the head of the file, the files are listed in the order: parent gridfiles, child gridfiles, grandchild gridfiles, etc. Multiple children can be described for a single parent.

• Example of filenames ... gridfile.dat

[Description Example]

```
SD01  SD01  1  bathy.SD01.grd  disp.SD01.grd                        ←parents
SD02  SD01  0  bathy.SD02.grd  disp.SD02.grd                        ←children
SD03  SD02  0  bathy.SD03.grd  disp.SD03.grd                        ←grandchildren
SD04  SD03  0  bathy.SD04.grd  disp.SD04.grd                        ←great-grandchildren
SD05  SD04  0  bathy.SD05.grd  disp.SD05.grd  wetordry.SD05.grd     ←great-great-grandchildren
```

• Delimiter: space
  • If the last line is blank, an error is issued.
• The columns are as follows:
  Column 1: Terrain ID
  Column 2: ID of that terrain ID's parent (for the ultimate parent, its own ID is given)
  Column 3: Linear (1) or non-linear (0) long-wave equations (linear/non-linear can be specified at the grid cell level)
  Column 4: Terrain datafile name
  Column 5: Crustal displacement datafile name
  Column 6: WetOrDry datafile name (optional)

### B. test_tgs.txt – Parameter file specifying waveform output points

This file specifies the coordinates of points (stations) where waveforms should be outputted, such as tide-gauge stations (tidal stations) and ocean bottom pressure gauges. The grid points closest to the specified coordinates are used as the output points.

• Example of filename ... test_tgs.txt

[Description Example]

```
6                                        ←Total number of stations
    40.116667    142.0666667  1  #GPS807
    39.627222    142.1866667  2  #GPS804
    39.258611    142.0969444  3  #GPS802
    38.857778    141.8944444  4  #GPS803
    38.2325      141.6836111  5  #GPS801
    36.971389    141.1855556  6  #GPS806
```

• Delimiter: space
• The columns are as follows:
  Line 1 ... Column 1: total number of stations

Line 2, etc. ... Column 1: station latitude
Column 2: station longitude
Column 3: station number (avoiding duplication)
From # to end of line: comment

C. tsun.par – Computing parameter file

This file specifies the parameters required for computing. Do not omit "&params" in the first line and "/" at the end. For details of the parameter file, see "8-2. Parameter list for running JAGURS"

• Example of filename ... tsun.par

[Description Example]

| | |
|---|---|
| &params | ←Marks start of parameters |
| gridfile="gridfile.dat" | ←Parameter filename defining terrain nesting parent-child relationships |
| maxgrdfn="zmax.grd" | ←Maximum wave height filename |
| vmaxgrdfn="vmax.grd" | ←Maximum velocity filename |
| tgstafn="test_tgs.txt" | ←File specifying waveform output points |
| dt=0.05 | ←Time step width (s) |
| tend=120 | ←Computing stop time (s) |
| itmap=1200 | ←Water height and velocity damping out interval (steps) |
| tau=60 | ←Rise time (s) |
| cf=-0.025 | ←Friction coefficient at the sea |
| cfl=-0.025 | ←Friction coefficient at the land |
| coriolis=0 | ←Coriolis force (1=enable, 0=disabled) |
| c2p_all=1 | ←Copy data from child domain to parent domain (1=enable, 0=no processing) |
| def_bathy=1 | ←Deform terrain in accordance with crustal displacement (1=enable, 0=no processing) |
| plotgrd=-1 | ←Water height gridfile ID to output (–1=output all domains) |
| velgrd=0 | ←Velocity gridfiles to output (1=output, 0=do not output) |
| !procx=2 | ←Number of divisions in gridfile in east–west direction (used in the Parallel Version) |
| !procy=4 | ←Number of divisions in gridfile in north–south direction (used in the Parallel Version) |
| / | ←Marks end of parameters |

• Delimiters: space/tab character
• Commented out: from "!" to end of line

• dt

This parameter sets the computing time step interval. To define it, use the GMT grdinfo command to extract a maximum value (zmax) of z from the terrain datafile of each domain and put each value of zmax into the equation below. Computing will be stable if the interval used during the execution of JAGURS is adequately smaller than the smallest of the values of dt calculated for all domains.

$$\frac{\Delta x}{\sqrt{2gh}}$$

| ID | Grid(Sec.) | Grid(m) | zmax(m) | dt |
|---|---|---|---|---|
| SD01 | 18 | 450 | 9788.53125 | 1.027367238 |
| SD02 | 6 | 150 | 2310.686523 | 0.704843514 |
| SD03 | 2 | 50 | 331.7849121 | 0.620031355 |
| SD04 | 2/3 | 16.66666667 | 36.26066589 | 0.625176761 |
| SD05 | 2/9 | 5.555555556 | 31.81464577 | 0.222477407 |
| SD06 | 2/27 | 1.851851852 | | |

g ... gravitational acceleration (9.8 $m/s^2$)
h ... maximum depth (zmax (m))
Δx ... spatial grid interval (grid (m))

• tau

Specifies the time until seawater lifts up after earthquake starts.

• cf/cfl

Friction coefficients at the sea bottom. A positive value is treated as a dimensionless friction coefficient. A negative value is treated as a Manning's roughness coefficient.

• coriolis

Flags whether the Coriolis force is included in equations of motion (1) or not (0). To be used when the Coriolis force should be considered, such as for long-distance tsunamis that travel across the open ocean.

• c2p_all

Flags whether all velocities and all wave heights in a child domain are copied to the parent domain (1) or not (0). When these values are copied, data matches up between the domains.

• def_bathy

Flags whether terrain is deformed in accordance with crustal displacement (1) or not (0).

D. qsub.sh ... Queue submission script

This is a queue submission script. The sample data is specified for the JAMSTEC system; it should be modified as appropriate for system environment used.

## 5. Computing in the Serial Version (Non-MPI)

### 5-1. Compiling for the serial version

The serial version is compiled with the compile option of "MPI=OFF" or "#MPI=ON" in Makefile. "#" means comment out. OpenMP parallelization is available with in the serial version. To enable OpenMP, an option of compiler is mentioned, like as "-openmp"

[Makefile]

```
#MPI=ON                     ←Non-MPI／MPI switch : First "#" means comment out the line
```

### 5-2. Serial Version input files

Put the following files into the same directory.

| | | |
|---|---|---|
| Source code | JAGURS-D_V0xxx/ | |
| Terrain datafiles | bathy.SD01.grd<br>bathy.SD02.grd<br>bathy.SD03.grd<br>bathy.SD04grd<br>bathy.SD05.grd | |
| Crustal displacement datafiles | disp.SD01.grd<br>disp.SD02.grd<br>disp.SD03.grd<br>disp.SD04.grd<br>disp.SD05.grd | |
| WetOrDry file | wetordry.SD05.grd | (optional) |
| Parameter files | gridfile.dat | |
| | test_tgs.txt | |
| | tsun.par | |

### 5-3. Specifying the parameter files

Check the specification details of the parameter files and correct them as necessary.

- The parameter file defining terrain nesting (gridfile.dat)
- The parameter file defining waveform output points (test_tgs.txt)
- The computing parameter file (tsun.par)

### 5-4. Executing the Serial Version

After checking the specifications in the parameter file (tsun.par) and the terrain nesting file (gridfile.dat), run JAGURS with the following command.

```
$ ./JAGURS-D_V0xxx/jagurs par=tsun.par
```

## 5-5. Output files in serial version

Following files are generated in the current directory if the output file format option is default (#2).

| | | |
|---|---|---|
| Time history of water height and velocity at each waveform output point | tgs000001<br>~ tgsnnnnnn | • File format: text file (ASCII)<br>• The water heights and velocities at the respective output points are outputted in step order.<br>• The numbers in the filenames (000001, etc.) correspond to the station numbers in the waveform output points file (test_tgs.txt). |
| Initial water level in each domain | SD01.initl_disp.grd<br>SD02.initl_disp.grd<br>SD03.initl_disp.grd<br>SD04.initl_disp.grd<br>SD05.initl_disp.grd | • File format: GMT gridfile<br>• An initial water level is outputted for each domain. |
| Maximum water height in each domain | SD01.zmax.grd<br>SD02.zmax.grd<br>SD03.zmax.grd<br>SD04.zmax.grd<br>SD05.zmax.grd | • File format: GMT gridfile<br>• A maximum wave height is outputted for each domain. |
| Maximum velocity in each domain | SD01.vmax.grd<br>SD02.vmax.grd<br>SD03.vmax.grd<br>SD04.vmax.grd<br>SD05.vmax.grd | • File format: GMT gridfile<br>• A maximum velocity is outputted for each domain. |
| Water height and velocity for each domain and unit of time passed(#3) | SD01.00001200.grd<br>~ SD01.nnnnnnnn.grd<br>[...]<br>SD05.00001200.grd<br>~ SD05.nnnnnnnn.grd | • File format: GMT gridfile<br>• Water heights and velocities for each unit of time passed (snapshot output interval) are outputted for each domain.<br>• The numbers in the filenames (00001200, etc.) correspond to values set by the snapshot output interval (itmap) in the computing parameter file (tsun.par). |

(#2) The file output format is specified in the compile options (See 8-1. compile options).
(#3) Whether the velocities are outputted or not is specified in the computing parameter file (tsun.par).

# 6. Computing in the Parallel Version (MPI)

## 6-1. Compiling for the MPI parallel version

The parallel version is compiled with the compile option of "MPI=ON" in Makefile. OpenMP parallelization is simultaneously available in the parallel version. To enable OpenMP, an option of compiler is mentioned, like as "-openmp"

[Makefile]

```
MPI=ON              ←Non-MPI／MPI switch
```

## 6-2. Input files for parallel version

The input files of the serial version are used for input files of the parallel version.

