JAGURS Users Guide

ver. 2016.10.24
For JAGURS-D_V0400

1. Contents

1. Contents 2
2. Introduction 4
3. Explanation of the sample data 5
3-1. Sample data list 5
3-2. JAGURS-D_V0400/ – Complete source code 6
3-3. bathy.xxxx.grd – Terrain datafiles 6
3-4. disp.xxxx.grd – Crustal displacement datafiles 6
3-5. wetordry.xxxx.grd – WetOrDry files (optional) 6
3-6. Parameter files 7
A. gridfile.dat – Parameter file defining terrain nesting 7
B. test_tgs.txt – Parameter file specifying waveform output points 7
C. tsun.par – Computing parameter file 8
3-7. Utilities 10
A. JAGURS_Tools_V0110/ 10
B. ncdmerge.V0180/ 10
C. Other 10
4. Installing the execution environment (libraries) 11
4-1. NetCDF 11
4-2. PROJ4 11
4-3. FFTW3 11
5. Computing in the Serial Version 12
5-1. Serial Version input files 12
5-2. Specifying the parameter files 12
5-3. Compiling the Serial Version 12
5-4. Executing the Serial Version 13
5-5. Serial Version output files 13
6. Computing in the Parallel Version 14
6-1. Parallel Version input files 14
6-2. Dividing the gridfiles 15
6-3. Specifying the parameter files 16
6-4. Compiling the Parallel Version 17
6-5. Executing the Parallel Version 17
6-6. Parallel Version output files 18
6-7. Joining the gridfiles 19
7. Advanced calculations 20
7-1. Inputs 20
A. Obtaining initial water heights by interpolating heights in the parent domains 20
B. Considering rupture propagation in a fault 21
C. Inputting fault parameters and computing crustal displacement in JAGURS 22
D. Simultaneous considering of A, B & C 23
E. Effect of horizontal displacement of a seafloor slope 24
F. Kajiura filter 24
G. Obtaining initial crustal displacement in a Gaussian distribution 25
H. Inputting a sinusoidal wave at a boundary face 26

7-2. Computations 27
A. Computing with Boussinesq dispersion 27
B. Absorbing boundary conditions 28
C. Elastic loading and seawater compressibility 29
D. Executing multiple scenarios in a single job 30
E. Restarting computing 32
F. Cartesian coordinates 32
7-3. Outputs 33
A. Directory output (DIROUT) 33
B. NetCDF format output (NCDIO) 34
C. Calculating maximum flow depths 38
8. Parameter summary 39
8-1. Compile options 39
8-2. Parameter List for running JAGURS (tsun.par) 41
9. Afterword 44
10. Version history 45

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 a tsunami and variations in seawater density 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 and nesting of terrain grids can be applied. A range of other functions are also available. The code is written in Fortran 90 with parallelization by using OpenMP and MPI. References used in the development of JAGURS are listed below.

References

[Non-linear long waves]

Satake, K. 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, Towards spatially distributed quantitative assessment of tsunami inundation models, Ocean Dynamics, doi:10.1007/s10236-010-0312-4, 2010.

[Parallel computing]

Baba, T., N. Takahashi, Y. Kaneda, Y. Inazawa and M. Kikkojin, 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, DOI 10. 1007/978-94-007-7269-4_3, 2014.

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

[Dispersive waves]

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

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

[Effects of elastic loading and seawater compressibility]

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

[Others]

Kajiura, K. (1963), The leading wave of a tsunami, Bull. Earthq. Res. Inst., Vol. 41, pp. 535-571, 1963.

Tanioka, Y. and K. Satake (1996), Tsunami generation by horizontal displacement of ocean bottom, Geophys. Res. Lett., Vol. 23, pp. 861-864, 1996.

3. Explanation of the sample data

3-1. Sample data list

The sample dataset includes the following files.

[Source code]

JAGURS-D_V0400/ ... 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

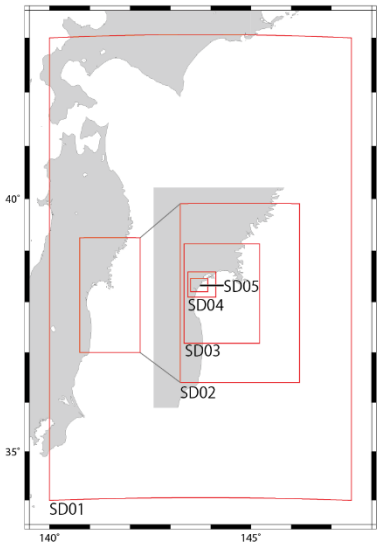
[Script files]

TOOLS/ ... shell scripts used for dividing and joining grid data, etc.

