OpenSWPC:
An Open-source
Seismic Wave Propagation Code
User's Guide

Version 3.0-E (2016-08-11)

Takuto Maeda

Table of Contents

Chapter 1

Set Up

1.1 System Requirements

To execute the `OpenSWPC`, a Fortran compiler that can handle (at least a part of) the Fortran 2003 standard, and a MPI library. The program can be run under the single CPU or CPU core without parallelization, however, the MPI library is necessary even in that case. Also, `NetCDF` library compiled by the same Fortran compiler is recommended to use the direct input/output of the `NetCDF`-formatted files.

The source code of SEISM almost strictly follows the language standard of the Fortran2003. As an exception, a system call (`system()` subroutine) is used. However this extension is supported most of the available Fortran compilers. The major functionality of Fortran2003 used in the SEISM is stream I/O. Note that functionality is not implemented in the old compilers.

This code was developed under the following environment:

- Apple OSX El Capitan
- GNU `gfortran` 6.1.0
- `OpenMPI` 1.10.2

In addition, the following computers are confirmed to work fine with the `OpenSWPC`:

- EIC computer, ERI/UTokyo (ver. 2015; SGI Altix; intel fortran)
- JAMSTEC Earth Simulator (NEC SX-ACE; NEC compiler)
- AICS K computer (Fujitsu compiler)
- Nagoya University (FX10/Fx100; Fujitsu compiler)
- Linux Cent OS 6.6 (gfortran 4.9.2 & mpich)
- Linux Ubuntu 16.04LTS (gfortran 5.4 & OpenMPI)

1.2 Code directory tree

```
|-- doc : manuals
|-- bin : executable binaries (*.x)
|-- example : example input files
\-- src
```

Table 1.1 `arch` options for various environments

arch name	target	`NetCDF` location
mac-intel	Mac OSX + Intel Compiler + Open MPI	`${HOME}/local`
mac-gfortran	Mac OSX + gfortran + Open MPI	`/usr/local`
eic	EIC (ERI, UTokyo) with the Intel Compiler	`${HOME}/local`
fx	Fujitsu FX10, FX100 and the K-computer	`${HOME}/xlocal`
es3	The Earth Simulator	Provided by the system
ubuntu-gfortran	Ubuntu 16.04LTS + gfortran + Open MPI	Installation by apt

```
|-- swpc_3d    : 3D problem
|-- swpc_psv   : 2D P-SV problem
|-- swpc_sh    : 2D SH problem
|-- tools      : Miscellaneous utility codes
\-- shared     : Modules commonly used for above programs
```

1.3 Compilation and Execution

1.3.1 make

Directories `src/swpc_3d`, `src/swpc_psv`, `src/swpc_sh`, `src/tools` contain `makefile`. Execute `make` command at each directory to generate executable binaries. Executable file (with *.x extension) will be stored `bin` directory. .

1.3.2 Specifying Compiler Options

In the makefiles, the following variable must be specified according to the environment:

`FC` compiler name
`FFLAGS` compiler option
`NCFLAG` `NetCDF` flag
`NCLIB` location of `NetCDF` library directory
`NCINC` location of `NetCDF` include directory
`NETCDF` linker option for the `NetCDF`

If `NCFLAG = -D_NETCDF` is specified, the make tries to compile `OpenSWPC` with `NetCDF`.

A set of the above variables under different computer environments are defined in `src/shared/makefile.arch` and `src/shared/makefile-tools.arch`. The former is for compilation of `swpc`, and the latter is for misc tools. The user can specify `arch` option at make like:

```
1 make arch=eic debug=true
```

The list of pre-defined architecture (`arch`) options are described in Table 1.1.

1.3.3 More about NetCDF library

The NetCDF library consists of following items:

`libnetcdf.*` NetCDF library file (static)
`libnetcdff.*` NetCDF Fortran library file (only NetCDF version 4 or later)
`netcdf.mod` Fortran module information file

The extension of library files may be `*.a` (static library) or `*.so` (dynamic library), depending on the installation. All of these files are necessary for successful compilation with NetCDF. In particular, `netcdf.mod` file must be created by the same Fortran compiler with the OpenSWPC. If the NetCDF is installed through packaging tools such as `yum, apt, homebrew`, the use of `gfortran` is implicitly assumed.

1.3.4 Preparing Dataset

Subsurface Velocity Structure Model
In OpenSWPC, the 3D inhomogeneous medium can be represented as a set of velocity discontinuities with NetCDF formatted-files (see section 2.7 for detail). As an example of velocity structure beneath Japanese Archipelago, an automatic model generation script for the Japan Integrated Velocity Structure Model (JIVSM; Figure 1.1; (*Koketsu et al.*, 2012)) developed and originally distributed by the Headquarters for Earthquake Research Promotion, Japan. An extension of the JIVSM (eJIVSM) that covers wider area also is provided. These velocity structure model contains ground surface (topography and bathymetry), subsurface soil, moho, and oceanic crust of two subducting plates. To generate these models, GMT version 4 is required. If user do not use this model, the following processing may not be necessary.

First, download the original model files of `lp2012nankai-e_str.zip` and `lp2012nankai-w_str.zip` and store them at `dataset/vmodel`. The URLs of these files can be found at the comments of `gen_JIVSM.sh` script. To generate eJIVSM model, the topography data of ETOPO1 (`ETOPO1_Bed_g_gmt4.grd`) is necessary too.

Then, specify the fortran compiler name to `FC` variables in the parameter file `params.sh`. Grid spacing `dlon, dlat` in the parameter file can be modified if necessary. Note that this spacing is not directly related to the grid width of numerical simulations. At the time of simulation, the OpenSWPC program automatically interpolate the velocity model data.

After these preparations, exectute generation script:

```
1 ./gen_JIVSM.sh
```

After successful execution, 23 NetCDF-formatted files are generated at two model directories of texttt{JIVSM and `eJIVSM`. These files can be read and visualized from GMT by such as `grdimage` or `grd2xyz` moduels. The filename of netcdf contains five integer numbers; they correspond to mass density (in kg/m^3), P wavespeed (m/s), S wavespeed (m/s), Q_P, adn Q_S. They are material information below the discontinuity defined the file. List files of these NetCDF files (`jivsm.lst` and `ejivsm.lst`) for using in the OpenSWPC also will be generated.

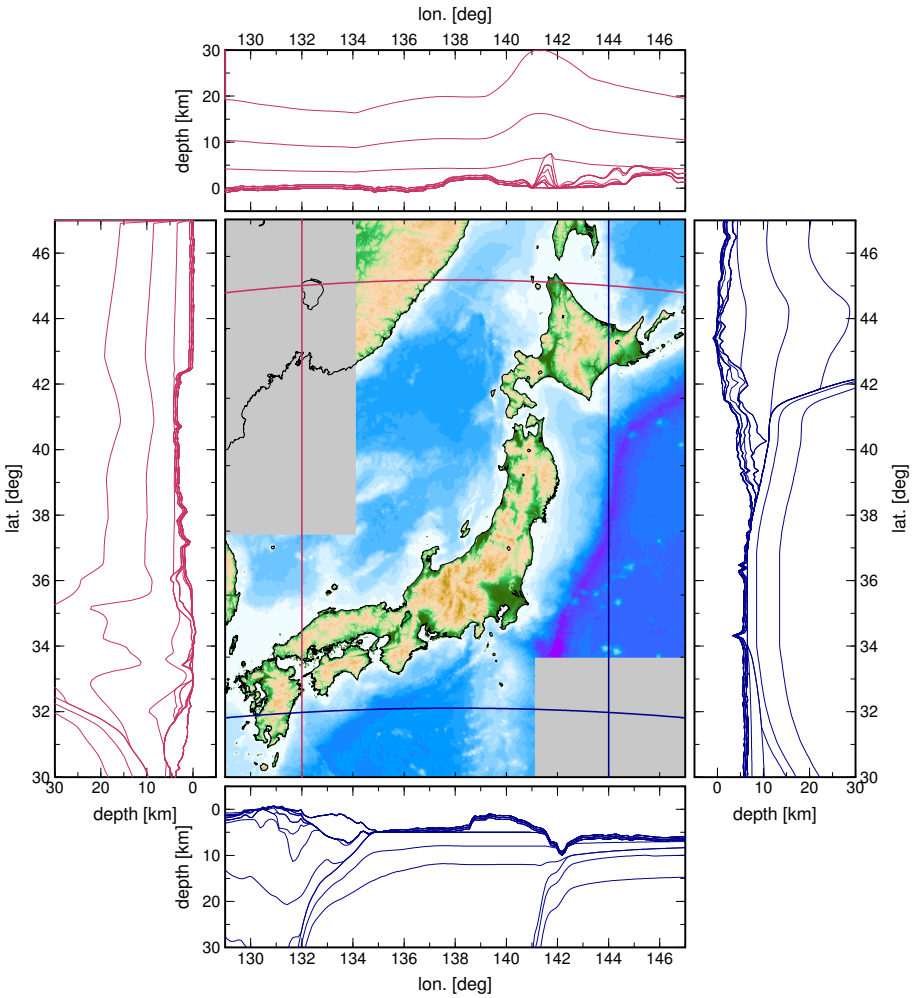

Fig. 1.1 The area of JIVSM and eJIVSM models. The colored area in the map is where the original JIVSM is defined. eJIVSM extends to gray-shaded area by extrapolation. Surrounded graphs show depth-section along the lines on the map of the model.

Station List

An example script to generate a station list file is stored in dataset/station/gen_stlst_hinet.sh. This script generates a formatted list of the High-sensitivity seismograph network Japan (Hi-net) provided by National Research Institute for Earth Science and Disaster Resilience (NIED).

To use this script, first download the station csv list from Hi-net website following to comments in the `gen_stlst_hinet.sh` file. Then, executing this bash script will give the station list file for `OpenSWPC`.

1.3.5 On Embedding Parameters

Although most of the behavior of `OpenSWPC` is controlled dynamically by input parameter file, several parameters are embedded in the source code to achieve high-computational performance as described below. These parameters are defined in `src/swpc_??/m_global.F90`.

If these parameters are changed, re-compilation (make clean; make) is necessary.

`UC = 1E-15` (`1E-12` for 2D codes)
: A number to convert simulation result into SI unit system. Modification may be necessary to use a different unit system.

`MP = DP`
: Precision of finite difference computation. By default (`MP=DP`), a part of the computation are performed in double precision, while the other unnecessary variables are defined and calculated in the single precision for saving memory space and computation time. User may change it to `MP=SP` to make whole computation in the single precision, which will decrease the required memory up to 2/3 with faster computation speed. However in this case, an noisy seismic waveform might be observed in particular around the seismic source due to overflow floating point numbers.

`NM = 3`
: Number of generalized Zener viscoelastic bodies. If this number is larger than 1, it represents nearly-frequency-independent constant Q model among specified frequency range. If this is set to zero, the simulation will be done with elastic body without attenuation. Increasing this number enables to reproduce wider frequency range of constant-Q, however, it may result significant increase of computational loads for 3D simulation.

1.3.6 Execution

To run the program, MPI program is necessary such as

```
> mpirun -np ${NP} ./bin/swpc_3d.x -i ${input}
```

where `${NP}` is number of process of MPI, `${input}` is a name of the input file. Note that name of the `mpirun` command may be different at different computational system.