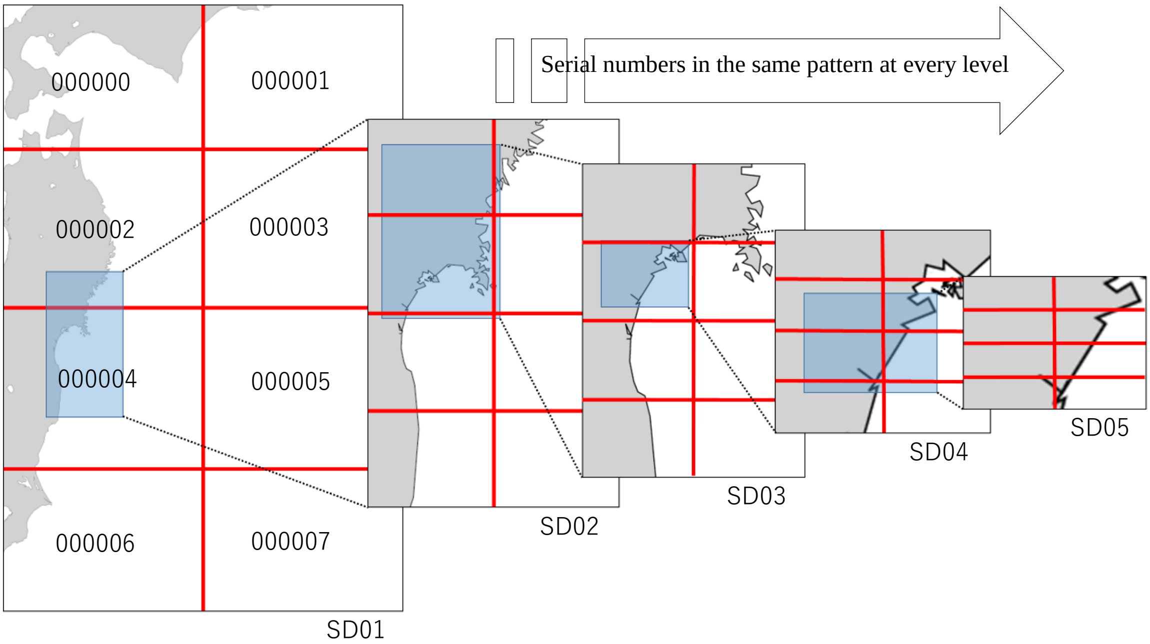
※In the previous JAGURS by V0407, each domain files for bathymetry and crustal displacement were divided into sub-domains corresponding to number of computational nodes. Since V0507, the same data set with that of serial version is available for the parallel version with a compile option (ONFILE = ON). Please see "8-1. Compile options" for detail.

## 6-3. Specifying the number of divisions (number of MPI parallelization)

In the parallel version, mention specifications of the numbers of divisions along x and y directions in the computing parameter file (tsun.par)

```
procx=2 ←Append: number of
divisions in X direction (east–west)
procy=4 ←Append: number of
divisions in Y direction (north–south)
```

For the example case, calculation domains are divided as follows



6-4. Executing the MPI parallel version

For the parallel computing using MPI, the computations are often submitted by a manner depends on the processing system. A job script should be prepared for each computer to be used. In the JAMSTEC system, hybrid jobs of OpenMP and MPI are available using the following job script.

```
#!/bin/bash
#BSUB -n 128
#BSUB -W 360
#BSUB -R rusage[mem=512]
#BSUB -a ICE
#BSUB -J jagurs
#BSUB -o stdout
#BSUB -e stderr

export OMP_NUM_THREADS=16
hybrid -mpi 8 -omp 16 -mpipn 1 "./JAGURS-D_V0xxx/jagurs par=tsun.par"
```

This example is a job script (qsub.sh) for a case in which 16 cores are run in parallel in each node by OpenMP and 8 nodes are run in parallel by MPI (the numbers of divisions of the gridfiles are 2×4).

In the JAMSTEC system, the job submit command is as follows.

```
$ bsub < qsub.sh
```

6-5. Output files for parallel version

The same set of output files with the serial version are generated by the MPI parallel computation (See 5-5. Output files in serial version)

# 7. Advanced calculations

## 7-1. Inputs

### A. Obtaining initial water heights by interpolating heights in the parent domains

If this function is enabled, initial water heights are obtained by interpolation from the parent domain.

[Files to be edited]

◆Computing parameter file (tsun.par)

```
init_disp_interpolation=1   ! If set to 1, initial water heights are obtained by interpolation
                                from the parent domain.
use_linear=1                ! If set to 1, linear interpolation is used. Otherwise, spline
                                interpolation is used.
```

◆ Parameter file defining terrain nesting (gridfile.dat)

■ If the fault parameter list is specified only for the root domain:

```
SD01 SD01 1  bathy.SD01.grd  disp.SD01.grd               ←Crustal displacement file
SD02 SD01 0  bathy.SD02.grd  NO_DISPLACEMENT_FILE_GIVEN  ←No disp file required; this
SD03 SD02 0  bathy.SD03.grd  NO_DISPLACEMENT_FILE_GIVEN  ←is given instead.
```

- Delimiter: space
- A list file (fault.list) including successive fault lists is specified here.

B. Considering rupture propagation in a fault

Rupture propagation in a fault can be simulated giving multiple crustal displacement files in time series. A list file describing the displacement file names is prepared. It is mentioned in the parameter file defining terrain nesting (gridfile.dat). The crustal displacement files are inputted one by one at intervals of the rise time (tau) with the rise time of “tau”.

[Files to be edited]

◆Computing parameter file (tsun.par)

```
multrupt=1              ! If set to 1, a multiple rupture model is enabled. Otherwise, the
                          multiple rupture model is disabled.
```

◆Parameter file defining terrain nesting (gridfile.dat)

```
SD01 SD01 1   bathy.SD01.grd   disp.SD01.list    ←Sets the name of the list file to replace the
SD02 SD01 0   bathy.SD02.grd   disp.SD02.list         crustal displacement file
SD03 SD02 0   bathy.SD03.grd   disp.SD03.list
SD04 SD03 0   bathy.SD04.grd   disp.SD04.list
SD05 SD04 0   bathy.SD05.grd   disp.SD05.list
```

• Delimiter: space

• Instead of crustal displacement files (disp.SDxx.grd), files (disp.SDxx.list) listing gridfile names to be successively read in are set.

◆If the list file (the list file specified in gridfile.dat) is disp.SD01.list:

```
./disp.multrupt/disp.SD01.00.grd    ←E.g., if tau=60, entered for 0–60 s
./disp.multrupt/disp.SD01.01.grd                 entered for 60–120 s    total duration:
./disp.multrupt/disp.SD01.02.grd                 entered for 120–180 s       180 s
```

• The crustal displacement files (including the locations (paths) of the files) that are actually successively read are described in this file.

• Each displacement file includes crustal deformation during each time span (tau), which is displacement in increment during the tau, not cumulative values since the rupture started.

• In this example, the crustal displacements are read at successive 60-second intervals, from disp.SD01.00.grd to disp.SD01.02.grd.

• In a case of computing in the Parallel Version,
the files actually read are not disp.SD01.00.grd—disp.SD01.02.grd
but must be divided data, disp.SD01.00.grd.000000—disp.SD01.02.grd.000000 and so on for the number of parallel divisions.

C. Inputting fault parameters and computing crustal displacement in JAGURS

If this function is enabled, the crustal displacement is not read from a crustal displacement datafile (disp.xxxx.grd). Instead, crustal displacement at the sea bottom is computed using a file that describes fault parameters. The crustal displacement is calculated by the method of Okada [1985, BSSA]. Initial water levels calculated in JAGURS are written to the file "[domain].initl_disp.grd".

[Files to be edited]

◆ Computing parameter file (tsun.par)

```
init_disp_fault=1                    ! If set to 1, the fault parameter reading function is ON.
                                       Otherwise, the function is OFF.
fault_param_file=”hoge.txt”          ! Specifies the name of a file describing fault parameters
```

◆ Fault parameter file (hoge.txt)

```
! lat[degree], lon[degree], depth[km], length[km], width[km],
!   ~ dip[degree], strike[degree], rake[degree], slip_amp[m]
40.1980 144.35 0.0 50 25 8.0 193.0 81.0 1.0
                    [...]
39.0000 143.00 0.0 50 25 8.0 193.0 81.0 1.0
```

←Lines starting with "!" are comments.

←One line, one fault (faults can be specified by arbitrary values).

←If the last line is blank, an error is issued.

- Delimiter: space
- The columns are defined as follows:

Column 1: Lat ... fault reference point latitude
Column 2: Lon ... fault reference point longitude
Column 3: Depth ... depth from surface (km)
Column 4: Length ... length of fault (km)
Column 5: Width ... width of fault (km)
Column 6: Dip ... angle (º)
Column 7: Strike ... direction (º)
Column 8: Rake ... slip angle (º)
Column 9: Slip ... amount of slip (m)

◆Parameter file defining terrain nesting (gridfile.dat)

```
SD01 SD01 1 bathy.SD01.grd  NO_DISPLACEMENT_FILE_GIVEN
SD02 SD01 0 bathy.SD02.grd  NO_DISPLACEMENT_FILE_GIVEN
                    [...]
```

←No disp file is required; this description is given instead.

- Delimiter: space

D. Effect of horizontal displacement of a seafloor slope

For a crustal displacement computed from fault parameters, the effects of tsunami excitation caused by horizontal displacement of a seafloor slope [Tanioka et al., 1996, GRL] are incorporated to produce initial water heights. (Note that this effect of horizontal displacement cannot be computed if the crustal displacement is read from a crustal displacement datafile "disp.xxxx.grd".)