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



3-2. JAGURS-D_V0400/ – Complete source code

This directory contains the JAGURS source codes.

3-3. 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 (cf format)[#1]
- Data orientation ... Vertical downward (negative values for land, positive values for sea depths), unit: m

3-4. 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.

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

3-5. wetordry.xxxx.grd – WetOrDry files (optional)

These files are used for explicitly distinguishing between land and 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, 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 (cf format) [#1]
- Data contents ... negative values for land, positive values for sea

[#1] Note that this is the cf format (#10), not the GMT default nf format (#18). Input and output files in JAGURS are in the cf format (#10).

3-6. 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) (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 the like. 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 length (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).

3-7. Utilities

These are included in TOOLS/.

A. JAGURS_Tools_V0110/

☆ grd2dsplit.sh ... grid file division script

This script divides a grid file into sub-domain files. Copy the script into the same directory as the terrain data that is to be divided to use it.

☆ split2dgrd.sh, chgxy.sh ... grid data joining script

The reverse of the division script, this script joins divided grid data that is outputted from JAGURS. Copy the script into the same directory as the output data to be joined to use it.

If "-nocopy" is added to split2dgrd.sh as an argument, the unjoined files are symbolically linked without being copied. (The unjoined files are modified; so, care should be taken when using them.)

☆ checkxyz.sh... grid data comparison script

This script compares two gmt files and outputs a file listing the differences.

☆ splittgs.sh ... station file conversion script

If "SINGLE_TGS=ON" is set in the JAGURS compilation options, the waveform output point data is outputted to each MPI process in the form "tgs_station.[rank number]". This script reads all these data, converts them to a conventional single file (for the respective stations), and outputs this file. Copy the script into the same directory as the output data to be converted to use it.

☆ omake/

Use these as necessary.

- split2dgrd.do.sh ... a version of split2dgrd.sh compatible with the DIROUT option(#2).
- drive_s2g/ ... for successive execution of split2dgrd.sh (or split2dgrd.do.sh)
- s2g.sh ... for successive execution of split2dgrd.sh (an old version)

B. ncdmerge.V0180/

This program joins files that have been divided and outputted from JAGURS in the NCDIO option(#3). After the source is compiled, copy the execution file "ncdmerge" and the parameter file "namelist" that have been created into the same directory as the output data to be joined and then use them. Execute as

```
$ ncdmerge < namelist
```

C. Other

☆ 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.

(#2) This is a compilation option to output the JAGURS output files directory by directory. See the [OUTPUT] section of "8-1. Compile options" for details.

(#3) This is a compilation option to combine and output the JAGURS output files in the NetCDF format. See the [OUTPUT] section of "8-1. Compile options" for details.

4. Installing the execution environment (libraries)

The environment (libraries) needed for running JAGURS is installed in advance. For installation procedures of the respective libraries, see the README files associated with the source code and should also be referred to.

4-1. NetCDF

This library is for handling data in the NetCDF format.

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

4-2. PROJ4

This library is for converting map projections from spherical to Cartesian coordinates and vice versa.

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

4-3. FFTW3

This library is for fast computing of discrete Fourier transforms.

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

5. Computing in the Serial Version

5-1. Serial Version input files

Put the following files into the same directory.

Source code	JAGURS-D_V0xxx/	
Terrain datafiles	bathy.SD01.grd bathy.SD02.grd bathy.SD03.grd bathy.SD04grd 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	(optional)
Parameter files	gridfile.dat	
	test_tgs.txt	(optional)
	tsun.par	

5-2. 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)

5-3. Compiling the Serial Version

(1) Edit the Make file (Makefile.SC_ICE) in the source code directory.

[Main editing points]

```
NETCDF=/opt/atlocal/netcdf/4.1.3                    ←Path to NetCDF library in the execution environment
PROJ4_DIR=/home/G10004/t-katou/JAGURS/local         ←Path to PROJ4 library in the execution environment
FFTW3_INCLUDE_DIR=$(MKLROOT)/include/fftw           ←Path to FFTW3 library in the execution environment
#MPI=ON                                            ←Switch between Serial Version and Parallel Version; comment this out for the Serial Version
```

(2) Execute compilation with the following command for example.

```
$ make -f Makefile.SC_ICE
```

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. Serial Version output files

If no file output format is specified[#4], the following files are written directly to the execution directory.