If the program runs properly, the following message will appear on the standard error output. The result may be slightly different among different programs (3D/P-SV/SH) or execution mode.

```
 ---------------------------------------------------------------------------
  SWPC_3D (benchmark mode)
 ---------------------------------------------------------------------------

  Grid Size               :      384 x  384 x  384
  MPI Partitioning        :        4 x    6
  Total Memory Size       :           12.705  [GiB]
  Node Memory Size        :            0.529  [GiB]
  Stability  Condition c  :            0.980  (c<1)
  Wavelength Condition r  :            7.000  (r>5-10)
  Minimum velocity        :            3.500  [km/s]
  Maximum velocity        :            6.062  [km/s]
  Maximum frequency       :            1.000  [Hz]

 ---------------------------------------------------------------------------

```

```
 it=0000050, 1.877 s/loop, eta 000:29:43, ( 5.00E-05  5.00E-05  4.96E-05 )
 it=0000100, 1.887 s/loop, eta 000:28:18, ( 1.75E-05  1.75E-05  1.05E-05 )
 it=0000150, 1.932 s/loop, eta 000:27:22, ( 1.02E-05  1.02E-05  5.41E-06 )
 it=0000200, 1.943 s/loop, eta 000:25:54, ( 6.59E-06  6.59E-06  4.35E-06 )

           .
           .
           .

 it=0000950, 1.986 s/loop, eta 000:01:39, ( 4.89E-07  4.89E-07  1.81E-06 )
 it=0001000, 1.982 s/loop, eta 000:00:00, ( 1.65E-07  1.65E-07  1.54E-07 )

--------------------------------------------------------------------------------

 Total time             :        1982.348 s

--------------------------------------------------------------------------------
```

The first part of message contains estimated memory usage, stability condition, wavelength condition and so on. As shown in the above example, the stability condition of $c < 1$ is mandatory to execute; if the specified parameter violates this condition, program aborts immediately. Also, the wavelength condition (ratio between spatial grid size and minimum wavelength) is recommended to $r > 5 - 10$. During the computation, the computation speed, the remaining time (eta; estimated time of arrival) and maximum velocity amplitude at components are shown.

Chapter 2

Parameter Settings

2.1 Notation of Parameter File

In parameter files, one parameter is defined in one line in the following format.

```
variable_name = value
```

The description of values should follow Fortran's notation. For example, logical (Boolean) values are denoted as `.true.` or `.false.`.

Lines that does not contains the equal sign (=) will be neglected, but in particular the lines starting from `!` or `#` are regarded as comment line and will be skipped. Comments can be followed by variable definition. For example, the following specification works. Comments can be written in the same line of the parameter definition. For example, the following parameter line works fine.

```
nx = 1024 ! number of grids
```

The order of parameter definition can be freely changed. If a parameter is not specified, `OpenSWPC` may use a pre-defined default variable in some case. In that case, the use of default-parameter will be informed to standard error output. However, there are parameters that must be defined explicitly. Multiple definitions of the same parameters in a single parameter file is not recommended, but if it is the case, the first definition may be adopted. It is okay to have blanks before and after the equal sign, however it is not allowed to have a blank between the minus character and succeeding numbers (e.g., `'- 35.0'` is not allowed). It is recommended to use quotation marks around the string parameter. Without them, the directory path character ('/') may be unexpectedly interpreted as a termination of string parameter.

2.2 An Example Parameter File

The following is a full set of example parameters. In for following sections, detailed descriptions to each parameter will be given.

```

!! ------------------------------------------------------------------- !!
!!
!!  SWPC input file
```

```
!!
!! ------------------------------------------------------------------------ !!


!! ------------------------------------------------------------------------ !!
!! Control
!!

title            = 'swpc'           !! exe title: used for output filenames
odir             = './out'          !! output directory
ntdec_r          = 50               !! screen report timing (1/cycle)


!! ------------------------------------------------------------------------ !!
!! Model/Grid Size and Area
!!

nproc_x          = 4                !! parallelization in x-dir
nproc_y          = 6                !! parallelization in x-dir
nx               = 384              !! total grid number in x-dir
ny               = 384              !! total grid number in y-dir
nz               = 384              !! total grid number in z-dir
nt               = 1000             !! time step number

dx               = 0.5              !! grid width in x-dir
dy               = 0.5              !! grid width in y-dir
dz               = 0.5              !! grid width in z-dir
dt               = 0.02             !! time step width

vcut             = 1.5              !! minimum velocity
                                    !- smaller velocity will be raised

xbeg             = -96.0            !! minimum in x-dir
ybeg             = -96.0            !! minimum in y-dir
zbeg             = -10.0            !! minimum in z-dir
tbeg             = 0.0              !! start time

clon             = 139.7604         !! center longitude
clat             = 35.7182          !! center latitude
phi              = 0.0              !! horizontal coordinate rotation
                                    !- measured clockwise from north

fq_min           = 0.02             !! minimum freq. for Q-const model
fq_max           = 2.00             !! maximum freq. for Q-const model
fq_ref           = 1.0              !! ref. freq. for physical dispersion

!! ------------------------------------------------------------------------ !!
!! Snapshot Output
!!

snp_format       = 'netcdf'         !! snapshot format (native or netcdf)

xy_ps%sw         = .false.          !! P&S amp. for xy section
xz_ps%sw         = .true.           !! P&S amp. for xz section
yz_ps%sw         = .false.          !! P&S amp. for yz section
fs_ps%sw         = .false.          !! P&S amp. for free surface
```

```
ob_ps%sw          = .true.           !! P&S amp. for ocean bottom

xy_v%sw           = .false.          !! 3-comp. velocity for xy section
xz_v%sw           = .true.           !! 3-comp. velocity for xz section
yz_v%sw           = .false.          !! 3-comp. velocity for yz section
fs_v%sw           = .false.          !! 3-comp. velocity for free surface
ob_v%sw           = .true.           !! 3-comp. velocity for ocean bottom

xy_u%sw           = .false.          !! 3-comp. disp. for xy section
xz_u%sw           = .true.           !! 3-comp. disp. for xz section
yz_u%sw           = .false.          !! 3-comp. disp. for yz section
fs_u%sw           = .false.          !! 3-comp. disp. for free surface
ob_u%sw           = .true.           !! 3-comp. disp. for ocean bottom


z0_xy             =  7.0             !! depth for xy cross section
x0_yz             =  0.0             !! x-value for yz cross section
y0_xz             =  0.0             !! y-value for xz cross section

ntdec_s           = 5                !! time decimation of snapshot
                                     !- (specify 1 for no decimation)

idec              = 2                !! x-decimation for snapshot
jdec              = 2                !! y-decimation for snapshot
kdec              = 2                !! z-decimation for snapshot

!! ------------------------------------------------------------------------ !!
!! Waveform Output
!!

sw_wav_v          = .true.           !! velocity trace output at stations
sw_wav_u          = .false.          !! displacement trace output at stations
ntdec_w           = 5                !! time decimation of waveform output
st_format         = 'xy'             !! station format: 'xy' or 'll'
fn_stloc          = './example/stloc.xy'  !! station location file
wav_format        = 'sac'            !! 'sac' or 'csf'

!! ------------------------------------------------------------------------ !!
!! Earthquake Source
!!

!! Moment tensor source format:
!!   xymoij / xym0dc / llm0ij / llm0dc / xymwij / xymwdc / llmwij / llmwdc
!! Body force source fomrat:
!!   xy or ll
stf_format        = 'xym0ij'

!! Basis source time function
!! 'boxcar' / 'triangle' / 'herrmann' / 'kupper' / 'cosine' / 'texp'
stftype           = 'kupper'

fn_stf            = "./example/source.dat"   !! Source grid file name

!! source depth correction
!! 'asis':use z value, 'bd{i}': i-th boundary (i=0...9)
sdep_fit          = 'asis'
```

```
117
118       !! ------------------------------------------------------------------- !!
119       !! Body force source mode
120       !!
121       bf_mode          = .false.
122
123
124       !! ------------------------------------------------------------------- !!
125       !! Plane wave source mode
126       !!
127       pw_mode          = .false.   !! plane wave input. neglects fn_stf
128       pw_ztop          = 100.      !! top z-coord of initial plane wave
129       pw_zlen          = 30.       !! wavelength of initial plane wave
130       pw_ps            = 'p'       !! 'p' P-wave 's' S-wave
131       pw_strike        = 0.0       !! strike direction of plane wave (deg.)
132       pw_dip           = 0.0       !! dip of plane wave (deg.)
133       pw_rake          = 0.0       !! rake of plane S-wave polarization (deg.)
134
135     !! ------------------------------------------------------------------- !!
136     !! Absorbing Boundary Condition
137     !!
138
139     abc_type         = 'pml'            !! 'pml' or 'cerjan'
140     na               = 20               !! absorbing layer thickness
141     stabilize_pml    = .true.           !! avoid low-v layer in PML region
142
143     !! ------------------------------------------------------------------- !!
144     !! Velocity model
145     !!
146
147     vmodel_type      = 'lhm'            !! velocity model type 'uni'/'grd'/'lhm'
148     is_ocean         = .true.           !! topography z<0 is covered by ocean
149     is_flatten       = .false.          !! Force topography variation zero
150
151       !! ------------------------------------------------------------------- !!
152       !! For uniform velocity model 'uni'
153       !!
154       vp0              = 5.0              !! P-wave velocity [km/s]
155       vs0              = 3.0              !! S-wave velocity [km/s]
156       rho0             = 2.7              !! mass density    [g/cm^3]
157       qp0              = 200              !! Qp
158       qs0              = 200              !! Qs
159       topo0            = 0                !! topography location
160
161       !! ------------------------------------------------------------------- !!
162       !! For GMT grid file input 'grd' ( requires netcdf library )
163       !!
164       dir_grd          = '${DATASET}/vmodel/ejivsm'     !! directory for grd file
165       fn_grdlst        = './example/grd.lst'            !! grd file list
166       node_grd         = 0                              !! input MPI node
167
168       !! ------------------------------------------------------------------- !!
169       !! For layered homogeneous medium model ('lhm')
170       !!
171       fn_lhm           = 'example/lhm.dat'    !! 1D velocity structure
172
```

```
    !! ------------------------------------------------------------------- !!
    !! For random medium models
    !!
    dir_rmed         = './in/'             !! location of random medium file
    fn_grdlst_rmed   = './example/grd.lst' !! grd file list
    rhomin           = 1.0                 !! minimum density threshold

  !! ------------------------------------------------------------------- !!
  !! Checkpoint/Restart
  !!
  is_ckp           = .false.          !! perform checkpoint/restart
  ckpdir           = './out/ckp'      !! output directory
  ckp_interval     = 1000000          !! interval for checkpoint check (1/cycle)
  ckp_time         = 1000000.         !! checkpoint time
  ckp_seq          = .true.           !! sequential output mode

  !! ------------------------------------------------------------------- !!
  !! Reciprocity Green's Function Mode
  !!
  green_mode       = .false.          !! reciprocity Green's function mode
  green_stnm       = 'st01'           !! virtual station name from fn_stlst
  green_cmp        = 'z'              !! virtual source direction 'x', 'y', 'z'
  green_trise      = 1.0              !! rise time
  green_bforce     = .false.          !! also calc. body force Green's function
  green_maxdist    = 550.             !! horizontal limit of source grid
  green_fmt        = 'llz'            !! list file format: 'xyz' or 'llz'
  fn_glst          = 'example/green.lst'   !! Green's function grid point list

  !! ------------------------------------------------------------------- !!
  !! MISC
  !!

  stopwatch_mode   = .true.           !! measure computation time at routines
  benchmark_mode   = .true.           !! benchmark mode

  ipad             = 0                !! memory padding size for tuning
  jpad             = 0                !! memory padding size for tuning
  kpad             = 0                !! memory padding size for tuning
```

2.3 Controlling Parameters

`title`

Title of the computation. It will be used for output filename.

`odir`

Name of output directory. This is a relative directory path from the location of program execution. If this directory does not exist at the time of run, `OpenSWPC` will automatically create it.