[Files to be edited]

◆Computing parameter file ( tsun.par)

| | |
|---|---|
| hzdisp_effect=1 | ! If set to 1, horizontal displacement contributions are incorporated in initial wave heights. |
| min_depth_hde=100.0d0 | ! Sets a water depth (m) below which horizontal displacements are taken into account |

・min_depth_hde

Default value: 50 m

Horizontal displacement is considered only in grid cells deeper than the water depths of “min_depth_hde”.

E. Kajiura filter

If crustal displacements include short-wavelength patterns, a Kajiura filter [Kajiura, 1963, Bull. Earthq. Res. Inst.] may be applied. The Kajiura filter can be applied to initial water heights computed from fault parameters or to initial water heights provided in a "disp" file.

[Files to be edited]

◆ Computing parameter file (tsun.par)

| | |
|---|---|
| apply_kj_filter=1 | ! If set to 1, Kajiura filter is applied to initial wave heights |

Computing is not performed where the distance from the wave origin is more than 30° (in latitude-longitude).

F. Simultaneous considering of A, B, C, D and E

Here, we consider simultaneously the following A-C.

A. Obtaining initial water heights by interpolation of the parent domain data (init_disp_interpolation)

B. Taking rupture propagation in faults into account (multrupt)

C. Inputting fault parameters and computing crustal displacement within JAGURS (init_disp_fault)

[Files to be edited]

◆ Computing parameter file (tsun.par)

```
init_disp_interpolation=1    ! If set to 1, initial water heights are obtained by
                             ! interpolation from the parent domain.
multrupt=1                   ! If set to 1, the multiple rupture model is enabled.
init_disp_fault=1            ! If set to 1, the fault parameter reading function is ON.
```

◆ Parameter file defining terrain nesting (gridfile.dat)

■ If the fault parameter list is specified only for the root domain:

```
SD01 SD01 1  bathy.SD01.grd  fault.list                  ←Fault parameter list
SD02 SD01 0  bathy.SD02.grd  NO_DISPLACEMENT_FILE_GIVEN
SD03 SD02 0  bathy.SD03.grd  NO_DISPLACEMENT_FILE_GIVEN
```

- Delimiter: space
- A list file (fault.list) including successive fault lists is specified here.

◆ List file (a list file specified in gridfile.dat): fault.list

```
fault.0-30sec.txt     ←E.g., if tau=30, entered for 0–30 s  ┐
fault.30-60sec.txt                       entered for 30–60 s ├ total duration: 90 s
fault.60-90sec.txt                       entered for 60–90 s ┘
```

- The fault parameter files (including the locations (paths) of the files) that are actually successively read are described in this file.
- In this example, the fault parameters are read at successive 30-second intervals, from fault.0-30sec.txt—fault.60-90sec.txt.
- Each fault file includes fault slips during each time interval (tau), which is increment during "tau", not cumulative values since the rupture started.

◆ Fault parameter file (a fault parameter file specified in fault.list): fault.0-30sec.txt

```
! lat[deg.] long[deg.] depth[km] length[km] width[km] dip[deg.] strike[deg.] rake[deg.] slip_amp[m]
40.19800 144.35000 0.0 50.0 25.0 8.0 193.0 81.0 0.000000
39.73800 144.33100 0.0 50.0 25.0 8.0 193.0 81.0 0.000000
          [...]
36.20372 140.90455 26.0 50.0 50.0 16.0 193.0 81.0 0.000000
```

- See "B. Inputting fault parameters and computing crustal displacement in JAGURS" for a description of the format of a fault parameter file.

Further, options of the horizontal displacement effect (D) and Kajiura filter (E) can be enable simultaneously with A, B and C.

[Files to be edited]

◆Computing parameter file ( tsun.par)

```
hzdisp_effect=1        ! If set to 1, horizontal displacement contributions are
                         incorporated in initial wave heights.
apply_kj_filter=1      !   If set to 1, Kajiura filter is applied to initial wave heights
```

G. <u>Obtaining initial crustal displacement in a Gaussian distribution</u>

If this function is enabled, instead of the initial crustal displacement being read in from a crustal displacement datafile (disp.xxxx.grd), values specified in a Gaussian distribution specification file ("gaussian") are provided. The Gaussian distribution specification file "gaussian" can set central coordinates (latitude and longitude) and a width L (km).

[Files to be edited]

◆ Computing parameter file (tsun.par)

```
init_disp_gaussian =1     ! If set to 1, the Gaussian distribution provision function is ON.
                            Otherwise, the function is OFF.
```

◆ Gaussian distribution specification parameter file (gaussian); the filename is fixed

```
&gaussian
  lon_o=141.300201d0         ! Center lon.[Degrees]   Central coordinates of
                                                      Gaussian distribution
  lat_o=38.337348d0          ! Center lat.[Degrees]
  L=80.0d0                   ! Width[km]      Width of Gaussian distribution
/
```

• This file is prepared only if specified in a wave origin specification parameter file "gaussian".

◆ Parameter file defining terrain nesting (gridfile.dat)

```
SD01 SD01 1 bathy.SD01.grd  NO_DISPLACEMENT_FILE_GIVEN   ←No disp file is required;
SD02 SD01 0 bathy.SD02.grd  NO_DISPLACEMENT_FILE_GIVEN    this description is given
                                                          instead.
                 [...]
```

• Delimiter: space

H. Inputting a sinusoidal wave at a boundary face

If this function is enabled, instead of the initial crustal displacement being read in from a crustal displacement datafile (disp.xxxx.grd), a wave is generated along a line specified in a wave origin specification file (sinwave) for the duration of the rise time (tau). The wave origin specification file "sinwave" can set a period T (s), a wave height a (m), and a location (x,y) at which the sinusoidal wave is inputted. Also, any waveform can be generated by preparing a water height datafile.

[Files to be edited

◆ Computing parameter file (tsun.par)

```
init_disp_sinwave=1      ! If set to 1, the sinusoidal wave origin application function
                           is ON. Otherwise, the function is OFF.
                         ! (This function takes precedence over init_disp_gaussian.)
```

◆ Wave origin specification parameter file (sinwave); the filename is fixed

```
&sinwave
  T=20.0d0                ! Period (s)
  A=0.03d0                ! Wave height (m)
  x_index=11              ! Location (grid cell) where sinusoidal wave is inputted
  input_file='hoge.dat'   ! Wave height datafile for intervals dt
/
```

・T and A

If a wave height datafile (input_file) is specified, T and A are ignored.

・x_index / y_index

A Y coordinate may be specified in the manner x_index=0, y_index=11.

Take care not to set both x_index and y_index to non-zero values because the sine wave will be produced in a cross shape.

◆Water height datafile for intervals dt (hoge.dat); e.g., if dt=0.1, tau=300:

```
0.1 -0.000054        ←t: 0.1 (take care if not starting from time 0)
0.2 -0.000054        ←t:0.2
        [...]
299.9 -0.000162      ←t:299.9
300.0 -0.000162      ←t:300.0 (this exceeds the value of tau, so is not read in)
```

• Delimiter: space
• This file (hoge.dat) is read only if specified in the wave origin specification parameter file "sinwave".
• The data is read in sequence from the top in units of dt (steps) from the first step to time tau. (Note that there is a one-step offset if a line for time 0 is included.)
• If there is no data to be read, nothing is done.
• Only rows containing two real values are effective. Other data are discarded.

◆ Parameter file defining terrain nesting (gridfile.dat)

```
SD01 SD01 1 bathy.SD01.grd   NO_DISPLACEMENT_FILE_GIVEN
SD02 SD01 0 bathy.SD02.grd   NO_DISPLACEMENT_FILE_GIVEN
                [...]
```

←No disp file is required; this description is given instead.

• Delimiter: space

## 7-2. Computations

### A. Computing with Boussinesq dispersion

JAGURS can compute tsunami incorporating the Boussinesq-type dispersive term [Baba et al., 2015, PAGEOPH]. The dispersive terms are solved by an implicit method so that the duration of computing increases.

[Files to be edited]

◆ JAGURS Make file

```
CONV_CHECK=ON          ! If set to ON, a convergent computation check is enabled.
                         Otherwise, the check is disabled.
```

If enabled, the computing is terminated when it has converged or reached the number of times specified by “conv_val” or “max_step”, respectively, in computing parameter file (tsun.par).

If disabled, the computing is always repeated the “max_step”.

◆ Computing parameter file (tsun.par)

```
with_disp=1            ! If set to 1, a dispersive term is applied. Otherwise, it is OFF.
max_step=150           ! Maximum number of iteration (convergence) computations.
conv_val=1.0d-8        ! Error threshold (m/s) for termination at an iteration
                          (convergence) computation check
min_depth=10           ! Minimum water depth (m) for computing dispersion
```

・max_step

Default value: 9999

・conv_val

Default value: 1.0d-8

When a difference in velocities between iterative (convergent) computations falls below this value, the computing is judged to have converged and it proceeds to the next time step. (This term is only used if the compilation option CONV_CHECK is enabled.)

B. Absorbing boundary conditions

In the default setting, the tsunami wave is transmitting at the outer edges of the root domain. If absorbing boundary conditions are specified, the tsunami waves reaching the outer edges of the root domain are absorbed at those boundaries by the method of Cerjan et al. (1985).