Time history of water height and velocity at each waveform output point	tgs000001 ~ tgsnnnnnn	• File format: text file (ASCII) • The water heights and velocities at the respective output points are outputted in step order. • 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 SD02.initl_disp.grd SD03.initl_disp.grd SD04.initl_disp.grd SD05.initl_disp.grd	• File format: GMT gridfile • An initial water level is outputted for each domain.
Maximum water height in each domain	SD01.zmax.grd SD02.zmax.grd SD03.zmax.grd SD04.zmax.grd SD05.zmax.grd	• File format: GMT gridfile • A maximum wave height is outputted for each domain.
Maximum velocity in each domain	SD01.vmax.grd SD02.vmax.grd SD03.vmax.grd SD04.vmax.grd SD05.vmax.grd	• File format: GMT gridfile • A maximum velocity is outputted for each domain.
Water height and velocity for each domain and unit of time passed[#5]	SD01.00001200.grd ~ SD01.nnnnnnnn.grd […] SD05.00001200.grd ~ SD05.nnnnnnnn.grd	• File format: GMT gridfile • Water heights and velocities for each unit of time passed (snapshot output interval) are outputted for each domain. • The numbers in the filenames (00001200, etc.) correspond to values set by the snapshot output interval (itmap) in the computing parameter file (tsun.par).

[#4] The file output format is specified in the compilation options.

[#5] Whether the velocities are outputted or not is specified in the computing parameter file (tsun.par).

6. Computing in the Parallel Version

JAGURS is designed to carry out parallel computations using OpenMP and MPI.

6-1. Parallel Version input files

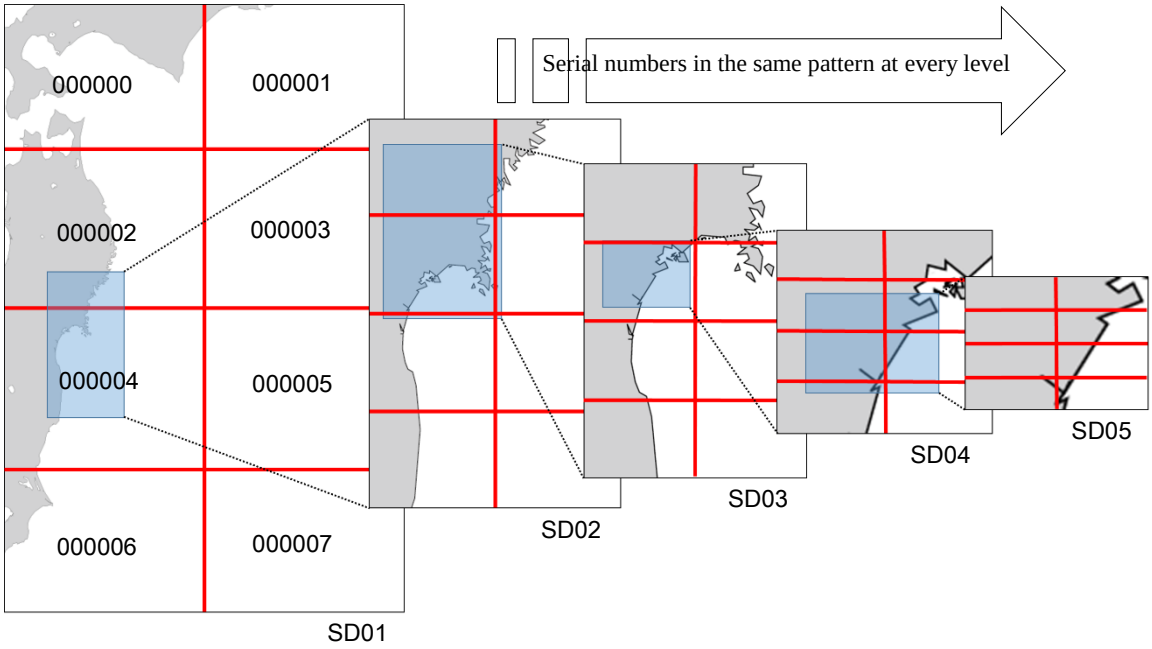
Put the following files into the same directory.