`ntdec_r`

Number of **T**ime-step **DEC**imation factor for screen **R**eporting. Maximum amplitude of velocity at components are reported to the standard error output at every `ntdec_r`

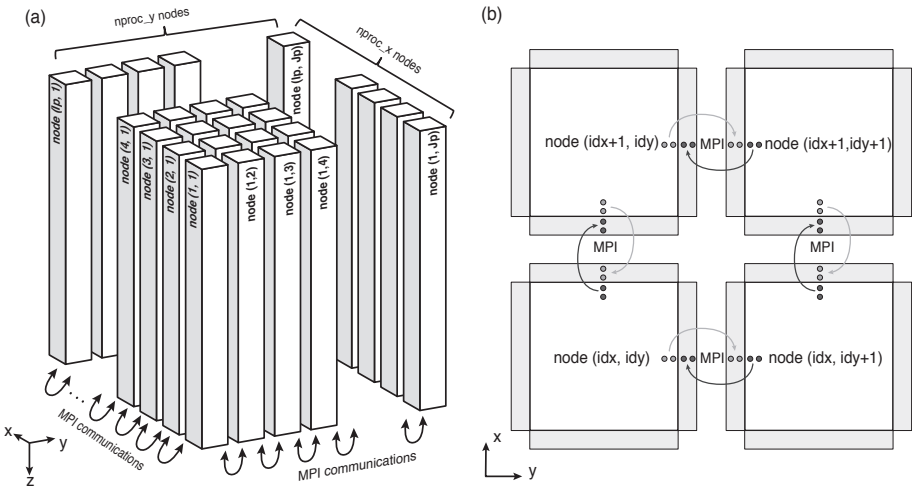

Fig. 2.1 (a) Partitioning of computational domain by for MPI (b) Schematic illustra- tion of data exchange by MPI protocol (modified from *Maeda et al.*, 2013).

steps. This screen output is usually used to confirm it is working correctly. Too short cycle (too small this parameter) may lead the slow-down of the computation.

2.4 Coordinate System and Parallel Computation

For parallel computation, `OpenSWPC` performs 2D model partitioning for 3D code (figure 2.1), and 1D partitioning for 2D code, in horizontal directions in either cases. The computation is performed in the Cartesian coordinate. We adopt the computational coordinate system depicted in figure 2.2. On default, coordinate axes x, y, z represent the north-, east- and depth directions, respectively. They cover the region `xbeg` $\le x \le$ `xend`, `ybeg` $\le$ y $\le$ `yend`, `zbeg` $\le z \le$ `zend`. We note that the z axis is defined to be positive downward. Since he free surface is usually defined at $z = 0$, it is recommended to take `zbeg` negative value to include $z = 0$ in the model.

This volume is discretized into `nx`, `ny`, and `nz` grids with spatial grid widths of `dx`, `dy`, `dz` along each direction. The parameter file must provide definitions of `xbeg`, `ybeg`, `zbeg`, and `nx`, `ny`, `nz`; other parameters (`xend`, `yend`, `zend`) are automatically computed by them. The center of the Cartesian coordinate ($x = 0$, $y = 0$) corresponds to the center longitude (`clon`) and latitude `clat`. The geographical coordinate is projected to the Cartesian coordinate by the Gauss–Krüger transform as follows (see Fig. 2.2) :

1. First generate evenly-spaced grid in the Cartesian Coordinate from input parameters of `phi` and that related to `x, y` coordinates.
2. Project grid location to the geographical coordinate by the Gauss–Krüger transform with center location of (`clon`, `clat`).
3. Obtain medium parameter at the grid location by bicubic interpolation of input velocity structure model.

If the specified area exceeds from the input velocity model, the outermost value of the velocity

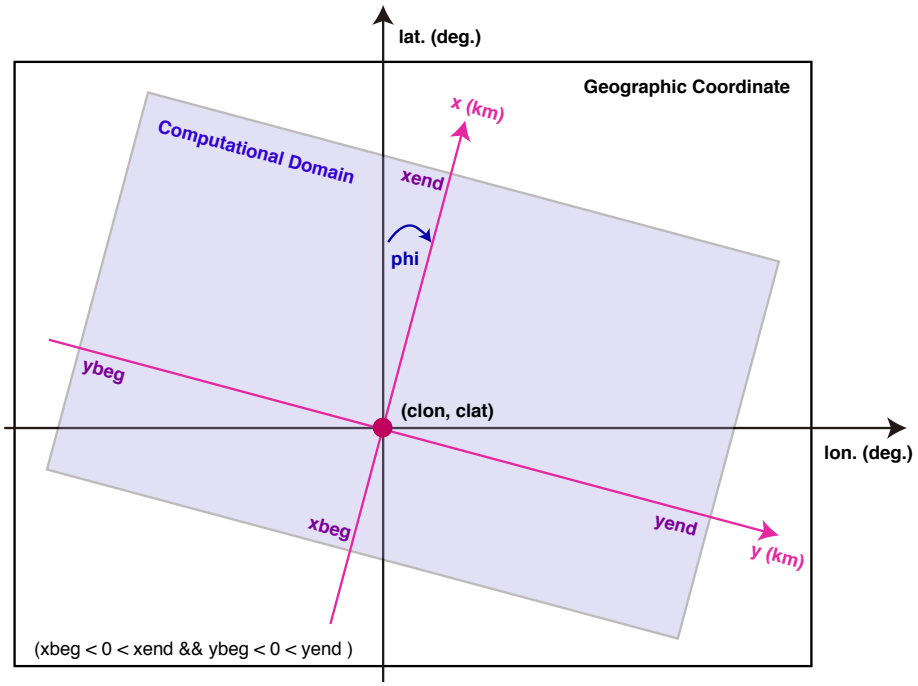

Fig. 2.2 Relation between computation coordinate and geographical coordinate systems.

structure is used for extrapolation.

2.4.1 Staggered Grid

`OpenSWPC` adopts the staggered-grid coordinate system shown in Figure 2.3. The unit volume shown in the Figure 2.3 is defined to as "voxel" at grid indices (`I,J,K`). A grid location x belongs to voxel number of

$$I = \left\lceil \frac{x - x_{\rm beg}}{\Delta x} \right\rceil, \tag{2.1}$$

and if voxel number `I` is given, its center coordinate location is veby by

$$x = x_{\rm beg} + \left(I - \frac{1}{2} \right) \Delta x, \tag{2.2}$$

where $\lceil \cdot \rceil$ is a ceiling function, $x_{\rm beg}$ is the minimum value of the x-coordinate. We note that the $x_{\rm beg}$ is set to belong to voxel `I=1`.

A voxel has a volume at

$$\begin{aligned} x_{\rm beg} + (I-1)\Delta x < x \le x_{\rm beg} + I\Delta x, \\ y_{\rm beg} + (J-1)\Delta y < y \le y_{\rm beg} + J\Delta y, \\ z_{\rm beg} + (K-1)\Delta z < z \le z_{\rm beg} + K\Delta z, \end{aligned} \tag{2.3}$$

The normal stress tensor components are denied at the of the voxel, shear stress on the edge, and velocity vector components on the surface of it (Figure 2.3). Medium parameters are

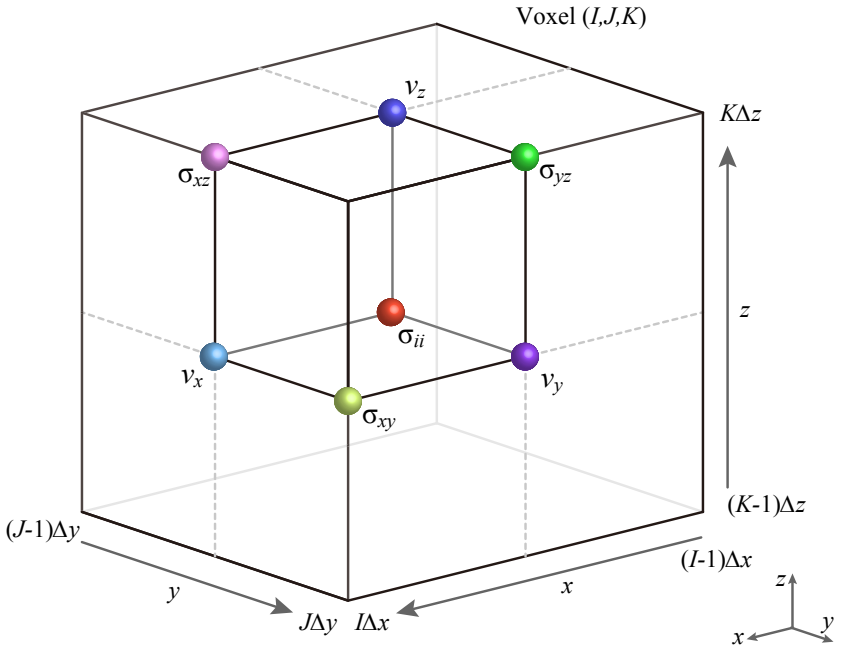

Fig. 2.3 Staggered grid layout in 3D space for the case of `xbeg=ybeg=zbeg=0`.

defined at the center of the voxel at

$$\begin{aligned} &x_{\text{beg}} + (I - 1/2)\Delta x, \\ &y_{\text{beg}} + (J - 1/2)\Delta y, \\ &z_{\text{beg}} + (K - 1/2)\Delta z. \end{aligned} \tag{2.4}$$

If necessary, averaging will be performed among neighbor voxels.

The spatial grid width Δx, Δy, Δz and the time step widthΔt must satisfy the stability condition. The stability condition in the N_D-dimension space for the order of finite difference method P is given by

$$\Delta t < \frac{1}{V_{\max}} \left(\sum_{i=1}^{N_D} \frac{1}{\Delta x_i^2} \right)^{-1/2} \left(\sum_{p=1}^{P/2} C_p \right)^{-1}, \tag{2.5}$$

where $V_{\max}$ is the maximum velocity of the medium, C_p is the coefficients of the finite difference formula, Δx_i is the spatial grid width along the i-th direction, respectively. For the fourth order formula of the finite difference which is used in the code, the coefficients takes $C_1 = 9/8$, $C_2 = 1/24$. For example, fourth order finite difference with isotropic grid sizes ($\Delta x = \Delta y = \Delta z = h$) in the three-dimensional space, the stability condition is reduced to

$$\Delta t < \frac{6}{7} \frac{1}{V_{\max} \sqrt{\dfrac{1}{\Delta x^2} + \dfrac{1}{\Delta y^2} + \dfrac{1}{\Delta z^2}}} = \frac{6h}{7\sqrt{3} V_{\max}} \simeq 0.495 \frac{h}{V_{\max}}. \tag{2.6}$$

This condition can be interpreted as "A distance that seismic wave propagates within a single time step must be much smaller than the spatial grid width". The numerical simulation will be diverged immediately, in case if this condition is not satisfied.

Additionally, the minimum wavelength of simulated seismic waves should be much longer than the spatial grid width. If the wavelength becomes relatively smaller compared to this condition, a fictitious numerical dispersion appeared and it result inaccurate later phase. Usually, the wavelength is taken to be longer than 5–10 times of spatial grid width to avoid this effect. For this purpose, the minimum velocity (usually S-wave velocity) in the velocity model should be carefully selected. One may take the smaller spatial grid size to avoid it, however in this case time-step size must be shorten as well to satisfy the stability condition.

`nproc_x, nproc_y`
: Number of partitions along x- and y- directions (Figure 2.1). Total number of partitions will be `nproc_x` × `nproc_y` for 3D case, and `nproc_x` for 2D case. This total number of parallelization must be equal to the number of processes given as an option of `mpirun`. These numbers can be 1. If `nproc_x=nproc_y=1`, this will become serial (non-parallel) computation in practice.

`nx, ny, nz`
: Total number of spatial grids in each direction. The `nx` and `ny` are not necessary to multiples of `nproc_x` and `nproc_y`.

`dx, dy, dz`
: Spatial grid width in each direction in km-unit. The total computational size in the physical domain will be `nx`×`dx`, `ny`×`dy` and `nz`×`dz`. The grid widths in different directions are not necessary to be equal.

`nt`
: Number of time steps.

`dt`
: Width of the time step in second. Total (physical) time of the simulation will be `nt`×`dt`.