[Files to be edited]

◆ Computing parameter file (tsun.par)

```
with_abc=1      ! If set to 1, the seaward side is an absorbing boundary.
nxa=20          ! Width (in grid cells) of an absorbing boundary in the east–west direction (X)
nya=20          ! Width (in grid cells) of an absorbing boundary in the north–south direction (Y)
apara=0.018     ! Strength of an absorbing boundary
```

・nxa / nxy

Default value: 20[#4]

For example, with 18-second grid cells, the default setting of the width of the absorbing boundary is 18 seconds x 20 cells.

・apara

Default value: 0.055d0 [#4]

This is a parameter representing the strength of attenuation in the root domain. The larger the value, the sharper the attenuation.

[#4] When the boundary width is wide and attenuation is gentler, computing tends to be more stable (nxa/nxy=60, apara=0.002d0 or similar).

C. Elastic loading and seawater compressibility

These options determine whether or not to include the effects of elastic deformation of the crust caused by the weight of a tsunami and the seawater compressibility in the equation of continuity according to the method of Allgeyer and Cummins [2014, GRL].

◆ Computing parameter file (tsun.par)

| (Parameters relating to elastic loading) | |
|---|---|
| with_elastic_loading=1 | ! If set to 1, elastic loading is ON. Otherwise, it is OFF. |
| m_radius=2.0d3 | ! Meaning the "Radius in which the loading effect will be estimated" |
| m_pyfile='PREM_Ggz.nc' | ! The name of a file in the NetCDF4 format providing the Green function |

・m_radius

Default value: 2.0d3

This sets a range for computing elastic deformation of the crust by a point load. (The default value is usually used.)

・m_pyfile

Default value: 'PREM_Ggz.nc'

A Green function representing elastic deformation of the crust by a point load is provided as a file in NetCDF4 format. The file with the default name is usually used.)

| (Parameters relating to the effect of seawater density) | |
|---|---|
| with_density=1 | ! If set to 1, seawater density effect is ON. Otherwise, it is OFF. |
| m_rho=1025.5d0 | ! Density at the sea surface ($kg/m^3$) |
| m_K=2.2d9 | ! Bulk modulus of seawater (Pa) |

・m_rho

Default value: 1025.5d0

This specifies the density at the sea surface. (The default value is usually used.)

・m_K

Default value: 2.2d9

This specifies the bulk modulus of seawater. (The default value is usually used.)

D. Executing multiple scenarios in a single job

This function executes multiple computations in a single job. Prepare a directory for each scenario and put all the files to be used in the computations in the directories. The directory names should be "input.[case ID (6 digits)]". (The directory names are fixed as "input.*".)

To execute the job, set the range of case IDs as an argument in the execution shell and perform execution with the MPI parallelization number being the same as the product of number of cases and MPI parallelization number of each case. Execute it on the directory “input.*” directories are put on.

The computation results are written to a directory for each case with the name "member.[case ID (6 digits)]".

[Files to be edited]

◆ JAGURS Make file

```
MULTI=ON          ! Enables the multiple scenario execution function
                  (disabled if set to OFF or not set)
```

◆ Directories for crustal displacement cases

Directory structure if 16 cases are prepared ("input.000001"—"input.000016")

```
JAGURS-D_Vxxx/jagurs
qsub.sh.noomp
input.000001/bathy.SD01.grd
             bathy.SD02.grd
                  [...]
             disp.SD01case01.grd
             disp.SD02case01.grd
                  [...]
             gridfile.dat
             tsun.par
             test_tgs.txt
    [...]
input.000016/bathy.SD01.grd
             bathy.SD02.grd
                  [...]
             disp.SD01case16.grd
             disp.SD02case16.grd
                  [...]
             gridfile.dat
             tsun.par
             test_tgs.txt
```

• In this example, the gridfiles in each pattern are not divided because of a serial run for each scenario. Gridfiles may be divided into sub-domain gridfiles as necessary (in which case the number of nodes described in the following execution shell should be modified appropriately).

◆ Execution shell example, for the JAMSTEC system

```
#!/bin/bash
#BSUB -n 16                                   ←16 cases; 16 parallel executions
#BSUB -W 2880
#BSUB -R "rusage[mem=1024] span[ptile=16]"
#BSUB -a ICE
#BSUB -J Tsunami-MPI
#BSUB -o stdout
#BSUB -e stderr

export OMP_NUM_THREADS=16

mpijob "./JAGURS-D_V0xxx/jagurs 1-16 tsun.par"
                                 ↑16 cases ("input.000001"–"input.000016")
```

Note that the execution procedure is a little different from the usual version (MULTI=OFF).
**Usual version: jagurs par=tsun.par**
**Multiple scenario version: jagurs 1-16 tsun.par**

**In all cases, the parameter file (tsun.par in the above example) must be identical. Even if parallel calculations are not performed in each case, it is necessary to specify procx=1 and procy=1. Parallelization using OpenMP is also possible in each case.**

(Example of 48-parallel calculation setting)

For 6 scenarios with procx=2, procy=4 (MPI 8-parallel) … mpirun -np 48

For 48 scenarios with procx=1, procy=1 (no MPI parallel) … mpirun -np 48

For 24 scenarios with procx=1, procy=1 (no MPI parallel, OpenMP 2-parallel) … mpirun -np 24

E. Restart computing

Computing may not be completed within a specified duration due to the run-time limit of a computer. If this happens, computed data can be saved in "restart data files" at restart points and the remaining computing can be resumed on the basis of this data.

[Files to be edited]

◆ Computing parameter file (tsun.par)

| (Parameters relating to restarting) | |
|---|---|
| restart_interval=12000 | ! Writing interval of restart data files (in steps) |
| restart =60000 | ! Step number of restart data files (steps) |
| max_time='23:40:00' | ! Computing termination time |

・restart_interval

Just before completion of the specified number of steps (the step count is divided up by the number of steps of the restart interval), a reset data file is written with the following name.

restart.[step number (8 digits)]

・restart

If this option is specified, the restart data file with the specified step number is read in and computing is resumed at the first step after the restart.

If a file with the same name as a file written during the previous computing is saved (a waveform output file tgs0000** or the like), that file will be overwritten at the time of the restart; so, take measures such as saving the files beforehand.

・max_time

If this option is specified, when the computing run-time reaches this value, time step processing is stopped and end processing is carried out including generation of restart files. Set a value that allows a reasonable margin (corresponding to one time step + end processing) relative to a run-time limit set at job execution.

F. Cartesian coordinates

JAGURS is capable of dealing with terrain in both spherical coordinates and Cartesian coordinates.

◆Makefile

```
CARTESIAN=ON          ! If set to ON, the Cartesian coordinates Version is compiled.
                        Otherwise, the spherical coordinates Version is compiled.
```

Handling of input data is the same as with the Spherical coordinates version. Prepare input data with x, y coordinates in meters not only bathymetric, displacement data but also tide station locations and fault parameters.

G. Spatially heterogeneous friction coefficient

JAGURS in default uses a spatially uniform coefficient of bottom friction with a parameter of “cf”. For a detail study of tsunami inundation process, different value of the coefficient of bottom friction can be defined on each calculation point. User prepare a GMT grid file including distribution of the friction coefficient for the same region and the same grid interval with the bathymetric grid. A positive value is treated as a dimensionless friction coefficient. A negative value is treated as a Manning's roughness coefficient. The GMT grid file of the friction coefficient is mentioned in the parameter file defining terrain nesting (gridfile.dat). When the GMT grid file for bottom friction is mentioned in “gridfile.dat”, the parameters of “cf” or “cfl” in “tsun.par” are ignored.

◆Parameter file defining terrain nesting (gridfile.dat)

```
SD01  SD01  1  bathy.SD01.grd  disp.SD01.grd
SD02  SD01  0  bathy.SD02.grd  disp.SD02.grd
SD03  SD02  0  bathy.SD03.grd  disp.SD03.grd
SD04  SD03  0  bathy.SD04.grd  disp.SD04.grd
SD05  SD04  0  bathy.SD05.grd  disp.SD05.grd  wetordry.SD05.grd sodo.SD05.grd
```

In the above example, “sodo.SD05.grd” is the GMT grid file including distribution of coefficient of bottom friction. The friction coefficient file is used only in the SD05 domain. The domains from SD01 to SD04 don’t use the friction coefficient files. So for SD01-SD04, the default coefficients declared in the parameters of “cf” and “cfl” are applied.

A term of “NO_FRICTION_FILE_GIVEN” is also available indicating no-use of the friction coefficient file. Similarly, a term of “NO_WETORDRY_FILE_GIVEN” is available indicating no-use of the wet or dry file.

H. Line barriers

To consider effects of seawalls and breakwaters on tsunami propagation, line barriers can be defined in JAGURS. The line barriers are located at calculation points of water flow velocity (discharge) . JAGURS applied an overflow formula proposed by Honma (1940) using width of line barriers of zero. This function is useful to consider the seawalls and breakwaters smaller than the grid space in FDM, and available in both spherical and Cartesian coordinates system with compile option “BANKFILE=ON”.

◆Makefile

```
BANKFILE=ON              ! Enable function of the line barriers
```

The line-barrier data are made in ASCII format for each domain, like as

◆Line barrier data（ir.SD05.dat)

```
-116105 -17035 1 3.30
-116105 -16715 3 3.12
-116115 -17035 1 3.30
-116115 -16715 1 3.12
-116125 -17045 1 3.30
-116125 -17035 2 3.30
```

• Delimiter: space
• The columns are defined as follows:

Column 1: Lat ... location point in latitude (deg.) or y (m)
Column 2: Lon ... location point in longitude (deg.) or x (m)
Column 3: ID ... direction of wall (1: to east, 2: to north, 3: to east and north)
Column 4: Height ... wall height (m)

Although the line barriers are located at calculation points of water flow velocity (discharge), the line barrier data indicates locations at calculation points of water height as shown in the example. This is because that JAGURS uses staggered grids between the flow velocity and water height. The first line in the example data

-116105 -17035 1 3.30

means, a line barrier of height of 3.3 (m) in T.P. (absolute elevation) to block the flow in east-west direction at the flow velocity calculation point to the east (1) of the water height point at y=-116105(m) and x=-17035 (m).