Source code	JAGURS-D_V0xxx/	
Terrain data	bathy.SD01.grd.000000 ～ nnnnnn bathy.SD02.grd.000000 ～ nnnnnn bathy.SD03.grd.000000 ～ nnnnnn bathy.SD04.grd.000000 ～ nnnnnn bathy.SD05.grd.000000 ～ nnnnnn	(divided data)
Crustal displacement data	disp.SD01.grd.000000 ～ nnnnnn disp.SD02.grd.000000 ～ nnnnnn disp.SD03.grd.000000 ～ nnnnnn disp.SD04.grd.000000 ～ nnnnnn disp.SD05.grd.000000 ～ nnnnnn	(divided data)
WetOrDry files	wetordry.SD05.grd.000000～ nnnnnn	(divided data, optional)
Parameter files	gridfile.dat	
	test_tgs.txt	(optional)
	tsun.par	
Shell scripts	grd2dsplit.sh	These are in JAGURS_Tools_V0110/ and should be copied.
	split2dgrd.sh	
	chxy.sh	

6-2. Dividing the gridfiles

For computing using MPI, each grid file are divided into sub-domains.

For example, if the numbers of divisions are "X direction: 2, Y direction: 4", every domain is divided into the same numbers of divisions. Successive numbers are assigned to the divided data. Data with the same number in each domain are passed to the same MPI process. The results of computing are outputted separately from the respective MPI processes. The division must be applied to all the gridfiles that are being used (terrain data, crustal displacement data and WetOrDry data).



A gridfile division script (grd2dsplit.sh) is packaged with the sample dataset. Use these scripts to divide the input gridfiles.

[Description Example]

```
./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f bathy.SD01.grd
```

Column 1: -px ... number of divisions in the X direction (default: 1)
Column 2: -py ... number of divisions in the Y direction (default: 1)
Column 3: -eps ... latitude and longitude precision (default: 0.0000000000001)
Column 4: -ffmt... format descriptor when latitude and longitude are passed to the grdcut command (default: %.10f)
Column 5: bathy.xxxx.grd ... filename of the terrain data to be divided

[Operation procedure]

(1) Put the gridfile division script (grd2dsplit.sh) in the same directory as the data to be divided.

(2) Run the gridfile division script (grd2dsplit.sh).

If the terrain data is to be divided in 2 in the X direction and 4 in the Y direction:

```
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f bathy.SD01.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f bathy.SD02.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f bathy.SD03.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f bathy.SD04.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.000000001 -ffmt %.8f bathy.SD05.grd (#6)
```

(3) In the same manner, divide all gridfiles to be used (crustal displacement data and WetOrDry data) into the same numbers of divisions.

```
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f disp.SD01.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f disp.SD02.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f disp.SD03.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f disp.SD04.grd
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.000000001 -ffmt %.8f disp.SD05.grd (#6)
> ./grd2dsplit.sh -px 2 -py 4 -eps 0.000000001 -ffmt %.8f wetordry.SD05.grd (#6)
```

6-3. 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)

In the Parallel Version, do not forget specifications of the numbers of divisions 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)
```

(#6) This script calls the GMT command grdcut to divide the gridfiles. Because of numerical errors during the division, there may occur errors in latitude and longitude expression, as a result of which execution of the command may fail. If this happens, adjust the values of the options -eps and -ffmt and try again. (Try again with the precision of the option -eps in the sample dataset lowered by one step.)

6-4. Compiling the Parallel Version

(1) Move to the source code directory and edit the Make file (Makefile.SC_ICE).

[Main editing points]

```
NETCDF=/opt/atlocal/netcdf/3.6.3                    ←Path to NetCDF library in the execution environment
PROJ4_DIR=/home/G10004/t-katou/JAGURS/local         ←Path to PROJ4 library in the execution environment
FFTW3_INCLUDE_DIR=$(MKLROOT)/include/fftw           ←Path to FFTW3 library in the execution environment
MPI=ON          ←Switch between Serial Version and Parallel Version; make sure this is not commented out for the Parallel Version
```

(2) Execute compilation with the following command.

```
$ make -f Makefile.SC_ICE
```

6-5. Executing the Parallel Version

After dividing up all the gridfiles, and writing the numbers of divisions into the computing parameter file (tsun.par), execute the computing. In parallel computing, 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
```

6-6. Parallel Version output files

If no file output format is specified(#7), the following files are written directly to the execution directory.