`xbeg, ybeg, zbeg`
: Minimum value of the coordinates. If specification of `xbeg` or `ybeg` are omitted, they will automatically be set to as `xbeg` = – `nx` × `dx` / `2`, `ybeg` = – `ny` × `dy` / `2`. This setting is recommended to minimize distortion due to the map projection. The default value of `zbeg` is `-30`×`dz`.

`tbeg`
: Stating time. Usually it is set to zero.

`clon, clat`
: Center longitude and latitude in degrees. The map projection will be performed with this location as a reference point.

`phi`
: Horizontal rotation angle of the computational coordinate (see Figure 2.2). If `phi=0`, x-y-axis corresponds to the north and east directions, respectively. Note that the output files (snapshot and waveform) will be rotated if this value is nonzero.

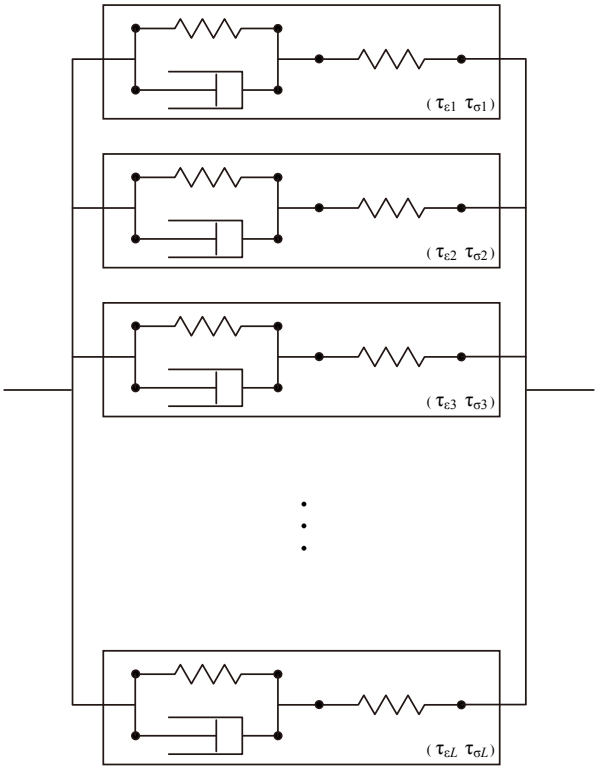

Fig. 2.4 Generalized Zener Body

2.5 Viscoelastic Body

`OpenSWPC` adopt the Generalized Zener Body (GZB) as a model of viscoelastic body. It consists of several viscoelastic Zener Bodies having different relaxation time to attain nearly-constant-Q in wide frequency range. As a consequence, it accompanies with the frequency-dependent body wavespeed by physical dispersion (e.g., *Aki and Richards*, 2002); thus the users should specify the reference frequency in which the velocity model is given.

The GZB consists fo N_M Zener bodies as schematically shown in Figure 2.4. This viscoelastic body is attributed by relaxation functions for elastic modlui $\pi \equiv \lambda + \mu$ and μ as

$$\begin{aligned} \psi_\pi(t) &= \pi_R \left(1 - \frac{1}{N_M} \sum_{m=1}^{N_M} \left(1 - \frac{\tau_m^{\varepsilon P}}{\tau_m^\sigma} \right) e^{-t/\tau_m^\sigma} \right) H(t) \\ \psi_\mu(t) &= \mu_R \left(1 - \frac{1}{N_M} \sum_{m=1}^{N_M} \left(1 - \frac{\tau_m^{\varepsilon S}}{\tau_m^\sigma} \right) e^{-t/\tau_m^\sigma} \right) H(t), \end{aligned} \tag{2.7}$$

where τ_m^σ is a relaxation time of m-th body, $\pi_R \equiv \lambda_R + 2\mu_R$ and μ_R are relaxed moduli, $\tau_m^{\varepsilon P}$ and $\tau_m^{\varepsilon S}$ are creep times of P- and S-waves, respectively. The wide-frequency-range of constant Q is achieved by connecting Zener bodies having different relaxation times. In addition, intrinsic attenuations of P- and S-waves (Q_P and Q_S) can be defined independently by choosing different creep times between elastic moduli π and μ.

The constitutive equation between stress and strain (or particle velocity) is written as

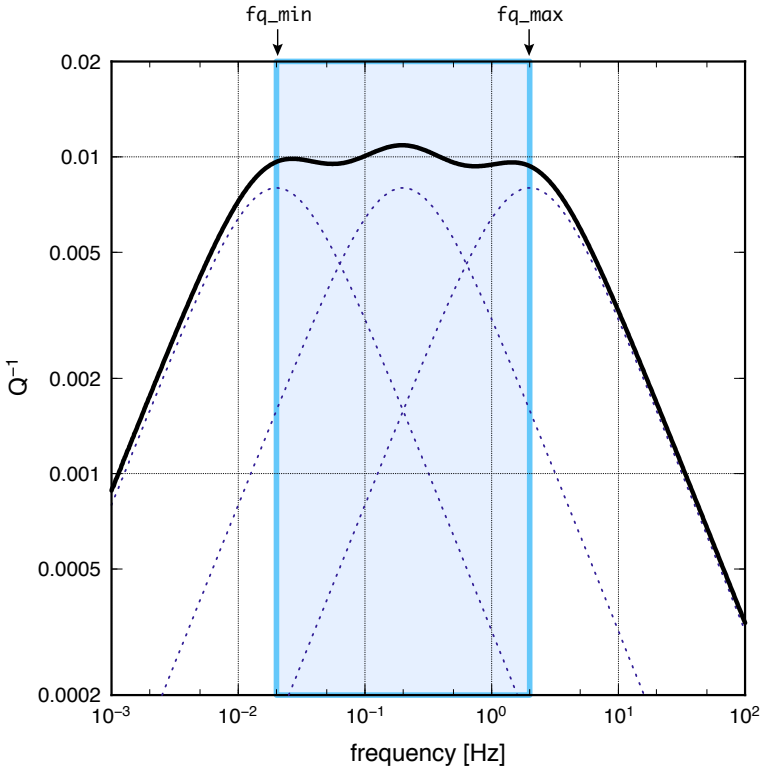

Fig. 2.5 Example frequency dependence of intrinsic attenuation Q^{-1} for Generalized Zener Body of `NM=3`. Thick solid line shows the attenuation of the whole model, while dotted lines show the attenuation model by each constituents Zener body. Vertical Lines show specified minimum and maximum frequency for the constant-Q range.

follows.

$$\begin{aligned}\dot{\sigma}_{ii}(t) &= \left(\dot{\psi}_{\pi}(t) - 2\dot{\psi}_{\mu}(t)\right) * \partial_k v_k(t) + 2\dot{\psi}_{\mu}(t) * \partial_i v_i(t) \qquad \text{(Do not take sum for } i\text{)} \\ \dot{\sigma}_{ij}(t) &= \dot{\psi}_{\mu}(t) * \left(\partial_i v_j(t) + \partial_j v_i(t)\right).\end{aligned} \tag{2.8}$$

The convolution appeared on the constitutive equation can be avoided by defining memory variables (*Robertsson et al.*, 1994) and solve auxiliary differential equations for them. We also adopt the τ-method of *Blanch et al.* (1994) to choose the creep times automatically that achieve constant Q condition.

`fq_min`
: Minimum frequency of constant-Q model

`fq_max`
: Maximum frequency of constant-Q model

`fq_ref`
: Reference frequency at which the velocity model is given

The Q^{-1} value become nearly constant between frequencies `fq_min` and `fq_max` as shown in the Figure 2.5. In the outside of the band, attenuation become weaker with increasing/decreasing frequency. As shown in this figure, the nearly-constant Q^{-1} is achieved by three different viscoelastic bodies. If one need to make Q^{-1} constant in wider frequency range,

Table 2.1 SAC headers automatically set by OpenSWPC

header name	description
kevnm	title of the parameter file
evlo,evla,evdp	The location of event (in degrees for horizontal, in m for depth)
o	Origin time of the event listed in the first line of the source list
kzdate,kztime	Date and time of execution of the simulation code
b	tbeg of the parameter file
delta	ntdec_w $\times$ dt
mag	The moment magnitude converted from the seismic moment
user0,..,user5	Moment tensor ($m_{xx}, m_{yy}, m_{zz}, m_{yz}, m_{xz}, m_{xy}$) of the first line of the source
user6,user7,user8	clon, clat, phi of the parameter file
kstnm	stnm of the parameter file
stlo,stla,stdp	Station location (in degrees for horizontal, in m for depth)
kcmpnm	Vx, Vy, Vz for velocities, or Ux, Uy, Uz for displacements
cmpinc,cmpaz	Station directions according to the Coordinate specification.
idep	7 for velocity, 7 for displacement

the hard-coded parameter NM should be increased. However, it leads significant increase of memory usage and computational loads. Frequency dependence of Q^{-1} with specified parameters above can be investigated by a program qmodel_tau.x (see section 3.3.1).

2.6 Simulation Data Output

2.6.1 Output Datafile Format

OpenSWPC can exports two types of data: spatio-temporal snapshot and seismic waveform at stations.

For snapshot files, the use may choose from an originally defined binary format or NetCDF file (recommended). The waveforms are usually exported in SAC format. The endian conversion is not performed at the time of data output. Even though, the official libraries of NetCDF and SAC automatically investigate the endian and convert them automatically if necessary. Thus, users not have to take care of the differences of endian among machines.

There is a utility program to read original-formatted data. We note taht the binary format may have slight differences among versions of OpenSWPC. Since the format change is tracked, the backward compatibility is always assured. It is recommended to use the utility of same version of simulation code. For SAC files, header components described in the Table 2.1 are automatically set. The units of SAC files are nm/s for velocity, and nm for displacement, following to the standard of the SAC. Though the earthquake source may be represented by multiple point sources, the header always represent a source listed in the first line of the source input file.

Snapshot file contains the header information listed in the Table 2.2. These headers are commonly defined either original format or NetCDF.

Table 2.2 Snapshot headers set by OpenSWPC

var	type	description
bintype	character(8)	Fixed to "STREAMIO"
codetype	character(8)	"SWPC_3D" or "SWPC_PV" or "SWPC_SH" depending on the code
hdrver	integer	Header version
title	character(80)	title in the parameter file.
exedate	integer	Date and time of the execution in the POSIX time
coordinate	character(2)	Snapshot cross section: 'xy', 'xz', 'yz', 'fs', or 'ob'.
datatype	character(2)	Data type: 'ps', 'v2' or 'v3'
ns1,ns2	integer	Number of data samples along first and second axis
beg1,beg2	real	Coordinate value at the first data point of axes
ds1,ds2	real	Snapshot grid spacing
dt	real	Time step width of the snapshot
na1,na2	real	Grid numbers of absorbing boundary layer in the snapshot
nmed	integer	Number of stored medium parameters
nsnp	integer	Number of snapshots per one time step
clon,clat	real	clon, clat in the parameter file
v1,v2,v3	real	Currently not being used

For the NetCDF, these headers are set as global attributes. The other headers are set following to the COARDS Conventions [*1] and the CF Convention [*2].

Note that horizontal directions of snapshot and waveforms are same with the Coordinate of computation. The x- and y-axis correspond to the north and east directions only if . phi=0. For waveform, this angle phi is stored on the cmpaz header. The vertical-component waveform is defined positive upward.

2.6.2 Snapshot Data Output

Spatio-temporal snapshot output may be created along cross sections of xy, yz, xz profiles and/or on the topography (fs) and/or bathymetry (ob). There are three types of snapshots; divergence and rotation of the velocity (ps), velocity (v) and displacement (u).

The use of spatial and temporal decimations are recommended to reduce the I/O load and export data size. Decimation in time is specified by ntdec_s starting from it=0 (before starting compuatation). For space, the decimations are performed by factors of idec, jdec, and kdec. In space, OpenSWPC tries to export the center of decimation window as schematically shown in the Figure 2.6. The numbers of exporting grid in each MPI node are not necessary to be same among nodes. Amplitude of these snapshot points will be gathered to specific nodes (see the Table 2.3) and exported as single files.

snp_format

Datafile format of snapshot files. "native" (original binary format) or "netcdf".