The line barrier data (ir.SD05.dat) are mentioned in the parameter file defining terrain nesting (gridfile.dat), like as

◆Parameter file defining terrain nesting (gridfile.dat)

```
SD01  SD01  1  bathy.SD01.grd  disp.SD01.grd
SD02  SD01  0  bathy.SD02.grd  disp.SD02.grd
SD03  SD02  0  bathy.SD03.grd  disp.SD03.grd
SD04  SD03  0  bathy.SD04.grd  disp.SD04.grd
SD05  SD04  0  bathy.SD05.grd  disp.SD05.grd  wetordry.SD05.grd sodo.SD05.grd ir.SD05.dat
```

If you don’t define the line barrier data, the column is filled by spaces. If user define the line barrier data only without using “wetordry” and “sodo” files. User should use “NO_WETORDRY_FILE _GIVEN” and “NO_FRICTION_FILE_GIVEN” to keep the location of column.

Tsunami overflow the line barrier when the water height is higher than the height of line barrier. At this timing, JAGURS can decrease the height of line barrier assuming collapse of the barrier by an option in the parameter file (tsun.par)

◆Computing parameter file (tsun.par)

```
broken_rate=0.4                ! Ratio to the original height
```

This example means that the height of the line barrier becomes to be 40 % to the original height after overflows. For no change after overflows, this option isn’t mentioned (comment out) or set negative value.

7-3. Outputs

A. Tsunami arrival times

JAGURS saves tsunami arrival times (seconds) by the same format (NetCDF) with the maximum tsunami height etc with an option as follows

◆Computing parameter file (tsun.par)

```
check_arrival_time=1              ! Enable to output tsunami arrival times (1:ON/0:OFF)
check_arrival_height=0.30d0       ! Threshold in water surface change to define
                                                    arrival of tsunami (m)
```

In the example, tsunami arrival times are saved at the time when the water change becomes larger than the check_arrival_height (0.3m) in the sea. For the land, tsunami arrival times are saved at the time when the flow depth becomes larger than the check_arrival_height (0.3m).

B. Calculating maximum flow depths

Differences can be calculated between [terrain height + crustal displacement] and maximum water height

= terrain height (measured vertically downward) – crustal displacement (measured vertically upward) + maximum water height (measured vertically upward)

To create a maximum flow depth file for domain SD05:

(1) Create a maximum flow depth gridfile

= [bathy.SD05.grd] – [disp.SD05.grd] + [SD05.zmax.grd]

```
$ grdmath bathy.SD05.grd disp.SD05.grd SUB SD05.zmax.grd ADD = SD05.fdmax.nc=nf -V
```

c. NetCDF format output (NCDIO)

If the "OUTPUT=NCDIO" option is set in the compilation options, Makefile, output files of wave height and velocity etc. are collected into a Single NetCDF file and written to the current directory.

◆Makefile

```
OUTPUT=NCDIO             ! One NCDIO file output
```

The command “nudump -h” reads the Single NetCDF file to get the statistical data. The grdreformat command can sample a particular time step in the Single NetCDF file including multiple data.

Procedures for extracting particular information in the GMT format (*.grd) from files in the NetCDF format (*.nc)
(The extracted file format is the nf format (#18).)

Example 1. Extracting maximum wave heights

Extraction source file name: nankai.2s.thk.nc
Extraction destination file name: nankai.2s.thk.max_height.grd
Extracted attribute name: max_height

```
$ grdreformat nankai.2s.thk.nc?max_height nankai.2s.thk.max_height.grd -V
```

Example 2. Extracting a wave height snapshot (in the initial time step)

Extraction source file name: nankai.2s.thk.nc
Extraction destination file name: nankai.2s.thk.wave_height_10s.grd
Extracted attribute name: wave_height[0]

```
$ grdreformat nankai.2s.thk.nc?wave_height[0] nankai.2s.thk.wave_height_10s.grd –V
```

("?", "[" and "]" require the escape character "¥")

Note: If information for subsequent time steps is to be extracted, the value of the index of wave_height[] must be changed. For example, extraction with wave_height[0] extract the first output. wave_height[1] is output for when the next output is performed after a specified interval has passed.
If snapshots are set to be produced at 10 second intervals, wave_height[0] is a snapshot after a cumulative duration of 0 seconds resulting in all data is zero and wave_height[1] is a snapshot after a cumulative duration of 10 seconds.

## 7-4. Inputs and Outputs

### A. File input/output in pixel format

The grid system for JAGURS' terrain data adopts a gridline format (storing coordinates and physical values at grid intersections). On the other hand, the grid system for terrain data in general tsunami numerical analysis is constructed in pixel format (storing coordinates and physical values at the center of the grid). With the method shown in this section, pixel format terrain data can be read directly.

(1) Settings at Make time

If you specify as follows during compilation,

```
PIXELIN=ON
```

terrain data files, crustal displacement data files, WetOrDry files, and friction coefficient files will be read in pixel format. Also, if you specify as follows during compilation,

```
PIXELOUT=ON
```

outputs the wave height and flow velocity at each time step, maximum wave height, maximum flow velocity, and tsunami arrival time data in pixel format. These can be specified individually.

Please note that the following make-time settings are required when using file input/output in pixel format:

```
CARTESIAN=ON   ! Specifies calculation in orthogonal coordinate system
ONEFILE=ON     ! Specifies that JAGURS will handle domain decomposition when MPI
               ! parallelization is enabled
```

(2) Execution Method

The program execution method is the same as when not using the pixel format, but the pixel format data does not include information such as "the number of grids in the east-west and north-south directions", "grid size", and "coordinates of the eastern, western, southern, and northern ends". Therefore, it is necessary to create a "descriptor file" to specify these. An example of a descriptor file is shown below. The name of the descriptor file will be the terrain data filename of the corresponding domain with ".desc" added. For example, the name of the descriptor file for the terrain data filename "foo.dat" will be "foo.dat.desc".

◆Descriptor file (foo.dat.desc)

```
&desc
west=487200.0d0,   ! [Required] Coordinate of the western end of the domain [m]
east=727800.0d0,   ! [Required] Coordinate of the eastern end of the domain [m]
south=371700.0d0,  ! [Required] Coordinate of the southern end of the domain [m]
north=430200.0d0,  ! [Required] Coordinate of the northern end of the domain [m]
dx=810.0d0,        ! [Required] Grid interval in the east-west direction [m]
dy=810.0d0,        ! [Required] Grid interval in the north-south direction [m]
zmin=3694.47d0,    ! [Optional] Minimum elevation value of the domain [m]
zmax=7231.11d0,    ! [Optional] Maximum elevation value of the domain [m]
nx=1500,           ! [Required] Number of grids in the east-west direction
ny=990,            ! [Required] Number of grids in the north-south direction
/
```

If the descriptor file does not exist, an error message 'Error! Descriptor file "foo.dat.desc" does not exist!' will be output to the standard error output, and the program will terminate. Also, if the required values west, east, south, north, dx, dy, nx, and ny are not specified or if an invalid value is specified, the message 'Error! Invalid description on descriptor file "foo.dat.desc"!!!' will be output to the standard error output, and the program will terminate. The maximum and minimum elevation values of the domain are not required to be specified. No matter what values are specified, they will not affect the calculation content.

(3) Points to Note

- In order to properly perform nested interconnection between multiple domains during execution, the number of grids in the east-west and north-south directions of each domain must be 3n+1 (n is an integer). Therefore, the number of grids in the east-west and north-south directions will be reduced by "excluding one grid at each end of the east-west, north, and south" during calculation execution. Note that the same grid reduction is applied to the root domain, which has no particular restrictions on grid shape, in order to unify the handling of each domain.
- The input data size must have a grid count of 3n+1 by "excluding one grid at each end of the east-west, north, and south". In other words, the number of grids must be 3n.
- If pixel format is specified as the output file format, the value output for the reduced grids as described above will be 0 (to keep the file input and output grid sizes the same). If the output file format is not pixel format, the output will be for the domain with the reduced grid shape as is.
- To properly perform nested interconnection, the "shifted" values for the coordinates at the ends of the east-west and north-south directions should be used, shifting them by 1.5 grid units.
- The results of the grid shape changes as described above will be output to the standard output file (in rank 0 when MPI serialization is enabled) in a format similar to the bolded part below.