Time history of water height and velocity at each waveform output point	tgs000001 ~ tgsnnnnnn	• File format: text file (ASCII) • The water heights and velocities at the respective output points are outputted in step order. • 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.000000 ～ SD01.initl_disp.grd.nnnnnn […] SD05.initl_disp.grd.000000 ～ SD01.initl_disp.grd.nnnnnn	• File format: GMT gridfile • An initial water level is outputted for each domain. • The numbers in the filenames (000000, etc.) correspond to the serial numbers of the divided gridfiles.
Maximum water height in each domain	SD01.zmax.grd.000000 ~ SD01. zmax.grd.nnnnnn […] SD05. zmax.grd.000000 ~ SD05. zmax.grd.nnnnnn	• File format: GMT gridfile • A maximum water height is outputted for each domain. • The numbers in the filenames (000000, etc.) correspond to the serial numbers of the divided gridfiles.
Maximum velocity in each domain	SD01.vmax.grd.000000 ~ SD01.vmax.grd.nnnnnn […] SD05. vmax.grd.000000 ~ SD05.vmax.grd.nnnnnn	• File format: GMT gridfile • A maximum velocity is outputted for each domain. • The numbers in the filenames (000000, etc.) correspond to the serial numbers of the divided gridfiles.
Water height and velocity for each domain and unit of time elapsed(#8)	SD01.00001200.grd.000000 ~ SD01.00001200.grd.nnnnnn ～ SD01.nnnnnnnn.grd.000000 ~ SD01.nnnnnnnn.grd.nnnnnn […] SD05.00001200.grd.000000 ~ SD05.00001200.grd.nnnnnn ～ SD05.nnnnnnnn.grd.000000 ~ SD05.nnnnnnnn.grd.nnnnnn	• File format: GMT gridfile • Wave heights and velocities for each unit of time elapsed (snapshot output interval) are outputted for each domain. • The numbers in the middle of the filenames (00001200, etc.) correspond to values defined by the snapshot output interval (itmap) in the computing parameter file (tsun.par). • The numbers at the end of the filenames (000000, etc.) correspond to the serial numbers of the divided gridfiles.
Computation log	stdout.000000 ~ stdout.nnnnnn	• File format: text file • standard outputs are stored in the files corresponding to rank number

(#7) The file output format is specified in the compilation options.

(#8) Whether velocities are outputted or not is specified in the computing parameter file (tsun.par).

6-7. Joining the gridfiles

The types of output files are basically the same as in the Serial Version. However, each output gridfile is outputted in the same divided state as the input file. A gridfile joining script (split2dgrd.sh) and a support script (chgxy.sh) are packaged with the sample dataset. Use these scripts to join the output gridfiles.

[Description Example]

```
./split2dgrd.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f -ref bathy.SD01.grd SD01.00001200.grd
```

Column 1: -px ... number of divisions in the X direction (default: 1)
Column 2: -py ... number of divisions in the Y direction (default: 1)
Column 3: -eps ... latitude and longitude precision (default: 0.0000000000001)
Column 4: -ffmt ... format descriptor when latitude and longitude are passed to the grdcut command (default: %.15f)
Column 5: -ref ... name of a terrain datafile to be referred to during joining (latitudes and longitudes of the edges of the domain are corrected to match this file)
Column 6: xxxx.xxxx.grd ... base name of the gridfiles to be joined

[Operation procedure]

(1) Put the gridfile joining script (split2dgrd.sh) and the support script (chgxy.sh) in the same directory as the data to be joined.

(2) Run the gridfile joining script (split2dgrd.sh).

If the terrain data was divided in 2 in the X direction and 4 in the Y direction for the computing:

```
> ./split2dgrd.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f -ref bathy.SD01.grd SD01.00001200.grd
> ./split2dgrd.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f -ref bathy.SD02.grd SD02.00001200.grd
> ./split2dgrd.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f -ref bathy.SD03.grd SD03.00001200.grd
> ./split2dgrd.sh -px 2 -py 4 -eps 0.0000000001 -ffmt %.8f -ref bathy.SD04.grd SD04.00001200.grd
> ./split2dgrd.sh -px 2 -py 4 -eps 0.000000001 -ffmt %.8f -ref bathy.SD05.grd SD05.00001200.grd(#9)
```

(3) In the same manner, join all the output gridfiles (water heights, maximum water heights and maximum velocities in each output interval for all remaining snapshots).

(#9) Similarly to the division, this script calls the GMT command grdcut to join together the gridfiles. Because of numerical errors during the joining, there may occur errors in longitude and latitude expression, as a result of which execution of the command may fail. If this happens, adjust the values of the options -eps and -ffmt and try again.

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 by preparing 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. Simultaneous considering of A, B & C

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 long depth length width dip strike rake 0min
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.