[*1] http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html
[*2] http://cfconventions.org

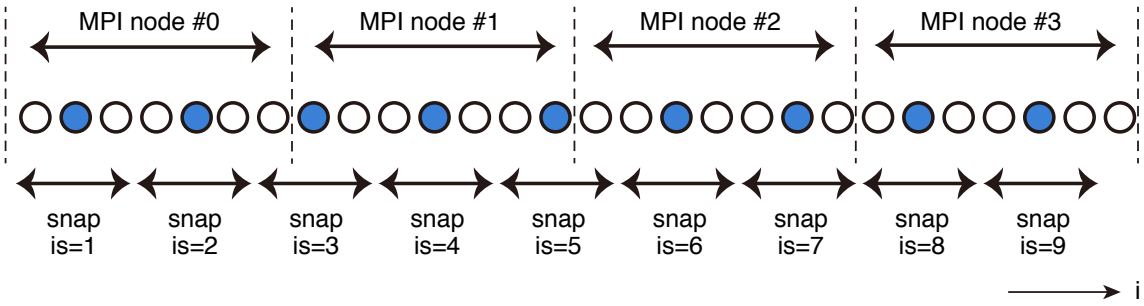

Fig. 2.6 Schematic illustration of spatial decimation for snapshot output. Vertical dotted lines show borders of MPI nodes. In this example, the data at grids colored by blue will be exported as snapshot data.

Table 2.3 MPI node number for exporting snapshot files.

section	type	node
yz	PS	0
xz	PS	mod(1, nproc)
xy	PS	mod(2, nproc)
fs	PS	mod(3, nproc)
ob	PS	mod(4, nproc)
yz	V	mod(5, nproc)
xz	V	mod(6, nproc)
xy	V	mod(7, nproc)
fs	V	mod(8, nproc)
ob	V	mod(9, nproc)
yz	U	mod(10, nproc)
xz	U	mod(11, nproc)
xy	U	mod(12, nproc)
fs	U	mod(13, nproc)
ob	U	mod(14, nproc)

Although the `NetCDF` file format is recommended for convenience on data handling, the use of this format may lead slight (~ 10 %) increase of computation time.

`xy_ps%sw, xz_ps%sw, yz_ps%sw, fs_ps%sw, ob_ps%sw`

Flags for exporting snapshot files of PS files (`.true.` or `.false.`). If they are set to be `.true.`, the divergence and rotation vector of the particle velocity will be exported.

`xy_v3%sw, xz_v3%sw, yz_v3%sw, fs_v3%sw, ob_v3%sw`

Flags for exporting snapshot files of velocities.

`xy_u3%sw, xz_u3%sw, yz_u3%sw, fs_u3%sw, ob_u3%sw`

Flags for exporting snapshot files of displacements.

Table 2.4 Format of the station location file

type	format				
'xy'	x	y	z	name	zsw
'll'	lon	lat	z	name	zsw

`z0_xy`
: Depth (km) of snapshot cross-section.

`x0_yz`
: X-coordinate value (km) of snapshot cross-section.

`y0_xz`
: Y-coordinate value (km) of snapshot cross-section.

`ntdec_s`
: Temporal decimation factor of the snapshot output. Snapshot will be exported at every `ntdec_s` time steps.

`idec, jdec, kdec`
: Spatial decimation factor of the snapshot output for x, y, and z directions.

2.6.3 Seismic Waveform Output

Seismic velocity and/or displacement records at specified stations can be obtained as SAC-formatted files by choosing parameters `sw_wav_v` and/or `sw_wav_u` to be `.true.`. Displacement record are calculated before the decimation, and thus it is expected to be more accurate than performing numerical integration to output velocity records. The traces are stored on memory during the computation, and exported at the end.

Station locations are given in the Cartesian coordinate (`xy`) or geographical coordinate (`ll`) as in the table 2.4. In the station list, lines starting from # will be ignored.

The depth of the station can be changed depending on the variable zsw in the station list as shown in table 2.5. This operation is important because the station at near the free surface occasionally locates above the approximated ground surface in the air due to the staircase approximation of the topography and bathymetry. Usually, it is recommended to specify `zsw=' obb'` ; this setting locates stations one-grid below the ground surface (or seafloor).

Multiple stations can be specified in the station list file. There is no fixed limit on number of stations. Number of stations are automatically counted, and only the station inside the computational region will be exported.

`sw_wav_v, sw_wav_u`
: Output velocity (`sw_wav_v`) and displacement (`sw_wav_v`) traces.

`ntdec_w`
: Decimation factor of waveform output. For `ntdec_w=1`, traces at every computational time step will be exported.

`st_format`
: Format of station list file. See table 2.4.

Table 2.5 Station depth specifications

zsw	Station depth setting
'dep'	Calculate station location from given station depth
'fsb'	One grid below of free surface (for ocean area, sea surface)
'obb'	One grid below of ocean bottom or ground surface (seafloor)
'oba'	One grid above of ocean bottom or ground surface (bottom of the sea)
'bdi' (i=0, ···, 9)	Internal velocity discontinuity specified by velocity model

Table 2.6 csf headers

header name	description
nvhdr	Format version numbers. Always zero.
ntrace	Number of traces in the file
npts	Number of time samples in the trace.

fn_stloc
: Station location filename.

wav_format
: Station file format. 'sac' (usual, recommended) or 'csf'

The csf format

Since the SAC format is defined to express data at one component of one station in a single file, the number of files may become extraordinary large. In this case, data transfer among computers will become quite inefficient. For the OpenSWPC version 3.0 or later, users can choose concatenated sac format (csf) for data output by specifying wav_format = 'csf'. This is a set of SAC binary files connected to single file, with headers as in the Table 2.6. The header is consisted of three of four-byte floating point numbers. After the header part, SAC-formatted trace records are repeated ntrace times. If this format was selected, the csf file are created at every computation node with node number in the filename for every components of the traces.

2.6.4 Output Filename Conventions

Output data names are determined as following rules:

- Snapshot (odir)/(title).(section).(type).snp
- Waveform (odir)/wav/(title).(stnm).(component).sac
- Computation time (odir)/wav/(title).tim

In the above rule, (section) takes cross section such as xy, yz. (type) takes v or ps depending on the snapshot data type. (component) takes Vx, Vy, Vz (for velocity) or Ux, Uy, Uz for displacement.

2.7 Velocity Model

2.7.1 Choice of Velocity Model Types

Users can choose velocity type from uniform (`uni`), layered homogeneous medium (`lhm`) and `NetCDF` (grd) file input (`grd`). In addition, randomly inhomogeneous medium calculated by an external program can be overlaid.

`vmodel_type`

Specify input velocity model. Choose from one of below.

`'uni'` Homogeneous medium with free surface. The following additional parameters are required:

`vp0` P-wave velocity [km/s] in the uniform model.
`vs0` S-wave velocity [km/s] in the uniform model.
`rho0` Mass density [g/cm^3] in the uniform model.
`qp0` Q_P of the uniform model.
`qs0` Q_S of the uniform model.
`topo0` Topography depth in the uniform model. If this value is larger than zero, seawater is filled from $z = 0$ to this depth.

`'lhm'`

Layered Homogeneous Medium. One-dimensional velocity structure file should be specified as below.

`fn_lhm`

Medium specification file. Every line specifies the depth of the top of layer, density, P-wave velocity, S-wave velocity, Q_P and Q_S below the depth. They must be separated by space(s) (see following example). Lines starting from # will be neglected.

```
# depth   rho(g/cm^3)   vp(km/s)    vs(km/s)      Qp       Qs
# ------------------------------------------------------------
      0         2.300       5.50       3.14      600      300
      3         2.400       6.00       3.55      600      300
     18         2.800       6.70       3.83      600      300
     33         3.200       7.80       4.46      600      300
    100         3.300       8.00       4.57      600      300
    225         3.400       8.40       4.80      600      300
    325         3.500       8.60       4.91      600      300
    425         3.700       9.30       5.31      600      300
```

`'grd'`

Velocity model input by `NetCDF` (GMT grd) files. The compilation of `OpenSWPC` should be done accompanied with using the `NetCDF` library. The following parameters are required.

`dir_grd`

Directory of velocity structure (`NetCDF`) files.

`fn_grdlst`

list file that specifies grd files and associated medium. Each line contains grd filename (with single or double quotation mark is recommended), density, P-wave velocity, S-wave velocity, Q_P, Q_S, and the layer number integers (0-9) separated by spaces (see following example). Lines starting from # will be neglected. Layer number is used for specifying source or station depth fit to the layer depth. The first `NetCDF` file will be treated as the ground surface. If the depth of the ground surface is deeper than zero, the depth range from $z = 0$ to the surface is assumed to be ocean layer. The grid above the free surface is treated to as the air column.

```
1  # grd filename          rho    vp    vs    QP    QS  sw
2  # ------------------------------------------------------
3  'eJIVSM_01_TAB_.grd'   1.80  1.70  0.35   119    70  0
4  'eJIVSM_02_BSM_.grd'   1.95  1.80  0.50   170   100  0
5  'eJIVSM_03_BSM_.grd'   2.00  2.00  0.60   204   120  0
6  'eJIVSM_04_BSM_.grd'   2.05  2.10  0.70   238   140  0
7  'eJIVSM_05_BSM_.grd'   2.07  2.20  0.80   272   160  0
8  'eJIVSM_06_BSM_.grd'   2.10  2.30  0.90   306   180  0
9  'eJIVSM_07_BSM_.grd'   2.15  2.40  1.00   340   200  0
10 'eJIVSM_08_BSM_.grd'   2.20  2.70  1.30   442   260  0
11 'eJIVSM_09_BSM_.grd'   2.25  3.00  1.50   510   300  0
12 'eJIVSM_10_BSM_.grd'   2.30  3.20  1.70   578   340  0
13 'eJIVSM_11_BSM_.grd'   2.35  3.50  2.00   680   400  0
14 'eJIVSM_12_BSM_.grd'   2.45  4.20  2.40   680   400  0
15 'eJIVSM_13_BSM_.grd'   2.60  5.00  2.90   680   400  0
16 'eJIVSM_14_BSM_.grd'   2.65  5.50  3.20   680   400  0
17 'eJIVSM_15_UPC_.grd'   2.70  5.80  3.40   680   400  0
18 'eJIVSM_16_LWC_.grd'   2.80  6.40  3.80   680   400  0
19 'eJIVSM_17_CTM_.grd'   3.20  7.50  4.50   850   500  0
20 'eJIVSM_18_PH2_.grd'   2.40  5.00  2.90   340   200  1
21 'eJIVSM_19_PH3_.grd'   2.90  6.80  4.00   510   300  0
22 'eJIVSM_20_PHM_.grd'   3.20  8.00  4.70   850   500  0
23 'eJIVSM_21_PA2_.grd'   2.60  5.40  2.80   340   200  2
24 'eJIVSM_22_PA3_.grd'   2.80  6.50  3.50   510   300  0
25 'eJIVSM_23_PAM_.grd'   3.40  8.10  4.60   850   500  0
```

`node_grd`

MPI node to input `NetCDF` data. All `NetCDF` files are first read by this node, and then they are transferred to all nodes by MPI data communication.

`is_ocean`

In default (`.true.`), from $z = 0$ to the topography will be treated as ocean layer. If this parameter is set to `.false.`, seafloor will be used as a free surface and no seawater will be used.

`vcut`

Cut-off velocity. For the models of `'lhm'` or `'grd'`, the velocity slower than this value will be overwritten by the `vcut` value. This parameter is used to avoid too short wavelength that violates wavelength condition (wavelength is recommended to be longer than 5-10 grids). This substitution will not be performed in the ocean area.