◆An example of standard output file

```
[…]
./jagurs: Reading grid data...

  *** grid 1
[pixcel format] 1 grid on each edge is ignored on "depth_2430-01.dat"!
[pixcel format] Grid size is dealt as 718 x 538
[pixcel format]     instead of it's original gird size 720 x 540.
[pixcel format] West edge is changed from -730200.000 to -726555.000.
[pixcel format] East edge is changed from 1019400.000 to 1015755.000.
[pixcel format] South edge is changed from -857700.000 to -854055.0 00.
[pixcel format] North edge is changed from 454500.000 to 450855.000.

          read_bathymetry_gmt_grdhdr(newsub.o): file name=depth_2430-01.dat
          nx=718 ny=538
          dx=2430.000000 dy=2430.000000
          zmin=-3171.920 zmax=9797.570
          west=-726555.000 east=1015755.000 south=-854055.000 north=450855.000
          mlon0=-726555.000 mlat0=-854055.000 dxdy=2430.000000
```

```
            read_bathymetry_gmt_grd(newsubs.o):
            min=-3171.920 max=9797.570
            NO_FRICTION_FILE_GIVEN
            NO_WETORDRY _FILE_GIVEN

  *** grid 2
[pixcel format] 1 grid on each edge is ignored on "depth_0810-01.dat"!
[pixcel format] Grid size is dealt as 1498 x 988
[pixcel format]     instead of it's original gird size 1500 x 990.
[pixcel format] West edge is changed from -487200.000 to -485985.000.
[pixcel format] East edge is changed from 727800.000 to 726585.000.
[pixcel format] South edge is changed from -371700.000 to -370485.000.
[pixcel format] North edge is changed from 430200.000 to 428985.000.

            read_bathymetry_gmt_grdhdr(newsub.o): file name=depth_0810-01.dat
            nx=1498 ny=988
            dx=810.000000 dy=810.000000
            zmin=-3694.470 zmax=7231.110
            west=-485985.000 east=726585.000 south=-370485.000 north=428985.000
            mlon0=-485985.000 mlat0=-370485.000 dxdy=810.000000
            read_bathymetry_gmt_grd(newsubs.o):
            min=-3694.470 max=7231.110
            NO_FRICTION_FILE_GIVEN
            NO_WETORDRY_FILE_GIVEN
[...]
```

・ Even when pixel format is specified as the output file format, the undefined values will be output as 1.0d10. Please be cautious when handling this in post-processing.

(4) File format

・ Input file

When the input file is set to pixel format, the grid file that specifies the input filename (typically named gridfile.dat) will be in pixel format as well. Among these four input files, both the terrain data file and the WetOrDry file are in Fortran-formatted file "10f8.2" format (8-digit overall, 2-digit fractional part, and 10 elements per row), while the crustal displacement data file and the friction coefficient file are in Fortran-formatted file "10f8.4" format (8-digit overall, 4-digit fractional part, and 10 elements per row).

・ Output file

When the output file is set to pixel format, all grid data output will be in pixel format. The output data is in Fortran-formatted file "10e15.6" format (15-digit overall mantissa, 6-digit fractional part, and 10 elements per column), and undefined values are output as 1.0d10. Also, a value of 0 is output for the one grid at the eastern-western and northern-southern ends that are excluded from the calculation.

## 7-5. Atmospheric pressure wave generated from great volcanic eruptions and air-coupled tsunami calculations

### A. Overview

We use the formulations of Harkrider and Press (1967, GJI), Press and Harkrider (1962, GJI) and Harkrider (1964, JGR) to calculate the normal modes and corresponding atmospheric pressure waveforms assuming a given one-dimensional structure (atmosphere-ocean model; assuming a rigid seafloor of constant depth, and a free surface at the top of the atmosphere). Subsequently, the calculated atmospheric pressure waves are inputted into JAGURS for air-coupled tsunami calculations.

### B. Workflow

Calculations are carried out in the following flow:

1. Preparation of the normal mode calculation program (refer to C-(1))
2. Calculation of dispersion curves by the normal mode calculation program (frequency domain; refer to C-(2))
3. Calculation of waveforms by the normal mode calculation program (time domain; one-dimensional; refer to C-(3))
4. Calculation of tsunamis by JAGURS using the computed atmospheric waveforms as input (JAGURS solver; two-dimensional; refer to C-(4))

### C. How to execute the normal mode calculations

(1) Preparations

・ Fortran version

In "meteotsunami/normalmode/fortran", you create the executable files of PHcalc_wave6 and the dump tool printp (refer to (5)) by using the make command. First, you download the Bessel function library amos from the internet, and make it by following procedure

1. Access the Netlib webpage at https://www.netlib.org/amos/, and download the source code for "zbesh.f" and related files. Please click on the "plus dependences" section.
2. Select "Tar-gzip (tgz)" as the compressed format, and you will get "netlibfiles.tgz" which includes the entire related source code. Expand this in "meteotsunami/normalmode/fortran", and you will get the "amos" directory.
3. The necessary source code modifications can be applied with "patch.amos". Run "patch < ../patch.ams" under the "amos" directory.

The subsequent steps for make are as follows.

```
$ cd meteotsunami/normalmode/fortran/amos
$ make # Make libamos.so
$ cd ../ # Modify Makefile if needed
$ make # Executables PHcalc_wave6 and print are made
```

Next, you will execute "PHcalc_wave6" with specifying the necessary runtime parameters as appropriate in the Fortran namelist file named normalmode.namelist (as shown below). Please refer to (5) for the parameters to be specified and those default values.

```
&normalmode
srclon=138.72d0 ! Longitude of the atmospheric pressure source [degree]
srclat=35.35d0  ! Latitude of the atmospheric pressure source [degree]
dist_s=0        ! Starting point of the distance from the source [km]
dist_e=2000     ! Ending point of the distance from the source [km]
tmax=3600.0d0   ! Computing time [s]
dt=1.0d1        ! Time interval for computation[s]
sind=10         ! Altitude of the atmospheric pressure source [bottom of the layer number n]
minmode=10      ! Minimum value of the target mode
maxmode=13      ! Maximum value of the target mode
Samp=1000.0d0   ! Amplitude of the atmospheric source [m]
SDt=2000.0d0    ! Period [s]
DCflag=.true.   ! Specify whether to compute the dispersion curves
SYNflag=.true.  ! Specify whether to compute the atmospheric waveforms
OH=5.0d0        ! Water depth [km]
atmfile='AtmModel.dat' ! Atmosphere structure file name
/
```

・ Python version

In "meteotsunami/normalmode/python", run it in a Python3 environment. Install the NumPy, SciPy, Pandas packages, the json module, and any packages they depend on.

In the following steps, execute "PHcalc_wave6.py with" specifying the necessary runtime parameters as appropriate in the JSON file "normalmode.json" (as shown below).

Please refer to (5) for the parameters to be specified here and the default values when not specified.

```
{
"srclon":138.72, ! Longitude of the atmospheric source [degree]
"srclat":35.35,  ! Latitude of the atmospheric source [degree]
"dist_s":0,      ! Starting point of the distance from the target source [km]
"dist_e":2000,   ! Ending point of the distance from the target source [km]
"tmax":3600,     ! Computing time [s]
"dt":10,         ! Time interval for computation[s]
"sind":10,       ! Altitude of the atmospheric source [bottom of the layer number n]
"minmode":10,    ! Minimum value of the target mode
"maxmode":13,    ! Maximum value of the target mode
"Samp":1000,     ! Amplitude of the atmospheric source [m]
"SDt":2000,      ! Period [s]
"DCflag":1,      ! Specify whether to compute the dispersion curves
"SYNflag":1,     ! Specify whether to compute the atmospheric waveforms
"OH":5,          ! Water depth [km]
"atmfile":"AtmModel.dat" ! Atmosphere structure file name
}
```

(2) Calculation of Dispersion Curves

Required parameters: minmode, maxmode, OH, atmospheric structure file (default file name is "AtmModel.dat")

The computation of normal modes assumes a constant water depth (OH), so you can use the average value of the great circle path connecting the wave source and the observation point.

The program calculates all the normal modes within the specified phase speed range and output the dispersion curves corresponding to from "minmode" to "maxmode". You would require trials and errors to find optimal parameters for your problem, so you may want to calculate only the dispersion curves without the waveforms in the beginning (Specify "DCflag=.true." and "SYNflag=.false." for the Fortran version, and '"DCflag":1' and '"SYNflag":0' for the Python version).

Please note that the computed normal modes are simply in order, which include acoustic mode and tsunami mode. The calculated modes align by GR11, GR10, GR9, ..., GR4, GW (tsunami), GR3, GR2, GR1, GR0 (Lamb wave), S0, S1, ...). The modes are numbered sequentially from GR11, so if you want to calculate the first higher order mode (GR1) and the basic mode of atmospheric gravity waves (GR0), you set "minmode=12", "maxmode=13". You can draw a graph of the calculated dispersion curves using the output file "DCPH5_[minmode]-[maxmode].txt" that includes the mode (mode number), kroot (wave number), vpkr (phase velocity), and omgkr (angular frequency). The dispersion curve is the relation between the frequency or period and phase velocity of each mode.

If you want to calculate only a single mode, specify the same number for minmode and maxmode.

Please make sure to obtain the dispersion curves correctly that you want before the next step.

(3) Calculation of Atmospheric Pressure Waveform

Required parameters: Use all parameters including those in the previous section

Based on the dispersion curves obtained in the previous section, the program generates the atmospheric pressure waveform at the specified distance. The atmospheric waveform is a function of time and distance from the atmospheric source.

The atmospheric source is placed at the bottom of the layer in the atmospheric and oceanic model. Assuming that the first layer is the ocean and the second and subsequent layers are the atmospheric layers, setting "sind=2" defines the source on the sea surface (bottom of the second layer).

The atmospheric pressure source is approximated by a sine function for one cycle according to equation (13) of Harkrider and Press (1967) (the amplitude is "Samp", the period is "SDt"). You probably need to adjust "Samp" and "SDt" to generate a pressure waveform what you require at the observation point.

“normalmode.dat” of computation results is formatted in binary. You can restore the pressure waveform at the specified location to ascii format using printp (refer to (5)).

Please note that at a location where the distance from the atmospheric source is 0, the calculation will not be performed because it diverges, and the value at distance=1 will be copied (i.e., there will be a change in atmospheric pressure at the source during tsunami calculation).

Because the calculated dispersion curves in the frequency domain are returned to the time domain by Fourier transformation, the waveform is looped on the time domains. Therefore, it is recommended to set substantially long time range.