E. 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".

F. 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).

G. Obtaining initial crustal displacement in a Gaussian distribution

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  ←No disp file is required;
SD02 SD01 0 bathy.SD02.grd  NO_DISPLACEMENT_FILE_GIVEN    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]. If a dispersive term is applied, the duration of computing increases. Therefore, this computing is generally performed using the Parallel Version. The procedure is the same as in "6. Computing in the Parallel Version".

[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 wave is transmitting at the outer edges of the root domain. If absorbing boundary conditions are specified, waves reaching the outer edges of the root domain are absorbed at those boundaries.

[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(#10)

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 (#10)

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

(#10) 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].

[Files to be edited]

◆ JAGURS Make file

```
REAL_FFT=ON
```

◆ 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.*".) Put the directories (input.xxxxxx/) at the same level as the JAGURS execution file.

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 number of cases.

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

E. Restarting 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.

Serial Version: restart.[step number (8 digits)]

Parallel Version: restart.[step number (8 digits)].[process number (6 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.

[Files to be edited]

◆ JAGURS Make file

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

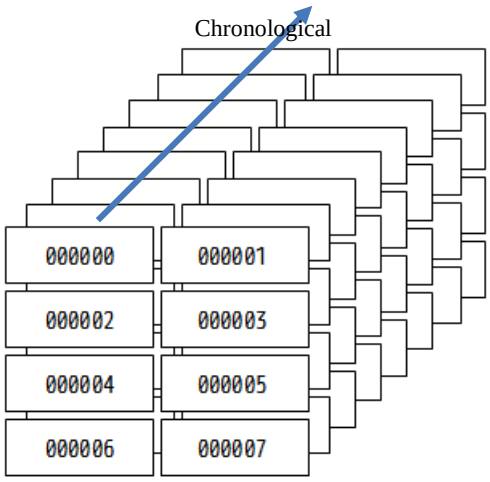
・ Handling of input data is the same as with the Spherical coordinates Version. Prepare input data with x,y coordinates in meters.

7-3. Outputs

A. Directory output (DIROUT)

If this option is set by OUTPUT being left unspecified in the compilation options, output files of wave height and velocity are written to the program execution directory. If "OUTPUT=DIROUT" is set, output files of wave height and velocity are gathered in a directory for each time step and then written to the program execution directory.

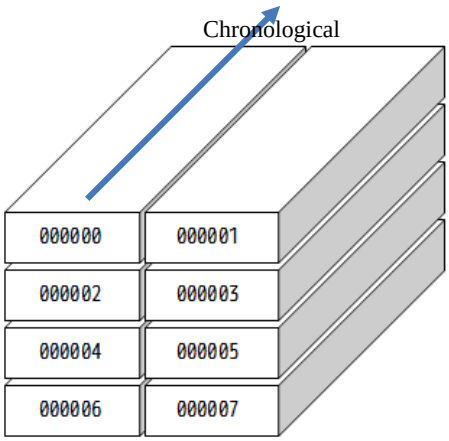
The files themselves are divided into node units (by MPI division numbers).



Files written for respective nodes (and steps)
(cf format (#10))

B. NetCDF format output (NCDIO)

If the "OUTPUT=NCDIO" option is set in the compilation options, output files of wave height and velocity are collected into a single NetCDF file for each node (each MPI division number) and written to the program execution directory.

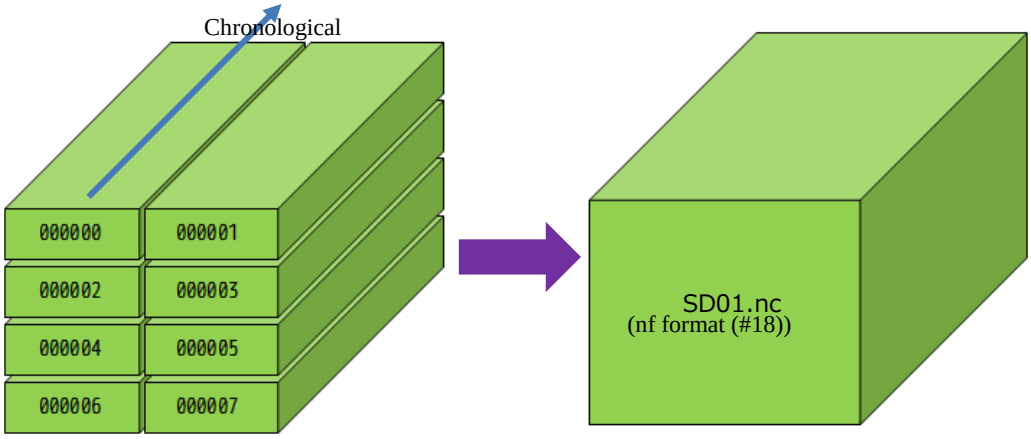


NetCDF files written for respective nodes
(nf format (#18))

B-1. Joining all data

The NetCDF files that have been written chronologically for each node (MPI division number) are all joined together by “ncdmerge”.