On Treatments of Air and Seawater Layer

In `OpenSWPC`, air column has mass density of $\rho = 0.001$ [g/cm^3], velocities of $V_P = V_S = 0$ [km/s], and intrinsic attenuation parameters of $Q^P = Q^S = 10^{10}$. In the ocean column, $\rho = 1.0$ [g/cm^3], $V_P = 1.5$ [km/s], $V_S = 0.0$ [km/s], $Q^P = Q^S = 10^6$ are assumed. The air column is practically treated as vacuum with no seismic wave propagation (with zero velocities). However, the mass density must not be zero to avoid division by zero. In the free surface and seafloor, the reduced order of the finite difference is performed according to (*Okamoto and Takenaka*, 2005; *Maeda and Furumura*, 2013). These discontinuities are automatically detected as boundaries which changes μ and λ from zero to a finite value.

2.7.2 Random Inhomogeneity

Users may overlay small-scale velocity inhomogeneity having specified power-law spectra to the background velocity models of `'uni'`, `'lhm'`, `'grd'`. The small-scale velocity inhomogeneity ξ is defined by external files. From the average velocity V_{P0}, V_{S0}, ρ_0, the fluctuated velocities and density are given as

$$\begin{aligned} V_P &= V_{P0}\,(1+\xi) \\ V_S &= V_{S0}\,(1+\xi) \\ \rho &= \rho_0\,(1+\nu\xi)\,, \end{aligned} \tag{2.9}$$

where $\nu = 0.8$ is a scaling parameter based on laboratory experiment (Birch's law; *Sato et al.*, 2012).

The velocity models having this small-scale inhomogeneity are specified by append `_rmed` to the original velocity models: `vmodel_type='uni_rmed'`, `'lhm_rmed'`, or `'grd_rmed'`. For random media generation, the readers are referred to the Section 3.4.1.

`dir_rmed`

A directory name for storing random media data files

The random media is given as two- or three-dimensional `NetCDF` files. In each grid locations, velocity fluctuation $\xi(I, J, K)$ is defined. The code automatically read the corresponding volume from the file; It is not necessary to decompose the `NetCDF` files into parts for parallel computation. If the computational size (`Nx,Ny,Nz`) is larger than the random media file size, the media is repeatedly used by applying circular boundary condition. The simulation codes do not care if the grid sizes of the simulation and the input random media file are identical.

Parameters for `uni_rmed`

The following parameter is required in addition to the parameters used in `vmodel='uni'`.

`fn_rmed0`

Name of random medium file.

In this model, the average velocity will be fluctuated based on the input `fn_rmed0`.

Parameters for `lhm_rmed`

In this model, the small-scale velocity fluctuation is applied to every layers defined by `vmodel='lhm'`. It is possible to assign different random velocity model at different layers.

The following parameter is substituted from `fn_lhm`:

`fn_lhm_rmed`

List file of velocity structure

The list file has similar format with the `fn_lhm`; It has filenames of the random media files at the rightmost column as the following example.

```
# depth   rho(g/cm^3)   vp(km/s)    vs(km/s)      Qp       Qs      fn_rmed
# ------------------------------------------------------------------------
      0         2.300       5.50        3.14      600      300      rmedia1.nc
      3         2.400       6.00        3.55      600      300      rmedia1.nc
     18         2.800       6.70        3.83      600      300      rmedia2.nc
     33         3.200       7.80        4.46      600      300      rmedia2.nc
    100         3.300       8.00        4.57      600      300      -
    225         3.400       8.40        4.80      600      300      -
    325         3.500       8.60        4.91      600      300      -
    425         3.700       9.30        5.31      600      300      -
```

In this example, the layers starting from depths of 0 km and 3 km has fluctuations defined by `rmedia1.nc`, layers from 18 km and 33 km has `rmedia2.nc`. The layer deeper than 100 km, a dummy filename (-) of random media is given. In this case (i.e., there are no files found), fluctuation will not be given. The dummy filename is mandatory for this case.

Parameters for `grd_rmed`

Overlay the random fluctuations to layers defined by the model of `vmodel='grd'`. It is possible to assign different random media at different layers. The stating depth of the velocity fluctuation can be either of free surface or depths defined by a layer.

The filename of the velocity fluctuation is given by the following parameter:

`fn_grdlst_rmed`

A list file that specifies velocity layer and random fluctuation files of each layer.

The list file takes two additional columns at rightmost; they are filename of random medium and reference layer number.

```
# grd  filename           rho     vp     vs    QP    QS  sw  fn_rmed       ref
# ------------------------------------------------------------------------
'eJIVSM_01_TAB_.grd'    1.80   1.70   0.35    119     70   0 'rmed3d_1.nc' 0
'eJIVSM_02_BSM_.grd'    1.95   1.80   0.50    170    100   0 'rmed3d_1.nc' 0
'eJIVSM_03_BSM_.grd'    2.00   2.00   0.60    204    120   0 'rmed3d_1.nc' 0
'eJIVSM_04_BSM_.grd'    2.05   2.10   0.70    238    140   0 'rmed3d_1.nc' 0
```

```
 7  'eJIVSM_05_BSM_.grd'   2.07  2.20  0.80   272   160  0 'rmed3d_1.nc' 0
 8  'eJIVSM_06_BSM_.grd'   2.10  2.30  0.90   306   180  0 'rmed3d_1.nc' 0
 9  'eJIVSM_07_BSM_.grd'   2.15  2.40  1.00   340   200  0 'rmed3d_1.nc' 0
10  'eJIVSM_08_BSM_.grd'   2.20  2.70  1.30   442   260  0 'rmed3d_1.nc' 0
11  'eJIVSM_09_BSM_.grd'   2.25  3.00  1.50   510   300  0 'rmed3d_1.nc' 0
12  'eJIVSM_10_BSM_.grd'   2.30  3.20  1.70   578   340  0 'rmed3d_1.nc' 0
13  'eJIVSM_11_BSM_.grd'   2.35  3.50  2.00   680   400  0 'rmed3d_1.nc' 0
14  'eJIVSM_12_BSM_.grd'   2.45  4.20  2.40   680   400  0 'rmed3d_1.nc' 0
15  'eJIVSM_13_BSM_.grd'   2.60  5.00  2.90   680   400  0 'rmed3d_1.nc' 0
16  'eJIVSM_14_BSM_.grd'   2.65  5.50  3.20   680   400  0 'rmed3d_1.nc' 0
17  'eJIVSM_15_UPC_.grd'   2.70  5.80  3.40   680   400  0 'rmed3d_1.nc' 0
18  'eJIVSM_16_LWC_.grd'   2.80  6.40  3.80   680   400  0 'rmed3d_3.nc' 0
19  'eJIVSM_17_CTM_.grd'   3.20  7.50  4.50   850   500  0 'rmed3d_3.nc' 0
20  'eJIVSM_18_PH2_.grd'   2.40  5.00  2.90   340   200  1 'rmed3d_2.nc' 18
21  'eJIVSM_19_PH3_.grd'   2.90  6.80  4.00   510   300  0 'rmed3d_2.nc' 18
22  'eJIVSM_20_PHM_.grd'   3.20  8.00  4.70   850   500  0 'rmed3d_3.nc' 18
23  'eJIVSM_21_PA2_.grd'   2.60  5.40  2.80   340   200  2 'rmed3d_2.nc' 21
24  'eJIVSM_22_PA3_.grd'   2.80  6.50  3.50   510   300  0 'rmed3d_2.nc' 21
25  'eJIVSM_23_PAM_.grd'   3.40  8.10  4.60   850   500  0 'rmed3d_3.nc' 21
```

The reference layer number defines the reference depth plane of the random media. If this number is zero, the depth grid number of the computation model is directly used to assign the random media. This is exactly same with the behavior of `uni_rmed` or `lhm_rmed` models. If the non-zero value of the reference layer number `NR` is specified, depth of the `NR`'s layer is treated as base plane. The depth grid of the random medium is measured from this depth. Introducing this reference plane, the inclined random media according to the velocity discontinuity (such as plate boundary) can be specified. In the above example, 18th and 21th layers are treated as reference of 18–20th and 21–23th layers.

Truncation of Velocity Fluctuations

If the magnitude of velocity fluctuation becomes too large, there can be a spot with non-physical velocity such as negative velocity or too high velocity as earth medium. The simulation may be unstable under the following conditions:

1. Fluctuated velocity $V = (1 + \xi)V_0$ exceeds the stability condition for the case of $\xi > 0$.
2. Velocity takes unrealistic negative values for the case of $\xi < -1.0$.
3. Mass density takes negative values for the case of $\xi < -1.25$.

To avoid such situations, `OpenSWPC` automatically limit the range of fluctuated velocity to `vcut`$\leq v \leq 0.95 \times v_{\max}$, where `vcut` is an input parameter and $v_{\max}$ is the possible maximum velocity derived from the stability condition.

In addition, the following parameter controls the minimum density.

`rhomin`

: Minimum mass density in g/cm^3. (1.0 g/cm^3 in default)

2.8 Earthquake Source Specification

2.8.1 Moment Rate Function

This section describes the moment rate functions $\dot{M}(t)$ that can be used in `OpenSWPC` by choosing parameter `stftype`. In the following, all moment rate functions have the duration (or characteristic time T_R), and are normalized so that the total moment will be 1.

Box-car function (`boxcar`)

$$\dot{m}^R(t) = \frac{1}{T_R} \qquad (0 \le t \le T_R) \tag{2.10}$$

Triangle function (`triangle`)

$$\dot{m}^T_R(t) = \begin{cases} 4t/T_R^2 & (0 \le t \le T_R/2) \\ -4(t - T_R)/T_R^2 & (T_R/2 < t \le T_R) \end{cases} \tag{2.11}$$

Herrmann function (`herrmann`)

$$\dot{m}^H(t) = \begin{cases} 16t^2/T_R^3 & (0 \le t \le T_R/4) \\ -2(8t^2 - 8tT_R + T_R^2)/T_R^3 & (T_R/4 < t \le 3T_R/4) \\ 16\,(t - T_R)^2\,/T_R^3 & (3T_R/4 < t \le T_R) \end{cases} \tag{2.12}$$

Cosine function (`cosine`)

$$\dot{m}^C(t) = \frac{1}{T_R}\left[1 - \cos\left(\frac{2\pi t}{T_R}\right)\right] \qquad (0 \le t \le T_R) \tag{2.13}$$

Küpper Wavelet (`kupper`)

$$\dot{m}^K(t) = \frac{3\pi}{4T_R}\sin^3\left(\frac{\pi t}{T_R}\right) \qquad (0 \le t \le T_R) \tag{2.14}$$

$t-$exp type (`texp`)

$$\dot{m}^E(t) = \frac{(2\pi)^2 t}{T_R^2}\exp\left[-\frac{2\pi t}{T_R}\right] \qquad (t \ge 0) \tag{2.15}$$

Figure 2.7 shows each moment rate functions and its Fourier spectrum. The moment rate functions have roll off of f^{-1} to f^{-4} at frequencies at $f \gg 1/T_R$. To avoid numerical dispersion, the source spectrum should be small enough at the highest target frequency. As this maximum frequency, here we adopt $f_{\max} = 2/T_R$ for all types of the source time functions (red dotted line in the Figure 2.7). If the parameter is appropriately set so that the numerical dispersion does not occur frequency below $f_{\max}$, the result should not be contaminated by the numerical dispersion problem. Besides, the uppermost frequency where the spectrum response of the source time function become flat in frequency domain is about $f \le 1/(2T_R)$ (blue dotted line in the Figure 2.7)