The output file of normalmode.dat and "nomalmode.namelist" are input to JAGURS, so copy them to the JAGURS execution directory. Please note that "nomalmode.namelist" is the input to the normal mode calculation program in the Fortran version, and in the Python version this is the output from the normal mode output program.

(4) Tsunami Calculation

You specify NORMALMODE=ON in the Makefile during the make of the JAGURS, so that JAGURS can calculate tsunami using the atmospheric pressure waves obtained in the previous step according to the distance from the source to each grid. You can find the air pressure fluctuations at the observation points in TGS files. Adding a specification of "dumpp=1" to the run-time configuration file (tsun.par) creates snap shots of air-pressure spatial distribution in Pa of "[domain name]-P.[step].grd" at the same timing as the output of the wave height snapshot files.

(5) Appendix

・Execution Parameters

The parameters specified in the runtime configuration file (in the Fortran version, the Fortran namelist file "normalmode.namelist"; in the Python version, the JSON file "normalmode.json") and the default values when they are unspecified are shown in the table below.

| Parameter | Comment | Default Value |
|---|---|---|
| srclon | Longitude of the atmospheric source [degree] | N/A |
| srclat | Latitude of the atmospheric source [degree] | N/A |
| dist_s | Starting point of the distance from the target source [km] | N/A |
| dist_e | Ending point of the distance from the target source [km] | N/A |
| tmax | Computing time [s] | N/A |
| dt | Time interval for computation[s] | N/A |
| sind | Altitude of the atmospheric pressure source [bottom of the layer number *n*] | 2 |
| minmode | Minimum value of the target mode | 3 |
| maxmode | Maximum value of the target mode | 5 |
| Samp | Amplitude of the atmospheric pressure source [m] | 1,000 |
| SDt | Period [s] | 2,000 |
| DCflag | Specify whether to compute the dispersion curve | .true./1 |
| SYNflag | Specify whether to compute the atmospheric pressure waveform | .true./1 |
| OH | Water depth [km] | 5 |
| atmfile | Atmosphere structure file name | AtmModel.dat |

・Atmospheric Structure Model

The atmospheric structural model up to an altitude of 220 km (AtmModel.dat) included in the package was a digitized version of the U.S. Standard Atmosphere. Each row designates parameters for each respective layer, with

1st column: bottom altitude of the layer (with sea level being 0) [km]
2nd column: thickness of the layer [km]
3rd column: speed of sound [km/sec]
4th column: density [kg/m3]

For reference, the original data for the U.S. Standard Atmosphere can be found in Campen, C.F., Cole, A.E. & Condron, T.P., 1960. Model atmospheres, in Handbook of Geophysics, revised edition, pp. 1.1-1.43, The Macmillan Company.

・How to Use the Dump Tool printp

"printp" dumps the values of pressure at specified coordinates from the output file normalmode.dat. "printp" is created along with the executable file "PHcalc_wave6" when making the Fortran version. You specify the longitude and latitude of the observation point in the Fortran namelist file "printp.namelist".

```
&params
distlon=140.86d0 ! Longitude of the observation point
distlat=38.26d0   ! Latitude of the observation point
/
```

Then, execute "printp". As shown in below, the values of pressure [Pa] at the specified coordinates from 0 to tmax [seconds] are output.

```
- Normalmode parameters in "normalmode.namelist"
srclon, srclat:          138.720          35.350
dist_s, dist_e:                0         2000
tmax, dt:               3600.000          10.000

- Reading "normalmode.dat"

- Observation point is specified in "printp.namelist"
distlon, distlat:          140.860          38.260
nm_ind:        375

------------------------------
Time[s]            P
------------------------------
            0.000       -0.666E+03
[...]
         3600.000       -0.872E+03
------------------------------
```

# 8. Parameter summary

## 8-1. Compile options

| Item | Value | Comment |
|---|---|---|
| PREC | REAL_DBLE | All calculations are performed with double precision. |
| | DBLE_MATH | Only intrinsic math functions are calculated with double precision. |
| | Other | All calculations are performed with single precision. |
| MPI | ON | MPI enabled (Parallel Version) |
| | Other | MPI disabled (Serial Version) |
| USE_ALLTOALLV | ON | The MPI_Alltoallv function is used for inter-nesting communications. |
| | Other | The MPI_Allreduce function is used for inter-nesting communications. |
| A2A3D | ON | The A2A3D function (Baba et al, 2015, High Per. Comp. App.) is used for inter-nesting communications.(This option is valid only if USE_ALLTOALLV=ON) |
| SINGLE_A2A | ON | The SINGLE_A2A function (Baba et al, 2015, High Per. Comp. App.) is used for inter-nesting communications.(This option is valid only if USE_ALLTOALLV=ON) |
| TIMER_DETAIL | ON | Enable timer output |
| | DETAIL | Enable more detailed timer output |
| | Other | Disable timer output |
| CONV_CHECK | ON | Enable convergent computation check<br>i.e., if the change in flow rate in a convergent calculation with a dispersive term falls below the threshold (the value specified by conv_val in the computing parameter file), the program proceeds to the next time step. |
| | Other | Disable convergent computation check (i.e., perform a set number of iterations) |
| OUTPUT | NCDIO | All water height and velocity output files are collected into a single NetCDF file for each node and written to the program execution directory.<br>• File format: NetCDF<br>• Filename: "[domain].[rank (6 digits, padded with zeros)].nc" |
| | DIROUT | Water height and velocity output files are gathered into a directory for each step and then written to the program execution directory.<br>• File format: GMT grid<br>• Filename: "[step].grd" / "[domain].[step number].grd.[rank number]" |
| | Other | All output files are written to the program execution directory.<br>• File format: GMT gird<br>• Filename: "[domain].[step number].grd.[rank number]" |

## 8-1. Compile options (cont.)

| Item | Value | Comment |
|---|---|---|
| SINGLE_TGS | ON | Water height and velocity at each waveform output point (station) are written into a single file per process.<br>• Output filename: "tgs_station.[rank number]" (written for each process)<br>• Information on multiple stations is combined and written to each output file. (Because "[station number]" is appended to the start of each line, the output of each station can be identified.)<br>• The attribution script "splittgs.sh" can be used to recreate the previous output format (one file for each station). |
| | Other | Water height and velocity at each waveform output point (station) are written into a single file per station.<br>• Output filename: "tgs[station (6 digits)]" |
| MULTI | ON | Execute multiple scenarios in a single job |
| | Other | Execute single scenario in a single job |
| SKIP_MAX_VEL | ON | Skip computation and output of max velocity to save time. |
| | Other | Perform computation and output of max velocity. |
| LESS_CC | ON | Perform convergence check on dispersive calculation only every 10 steps to save time. |
| | Other | Perform convergence check on dispersive calculation every 10 step. |
| CARTESIAN | ON | Use Cartesian coordinate system (y, x in meter) |
| | Other | Use spherical coordinate system (lat, lon in degrees) |
| UPWIND3 | ON | Use 3$^{rd}$-order upwind difference to solve advection term |
| | Other | Use 1$^{st}$-order upwind difference to solve advection term |
| BANKFILE | ON | Enable line barriers |
| | Other | Unable line barriers |
| HZMINOUT | ON | Output the minimum (max. in minus) tsunami height distribution |
| | Other | Not output the minimum (max. in minus) tsunami height distribution |
| ONEFILE | ON | Domain decompositions for MPI calculation are automatically done in JAGURS. Users don't need to split the input files according to the number of computational nodes before calculation with this option "ONEFILE=ON". This is new function since V0500. This manual assumes cases "ONEFILE=ON". |
| | Other | Users need to split the input files according to the number of computational nodes before calculation. With "ONEFILE=OFF", please refer the previous manual by V0400. |

## 8-1. Compile options (cont.)

| Item | Value | Comment |
| --- | --- | --- |
| PIXELIN | ON | Read terrain data files, crustal displacement data files, WetOrDry files, and friction coefficient files in pixel format. This is a feature from V0520. |
| | Others | Read terrain data files, crustal desplacement data files, WetOrDry files, and friction coefficient files in GMT grid format. |
| PIXELOUT | ON | Output wave height and flow velocity at each elapsed time, maximum wave height, maximum flow velocity, tsunami arrival time data, etc. in pixel format. This is a feature from V0520. |
| | Others | Output wave height and flow velocity at each elapsed time, maximum wave height, maximum flow velocity, tsunami arrival time data, etc. in GMT grid format. |
| OLD_SCHEME | ON | Perform the calculation of the equations of motion using the conventional method (prior to V0516). |
| | Others | Use Minami et al. (2023)'s friction term calculation method in the calculation of the equations of motion. |
| NORMALMODE | ON | Enable the reading of the normal mode input file. |
| | Others | Disable the reading of the normal mode input file. |