Example: Joining all computing results for domain SD01



(1) Edit the parameter file "namelist" for the processing program "ncdmerge".

```
&params
basename='SD01'          ←Domain name
only_step=0              ←0: Join all steps
/
```

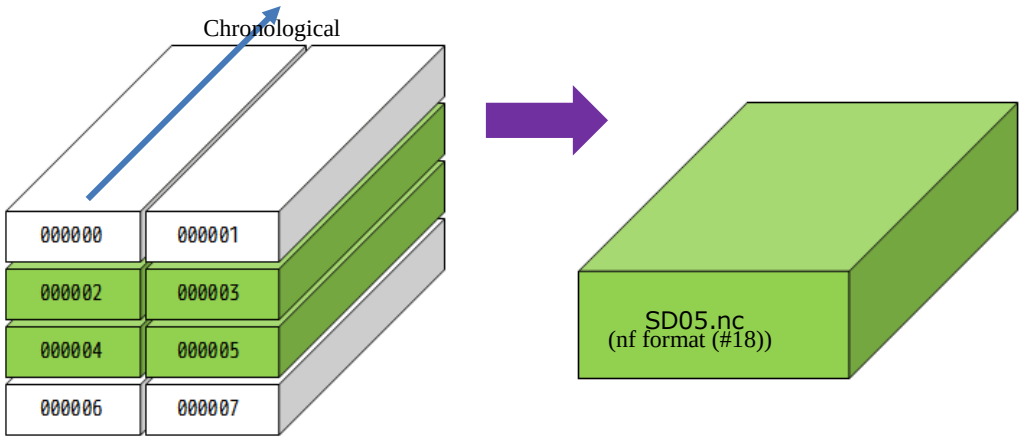
(2) Run "ncdmerge".

```
$ ./ncdmerge < namelist
```

B-2. Joining adjacent nodes

Adjacent nodes (MPI division numbers) are specified and joined.

Example: Joining data for MPI division numbers 000002, 000003, 000004 and 000005 in the computing results for domain SD05.



(1) Edit the parameter file "namelist" for the processing script "ncdmerge".

```
&params
basename='SD01'          ←Domain name
root_proc = 2            ←MPI division number of startpoint data
xnum = 2                 ←Number to be joined (X direction)
ynum = 2                 ←Number to be joined (Y direction)
/
```

(2) Run "ncdmerge".

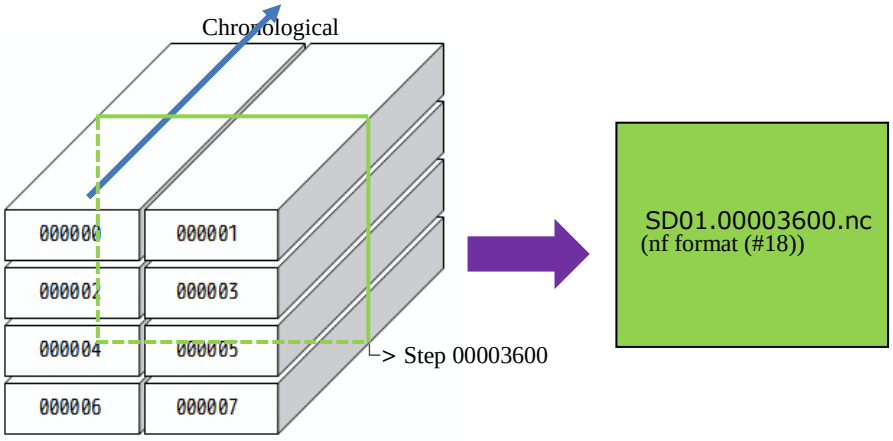
```
$ . /ncdmerge < namelist
```

- Only outputs corresponding to the processes counting for xnum in the X direction and ynum in the Y direction from the process specified by root_proc are merged.
- If root_proc is not specified, all processes are merged as usual.
- If root_proc is specified but values are not specified for xnum and ynum, all processes downward and to the right from root_proc are merged, as if xnum and ynum were set to the largest values at the extremes.