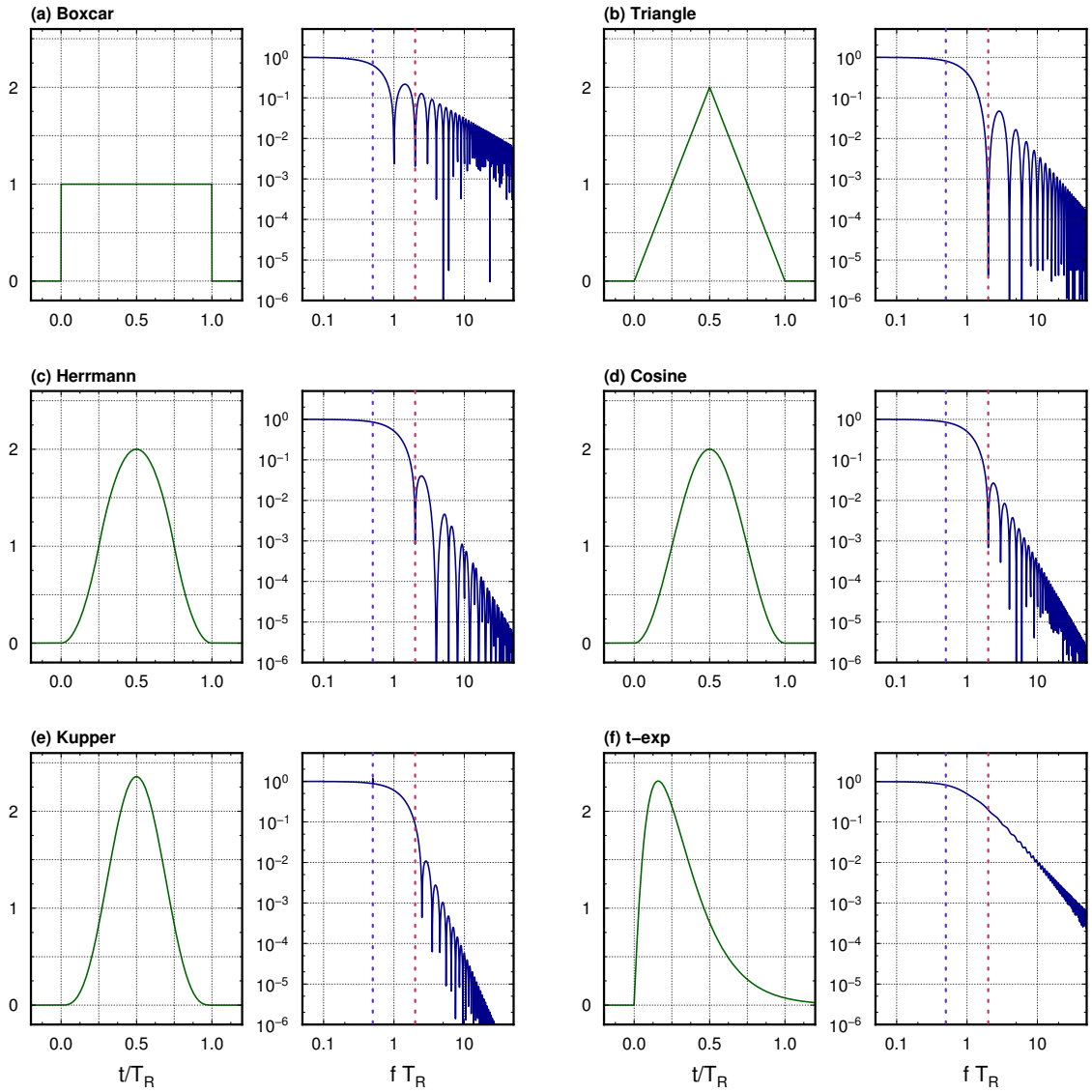

Fig. 2.7 Moment rate functions $\dot{m}(t)$ (left) and its Fourier spectrum (right).

2.8.2 Moment Tensor Source

Source mechanism of the faulting are given by 6-component moment tensor, or by three parameters of double couple source (`strike`, `dip`, `rake`). The source locations can be given either by computational coordinate or geographycal coordinate. Therefore, there are eight possible formats to describe source (see Table 2.7). In the program, sources are given as a stress-drop source by using moment rate function. The moment rate function is chosen from the given six functions (Figure 2.7). They require parameters for starting time T_0, duration T_R, and total moment M_0 in the source list file.

`OpenSWPC` can accept multiple point sources as multiple line in the source list file. There is

Table 2.7 Format of the source list file

Type	Format											
'xym0ij'	x	y	z	T_0	T_R	M_0	m_{xx}	m_{yy}	m_{zz}	m_{yz}	m_{xz}	m_{xy}
'xym0dc'	x	y	z	T_0	T_R	M_0	strike	dip	rake			
'llm0ij'	lon	lat	z	T_0	T_R	M_0	m_{xx}	m_{yy}	m_{zz}	m_{yz}	m_{xz}	m_{xy}
'llm0dc'	lon	lat	z	T_0	T_R	M_0	strike	dip	rake			
'xymwij'	x	y	z	T_0	T_R	M_W	m_{xx}	m_{yy}	m_{zz}	m_{yz}	m_{xz}	m_{xy}
'xymwdc'	x	y	z	T_0	T_R	M_W	strike	dip	rake			
'llmwij'	lon	lat	z	T_0	T_R	M_W	m_{xx}	m_{yy}	m_{zz}	m_{yz}	m_{xz}	m_{xy}
'llmwdc'	lon	lat	z	T_0	T_R	M_W	strike	dip	rake			

no fixed limit of the number of the source (practically it is determined by memory size). By gradually changing the starting time and source location, finite fault rupture can be mimicked. In the source list file, lines starting from # will be ignored. By setting `sdep_fit` the source depth can be changed so that it fit to the medium velocity boundary. In this case, the depth in the source list file will be ignored. The layer number should be specified in the list files `fn_grd` or `fn_grd_rmed`.

`stf_format`
: Format of source list file. Choose from `'xym0ij'`, `'llmwdc'` etc. See Table 2.7 for the complete list.

`stftype`
: Choice of the source time function. Select from `'boxcar'`, `'triangle'`, `'herrmann'`, `'kupper'`, `'cosine'`, and `'texp'`. See 2.7 for these functions.

`fn_stf`
: Filename of the source list.

`sdep_fit`
: Flag to fit the source depth to velocity discontinuity. `'asis'`: do not fit (default) `'bd{i}'`(i=1,2,$\cdots$9): fits to the `i`-th boundary specified by the rightmost column of the `fn_grdlst`.

2.8.3 Body Force Mode

Body force source can be used instead of the moment tensor source. In this mode, three-component force vector (f_x, f_y, f_z) should be specified. The force vector is assumed to have the bell-like shaped source time function as in the case of the moment tensor source. Although there is no restriction of numbers of body force elements, however it is not allowed to use both of moment tensor and body force at the same time.

`bf_mode`
: Flag for the body force mode. If this is `.true.`, the following parameters are used for

Table 2.8 File formats of the body force files

Type	Format							
'xy'	x	y	z	T_0	T_R	f_x	f_y	f_z
'll'	lon	lat	z	T_0	T_R	f_x	f_y	f_z

body force, and the moment tensor source will be ignored.

`stf_format`

Format of the source file. See the Table 2.8.

`stftype`

Choice of the source time function. Same with the case of the moment tensor source.

`fn_stf`

Filename of the source list file. Format is described on the Table 2.8.

`sdep_fit`

Flag to fit the source depth to specified velocity discontinuity. Same with the case of the moment tensor source.

2.8.4 Plane Wave Mode

Plane wave incidence from the bottom can be used as an input source instead from moment tensor or body force sources. In the `OpenSWPC`, the plane wave incidence is achieved by setting the velocity vector components and stress tensor components based on the analytic solution of plane wave propagating upward as initial conditions.

The specification of the initial conditions contains the depth of initial plane wave (`pw_ztop`) and its characteristic length (`pw_zlen`; corresponding to the wavelength), the strike and dip angle of the plane wave (`pw_strike`, `pw_dip`), polarization direction (rake angle) for the case of S-wave (`pw_rake`). See Figure 2.8 for the geometry. The definition of strike, dip, and rake parameters follow the definition of earthquake source fault geometry of *Aki and Richards* (2002). For the three-dimensional space, `pw_strike=0` results the plane dipping toward y-direction (East for `phi=0`). If the rake is `pw_rake=0`° or `pw_rake=180`°, it will be pure SH waves whose polarization is parallel to the free surface.

The initial plane wave occupies the depth range of `pw_zlen` (km) starting at depths from z =`pw_ztop` at the center of horizontal coordinate. The depth-dependence of the wave amplitude is determined by the source time functions used in the moment rate function, but as the function of space (Figure 2.8). By definition of the source time function, the integration of the initial plane wave along the propagation direction will be normalized to 1.

`pw_mode`

Flag to use the plane-wave mode. If it is `.true.` all point-source locations (body force or moment tensor source) will be ignored.

`pw_ztop`

z-value of the top of the initial plane wave at the $x = y = 0$.

`pw_zlen`

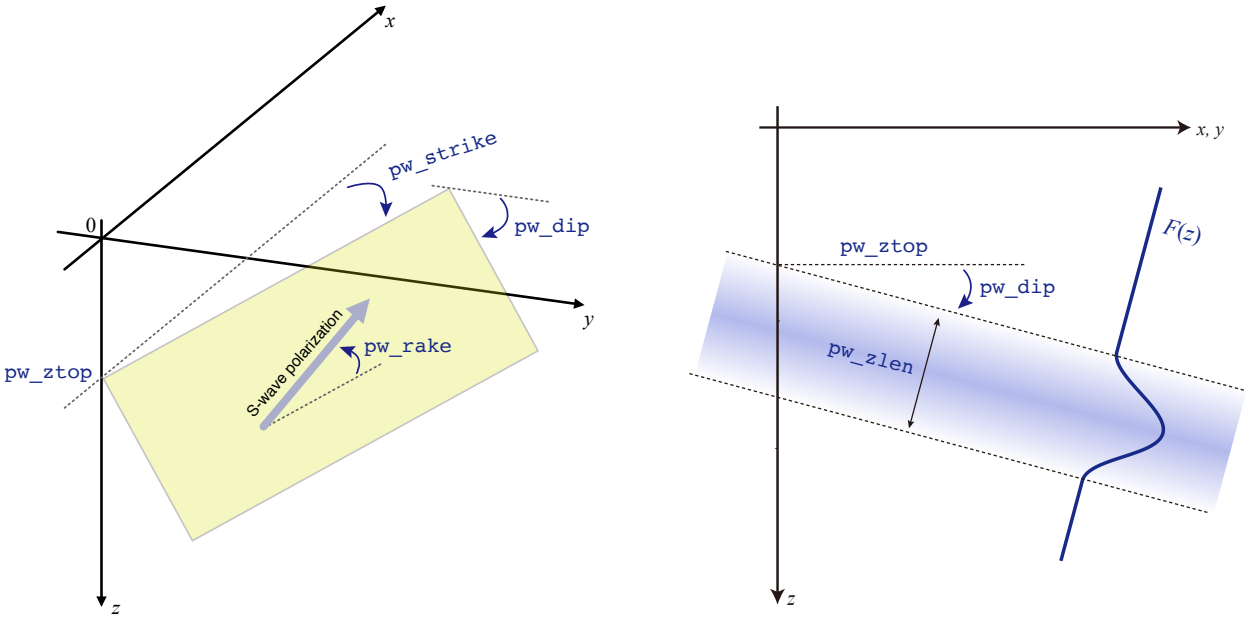

Fig. 2.8 Geometry of the plane wave specification. (Left) the specification of the uppermost plane and polarization direction. (Right) Depth cross-section of the initial plane wave.

Characteristic spatial scale of the initial plane wave.

`pw_ps`

Plane wave type. Choose from `'p'` or `'s'`

`pw_strike`

Strike direction of initial plane wave in degrees measured from the x-axis.

`pw_dip`

Dip angle of the initial plane wave in degrees. Initial plane wave propagates vertically if this angle is zero.

`pw_rake`

Polarization direction of initial plane S-wave in degrees measured from the horizontal plane.

`stftype`

Source time function type. Same with the cases of the moment tensor or body force source.