## 8-2. Parameter List for running JAGURS (tsun.par)

(∗:required items)

| Item | | Comment | Default Value |
|---|---|---|---|
| (Filename setting) | | | |
| gridfile | ∗ | Parameter file for nesting layer of gridfile (ex:gridfile.dat) | |
| maxgrdfn | | Output file for max water height (ex:zmax.grd) | maxgrdfn='zmax' |
| mingrdfn | | Output file for min water height (ex:zmax.grd) | mingrdfn='zmin' |
| vmaxgrdfn | | Output file for max velocity (ex:vmax.grd) | vmaxgrdfn='vmax' |
| tgstafn | ∗ | Name of parameter file specifying waveform output points (e.g., test_tgs.txt) | |
| tgstxtoutfile | | Heading characters of tsunami waveform files | tgstxtoutfile='tgs' |
| (Model Parameters in Basic) | | | |
| dt | ∗ | Time step [seconds] | |
| tend | ∗ | End time [seconds] | |
| itmap | ∗ | Snapshot interval [steps] (dt * itmap = dump interval) | |
| itmap_start | | ├ From [step] | itmap_start=1 |
| itmap_end | | └ To [step] | itmap_end=99999999 |
| tau | ∗ | Time step between ruptures [seconds] | |
| cf | ∗ | For sea — (positive value): Non-dimensional coefficient<br>(negative value): Manning's roughness coefficient | (If cfl is undefined, it will be overwritten by cf) |
| cfl | ∗ | For land | |
| froude_lim | | Limiter with max Froude number | froude_lim=2.0d0 |
| coriolis | ∗ | Coriolis force (1:ON, 0:OFF) | |
| c2p_all | | All Grids are copied by to coarse (1:ON, 0:OFF) | c2p_all=0 |
| nest_1way | | Only c2p interpolation is performed<br>(in other words, c2p copy is NOT performed) | nest_1way=0 |
| def_bathy | | Deform bathymetry based on crustal displacement (1:ON, 0:OFF) | def_bathy=1 |
| plotgrd | | Grid number (counting from 1) to output as a grd file<br>・plotgrd is <0 or > n-Grid: all of them are output<br>・Else: Only the domain which you choose are output<br>(ex:plotgrd=2,5 ... domain 2 & 5 are output<br>plotgrd=-1 ··· all domain are output) | plotgrd=-1 |
| velgrd | | Outputting velocity(x, y) gridfiles (1:ON, 0:OFF) | velgrd=1 |
| speedgrd | | Outputting velocity (absolute value) gridfiles(1:ON, 0:OFF)<br>(you can set this option separately from [velgrd]) | speedgrd=0 |
| start_date | | Simulation start time (for output with the NetCDF format) | start_date='2000-01-01 00:00:00' |
| itgrn | | Frequency of tide gauges (TGs) data output. TGs data is dumped every itgrn steps. | itgrn=1 |

## 8-2. Parameter List (cont.)（＊:required items）

| Item | | Comment | Default Value |
|---|---|---|---|
| (Initial sea surface displacement) | | | |
| multrupt | | Multiple ruptures (1:ON, 0:OFF) | multrupt=0 |
| init_disp_interpolation | | Initial displacement is calculated/read only for root domain and ones for child domains are given by interpolation (1:ON, 0:OFF) | init_disp_interpolation=0 |
| use_linear | | If init_disp_interpolation=1, you can choose interpolation method (1:linear interpolation, 0:3rd-order spline interpolation) | use_linear=0 |
| init_disp_fault | | Initial displacement with fault parameters (1:ON, 0:OFF) | init_disp_fault=0 |
| fault_param_file | | Name of a file describing fault parameters | fault_param_file="fault" |
| init_disp_gaussian | | Initial displacement with Gaussian distribution (1:ON, 0:OFF)<br>Parameters for Gaussian distribution are specified by the file named "gaussian" | init_disp_gaussian=0 |
| init_disp_sinwave | | Initial displacement with sin wave (1:ON, 0:OFF)<br>Parameters for sin wave distribution are specified by the file named "sinwave" | init_disp_sinwave=0 |
| (Dispersive wave) | | | |
| with_disp | | Dispersive (0:OFF, 1:ON, 2:ON but except root domain) | with_disp=0 |
| max_step | | Maximum steps of iterations for convergence | max_step=9999 |
| conv_val | | Truncation error [m/s]. This is available with a compile option of "CONV_CHECK" | conv_val=1.0d-8 |
| min_depth | | Minimum depth of sea [m] computing the dispersion | min_depth=5.0d0 |
| (Absorbing boundary condition) | | | |
| with_abc | | Absorbing boundary condition (1:ON, 0:OFF (transmission boundary)) | with_abc=0 |
| nxa | | Number of target Grids on East/West boundary | nxa=20 |
| nya | | Number of target Grids on North/South boundary | nya=20 |
| apara | | Absorbing parameter (that is amplitude) | apara=0.055d0 |
| (Horizontal movement of seafloor slope) | | | |
| hzdisp_effect | | Adopt horizontal displacement effect (1:ON, 0:OFF) | hzdisp_effect=0 |
| min_depth_hde | | Lower limit of depth to adopt horizontal displacement effect [m] | min_depth_hde=50.0d0 |
| (Kajiura filter) | | | |
| apply_kj_filter | | Apply Kajiura filter (1:ON, 0:OFF) | apply_kj_filter=0 |
| (Elastic Loading) | | | |
| with_elastic_loading | | Elastic loading (1:ON, 0:OFF) | with_elastic_loading=0 |
| m_radius | | Radius (km) in which the loading effect will be estimated | m_radius=2000.0d0 |
| m_pyfile | | NetCDF file to specify a Green's function | m_pyfile='PREM_Ggz.nc' |

8-2. Parameter List (cont.) (＊:required items)

| Item | | Comment | Default Value |
|---|---|---|---|
| (Seawater Compressibility) | | | |
| with_density | | Density (1:ON, 0:OFF) | with_density=0 |
| m_rho | | Water density at sea surface [kg/m$^3$] | m_rho=1025.5d0 |
| m_K | | Bulk modulus of seawater [Pa] | m_K=2.2d9 |
| (Restart) | | | |
| Restart | | Restart step [steps] (1:ON, 0:OFF) | restart=0 |
| restart_interval | | Restart file interval [steps] (0:OFF) | restart_interval=0 |
| max_time | | Force-quit time after end processing (ex: max_time='23:40:00') | |
| (MPI parallelization) | | | |
| procx | ＊ | East-West direction (procx) | |
| procy | ＊ | North-South direction (procy) | |
| (Arrival time) | | | |
| check_arrival_time | | Output arrival time (1:ON, 0:OFF) | check_arrival_time=0 |
| check_arrivel_height | | Threshold in water surface change to define tsunami arrival | check_arrivel_height= 0.01d0 |
| (Line barriers) | | | |
| broken_rate | | Ratio to the original height after overflow. For no change after overflows, this option isn't mentioned (comment out) or set negative value (default). | broken_rate=-1.0 |
| (Normalmode) | | | |
| dumpp | | Output the values of pressure [Pa] at each grid point at the same timing as the output of the flow velocity snapshot file. | dumpp=0 |

## 9. Afterword

The program was developed and improved through collaboration between research teams in Japan and Australia. We made the source code open in the spirit of sharing the fruits of its development with the whole tsunami research community for the benefit of society. We would also like to hear about any bug reports or other issues arising in the software's use. If you add new functions, please let us know so we can improve the functionality even further.

We appreciate for your cooperation developing JAGURS.

Toshitaka Baba
Tokushima University/
baba.toshi@tokushima-u.ac.jp

Phil R. Cummins
Australian National University/
Geoscience Australia
phil.cummins@anu.edu.au

## 10. Version history

| Version | Date | History of modification |
|---|---|---|
| ver. 2014.03.11 | 2014/03/11 | First publication |
| ver. 2014.06.03 | 2014/06/03 | • Corrected errors and omissions<br>• Other:<br>p02: 2-1. Version change with JAGURS-D_V0177 release<br>p03: 2-5. Added description<br>p04: 2-6-2. Added description<br>p05: 2-6-3. Changed sample parameter value<br>p06: 2-7. Added file description<br>p07: 3-2. Added description<br>p10: 4-2. Added description and changed sample parameter value<br>p11: 4-3. Added description<br>p19: 7-2. Added description of JAGURS-D_V0177 release |
| ver. 2014.07.14 | 2014/07/14 | • Corrected errors and omissions<br>• Other:<br>p02: 2-1. Version change with JAGURS-D_V0178 release<br>p06: 2-7. Added description of joining tool<br>p17: 7-1. Added description |
| ver. 2014.07.22 | 2014/07/22 | • Corrected errors and omissions<br>• Other:<br>pp18, 20: Added start_time parameter |
| ver. 2014.08.31 | 2014/08/31 | • Corrected errors and omissions |
| ver. 2015.10.13 | 2015/10/13 | • Amended for release of JAGURS-D_V0200 (&V0330)<br>p11: Added section 4<br>p20: Added section 7 and summarized relationships (& organized supplementary material)<br>p40: Added section 8. Compilation parameters<br>pp43, 45: Amended parameter descriptions in section 8<br>• Corrected errors and omissions |
| ver. 2015.11.11 | 2015/11/11 | • Amended for release of JAGURS-D_V0203 (&V0333) and ncdmerge.V0180<br>pp22, 23: Added section 7-2, C. and F.<br>p22: Amended section 7-2, E. (setting the water depth beyond which the horizontal displacement effect is considered)<br>pp43, 45: Amended parameter descriptions in section 8 |
| Ver. 2016.01.29 | 2016/01/29 | • Translated in English |
| Ver. 2016.10.24 | 2016/10/24 | • Amended for release of JAGURS-D_V0400<br>p40: Added description about compile options SKIP_MAX_VEL and LESS_CC<br>p41: Added description about runtime parameter itgrn |
| Ver. 2018.11.9 | 2018/11/9 | Major update for release of JAGURS-D_V0502. |

## 10. Version history (cont.)

| Version | Date | History of modification |
|---|---|---|
| Ver. 2024.01.31 | 2024/01/31 | Major update for release of JAGURS-D_V0520.<br>・ Added description regarding file input/output in pixel format<br>・ Modified description as nf format is supported as a GMT grid file format |
| Ver. 2024.05.17 | 2024/05/17 | Major update for release of JAGURS-D_V0600.<br>・ Added description regarding normal mode calcuration |
| Ver. 2024.06.03 | 2024/06/03 | Modifications on how to make the normal mode calculation program fortran version in "7-5. Atmospheric pressure wave generated from great volcanic eruptions and air-coupled tsunami calculations". |