B-3. Sampling only a particular time step and statistical data

A particular time step or maximum wave heights (zmax) or maximum velocities (vmax) or the like at a particular time step can be joined and extracted for a particular location. The file data does not include time data.

Example: Extracting step 3600 from computing results for domain SD01



(1) Edit the parameter file "namelist" for the processing script "ncdmerge".

```
&params
basename='SD01'          ←Domain name
only_step=3600           ←Target step number
/
```

only_step=0: Join all steps
only_step=-1: Join only initial water levels
only_step=-2: Join only maximum wave heights
only_step=-3: Join only maximum velocities

(2) Run "ncdmerge".

```
$ . /ncdmerge < namelist
```

☆ Additional note: The grdreformat command can sample only a particular time step and statistical data from files written in the NCDIO formatted file including multiple data (e.g. B-1).

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, to extract the next step after the initial time step, it must be set to wave_height[1].

However, extraction with wave_height[0] does not extract the initial time step but a snapshot of the time step at which the first output is performed, and wave_height[1] is for when the next output is performed after a specified interval has passed.

That is, if snapshots are set to be produced at 10 second intervals, wave_height[0] is a snapshot after a cumulative duration of 10 seconds and wave_height[1] is a snapshot after a cumulative duration of 20 seconds.

C. 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
```

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 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. • File format: NetCDF • 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. • File format: GMT grid • Filename: "[step].grd" / "[domain].[step number].grd.[rank number]"
	Other	All output files are written to the program execution directory. • File format: GMT gird • Filename: "[domain].[step number].grd.[rank number]"
SINGLE_TGS	ON	Water height and velocity at each waveform output point (station) are written into a single file per process. • Output filename: "tgs_station.[rank number]" (written for each process) • 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.) • 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. • Output filename: "tgs[station (6 digits)]"

Item	Value	Comment
MULTI	ON	Execute multiple scenarios in a single job
REAL_FFT(#12)	ON	Use real fast Fourier transforms to compute elastic loading
	Other	Use complex fast Fourier transforms to compute elastic loading
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.

(#12) There is no particular benefit in choosing to use complex FFT. We plan to remove this option when we have obtained sufficient results.

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)		
vmaxgrdfn	＊	Output file for max velocity (ex:vmax.grd)		
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	(If cfl is undefined, it will be overwritten by cf)
cfl	＊	For land	(negative value): Manning's roughness coefficient	
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 (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 ・plotgrd is <0 or > n-Grid: all of them are output ・Else: Only the domain which you choose are output (ex:plotgrd=2,5 … domain 2 & 5 are output 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) (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

（＊: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) 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) 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 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'

（＊: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)	

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 in advance.

Toshitaka Baba
Tokushima University/
Japan Agency Marine-Earth Science and Technology
baba.toshi@tokushima-u.ac.jp

Phil R. Cummins
Australian National University/
Geoscience Australia
phil.cummins@anu.edu.au
mailto:baba.toshi@tokushima-u.ac.jp

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 • Other: p02: 2-1. Version change with JAGURS-D_V0177 release p03: 2-5. Added description p04: 2-6-2. Added description p05: 2-6-3. Changed sample parameter value p06: 2-7. Added file description p07: 3-2. Added description p10: 4-2. Added description and changed sample parameter value p11: 4-3. Added description p19: 7-2. Added description of JAGURS-D_V0177 release
ver. 2014.07.14	2014/07/14	• Corrected errors and omissions • Other: p02: 2-1. Version change with JAGURS-D_V0178 release p06: 2-7. Added description of joining tool p17: 7-1. Added description
ver. 2014.07.22	2014/07/22	• Corrected errors and omissions • Other: 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) p11: Added section 4 p20: Added section 7 and summarized relationships (& organized supplementary material) p40: Added section 8. Compilation parameters pp43, 45: Amended parameter descriptions in section 8 • Corrected errors and omissions
ver. 2015.11.11	2015/11/11	• Amended for release of JAGURS-D_V0203 (&V0333) and ncdmerge.V0180 pp22, 23: Added section 7-2, C. and F. p22: Amended section 7-2, E. (setting the water depth beyond which the horizontal displacement effect is considered) 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 p40: Added description about compile options SKIP_MAX_VEL and LESS_CC p41: Added description about runtime parameter itgrn