The use of the PML absorbing boundary condition (`abc_type='pml'`; see Section 2.9) is strongly recommended for the case of the plane wave incidence. The simple Cerjan's (`abc_type='cerjan'`) condition always cause significant contamination of the artificial reflections (Figure 2.9). Even by using the PML boundary, the tilted plane wave incidence (with nonzero `pw_dip` angle) cause some amount of artificial reflections. It is highly recommended that to confirm the boundary effect from snapshot visualization for using this plane wave mode.

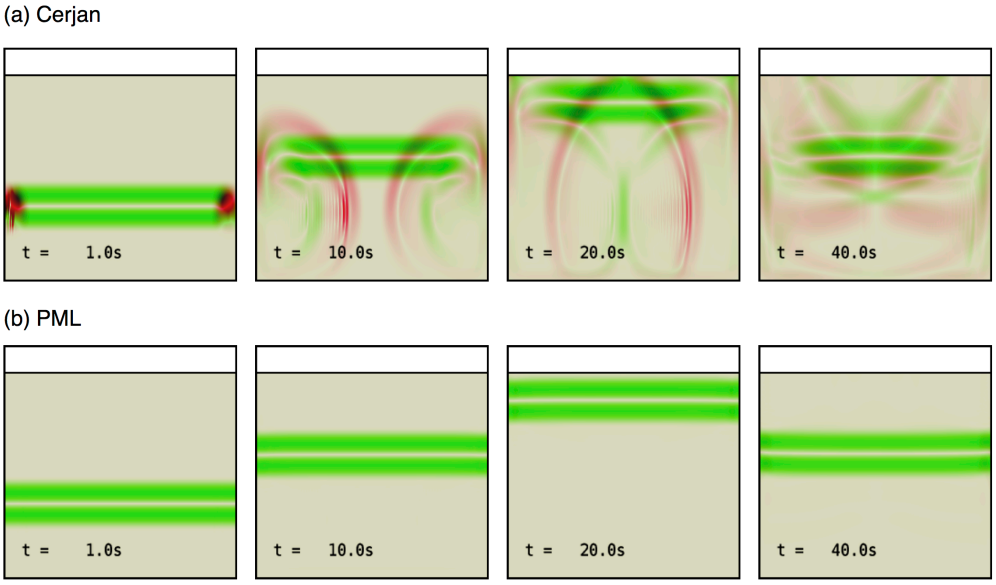

Fig. 2.9 Snapshots of absolute values of divergence (red) and rotation (green) for the case of vertical plane S-wave incidence with boundary conditions of (a) Cerjan's condition and (b) PML.

2.9 Absorbing Boundary Condition

Users can choose as an absorbing boundary condition from Auxiliary Differential Equation, Complex Frequency-Shifted Perfectly Matched Layer (ADE CFS-PML *Zhang and Shen*, 2010) and Cerjan's sponge condition (*Cerjan et al.*, 1985).

The whole computational domain is separated into the interior and exterior regions by the thickness of absorber `na` as shown in figure 2.10. Since this program assume the existence of free surface and ignores acoustic waves in the air column, the waves in the top boundary will not be absorbed. At given horizontal grid location `(I,J)`, the depth grid deeper than `kbeg_a` will be used as attenuator.

For PML boundary condition, the `OpenSWPC` does not solve the viscoelastic constitutive equation in the absorber for computational efficiency. Note that in the case of medium having very small Q values, it may leads velocity gap between interior and exterior regions due to physical dispersion.

For Cerjan's absorbing condition, the parameters suggested by *Cerjan et al.* (1985) is embedded in the source code. However, these parameters are scaled according to the width of the absorber `na`.

The PML absorber is usually much superior than the Cerjan's sponge in its efficiency on avoiding artificial reflection from boundaries. However, PML occasionally gives numerical instabilities, in particular for the medium having strong velocity contrast and after large number of time steps. In such case, Cerjan's sponge always gives very stable result.

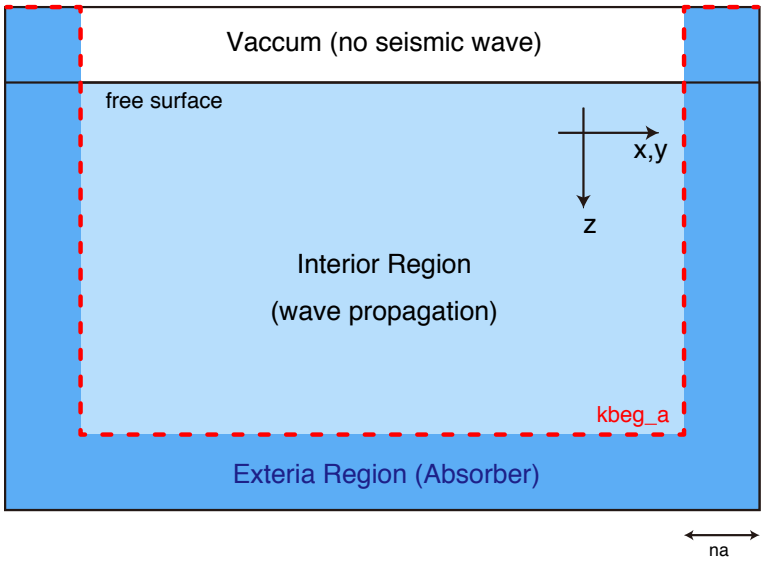

Fig. 2.10 A schematic illustration of the definition of absorber region. Red dotted line indicates the location of `kbeg_a(I,J)`.

`abc_type`
: Type of the absorbing boundary condition. Choose from `'pml'` or `'cerjan'`.

`na`
: Thickness of the absorbing layer in numbers of grids. Usually choose from 10–20.

`stabilize_pml`
: Remove low velocity layer if this flag is `.true.`, to stabilize the PML.

2.10 Checkpointing and Restarting

Some large-scale computers limit the computation time in a single job. To achieve long-time-duration computation, `OpenSWPC` can export all the memory contents to files at specific time (checkpointing), and then continue the simulation as another job (restarting.

If this function is turned on, `OpenSWPC` will terminate the computation after elapsed time of `ckp_time` (in seconds), with exporting all memory images.

In another job, `OpenSWPC` first tries to seek the directory `cdir` to find the checkpointing file. If there are checkpointing files, `OpenSWPC` read them to continue the simulation. Otherwise, `OpenSWPC` start simulation from the beginning.

After finishing the computation of all time step, `OpenSWPC` remove the most of the contents of the checkpointing file. However, it does not remove the checkpointing files. This is to avoid to unexpectedly start the computation from the beginning and to overwrite the output files.

This function is equipped only with the three-dimensional simulation code (`swpc_3d.x`).

`is_ckp`
: A flag to use the checkpointing/restarting.

`cdir`

Output directry name of the checkpointing file. At restarting, the checkpointing files are assumed to be in this directory.

`ckp_time`

Checkpointing time in seconds.

`ckp_interval`

Investigate if the computation time exceeds `ckp_time` at every this interval. To set this interval step too small may affect the perfomance of the computation.

`ckp_seq`

Sequential output mode. If this flag is `.true.`, the I/O of the checkpointing files are sequentially performed from the zero-th MPI node. If the file system is shared among large number of computational nodes, this flag may be effective to improve the I/O performance.

2.11 Reciprocity Mode

This mode excite the seismic wave at specified station location, and export velocity and/or strain velocity the multiple virtual source locations. Based on the reciprocity theorem, this result corresponds to the body force and/or moment tensor response from virtual source locations observed at specified station. If the time duration of the source time function is short enough, they can be treated as Green's functions.

If we denote the Green's tensor from the virtual source $\boldsymbol{\xi}$ to receiver $\boldsymbol{r}$ as $G_{ij}(\boldsymbol{r}, t; \boldsymbol{\xi})$, this mode simulates the convolution of spatial derivatives of Green's tensor convolved with the source time function $s(t)$ as

$$
\begin{aligned}
G_i^{M1}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv \frac{\partial G_{ix}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_x} * s(t) = \frac{\partial G_{ix}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_x} * s(t) \\
G_i^{M2}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv \frac{\partial G_{iy}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_y} * s(t) = \frac{\partial G_{iy}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_y} * s(t) \\
G_i^{M3}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv \frac{\partial G_{iz}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_z} * s(t) = \frac{\partial G_{iz}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_z} * s(t) \\
G_i^{M4}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv \left(\frac{\partial G_{iy}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_z} + \frac{\partial G_{iz}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_y} \right) * s(t) = \left(\frac{\partial G_{iy}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_z} + \frac{\partial G_{iz}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_y} \right) * s(t) \\
G_i^{M5}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv \left(\frac{\partial G_{ix}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_z} + \frac{\partial G_{iz}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_x} \right) * s(t) = \left(\frac{\partial G_{ix}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_z} + \frac{\partial G_{iz}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_x} \right) * s(t) \\
G_i^{M6}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv \left(\frac{\partial G_{ix}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_y} + \frac{\partial G_{iy}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_x} \right) * s(t) = \left(\frac{\partial G_{ix}(\boldsymbol{r}, t; \boldsymbol{\xi})}{\partial \xi_y} + \frac{\partial G_{iy}(\boldsymbol{\xi}, t; \boldsymbol{r})}{\partial \xi_x} \right) * s(t),
\end{aligned}
\tag{2.16}
$$

which corresponds to the moment tensor response. Optionally, the body-force response

$$
\begin{aligned}
G_i^{B1}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv G_{ix}(\boldsymbol{r}, t; \boldsymbol{\xi}) * s(t) = G_{ix}(\boldsymbol{\xi}, t; \boldsymbol{r}) * s(t) \\
G_i^{B2}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv G_{iy}(\boldsymbol{r}, t; \boldsymbol{\xi}) * s(t) = G_{iy}(\boldsymbol{\xi}, t; \boldsymbol{r}) * s(t) \\
G_i^{B3}(\boldsymbol{r}, t; \boldsymbol{\xi}) &\equiv G_{iz}(\boldsymbol{r}, t; \boldsymbol{\xi}) * s(t) = G_{iz}(\boldsymbol{\xi}, t; \boldsymbol{r}) * s(t)
\end{aligned}
\tag{2.17}
$$

can be calculated.

Table 2.9 Virtual source location format for the reciprocity mode.

Type	Format			
'xyz'	x	y	z	gid
'llz'	lon	lat	z	gid

To use this mode, the users should specify the station name green_stnm of receiver. This station name should be contained by the station list file. The OpenSWPC radiates the seismic wave by excitation force with a direction specified by the green_cmp parameter, with a source time function of rise time green_trise. To obtain full response of all components, three independent simulations with green_cmp='x', 'y', and 'z' are necessary.

Virtual source location should be given in the Cartesian coordinate or geographical coordinate and depth (the format is described in the Table 2.9), with unique integer ID numbers (gid). Multiple virtual source location can be specified in the simulation. The gids are not necessary to be sequential.

The output file is stored in the directory of (odir)/green/(gid) in the SAC format with the name convention of (title)__(green_cmp)__mij__.sac (for moment tensor response) or (title)__(green_cmp)__fi__.sac (for body force response).

The amplitudes of output files are multiplied by 10^9 to compare the SAC-formatted files in nm or nm/s units. The vertical component of the output file is changed to be positive upward. However, derivative with respect to depth is performed according to the original definition of positive downward.

green_mode
: Flags to turn the reciprocity mode on. If this is .true., the other earthquake source parameters will be ignored.

green_stnm
: Name of the virtual station. This name must be included in the station list.

green_cmp
: Component at the virtual receiver. Choose from 'x', 'y', or 'z'.

green_trise
: Rise time of the source time function convolved with the simulated Green's function.

green_bforce
: If .true., calculate body force response as well as the moment tensor response. Default setting is .false..

green_fmt
: Format specification of the virtual source location. Choose from 'xyz' (Cartesian coordinate; default) or 'llz' (longitude, latitude and depth).

green_maxdist
: Reciprocity wave will be calculated only if the horizontal distance is shorter than this parameter. Specify in km unit.

fn_glst
: Name of the virtual source location file.

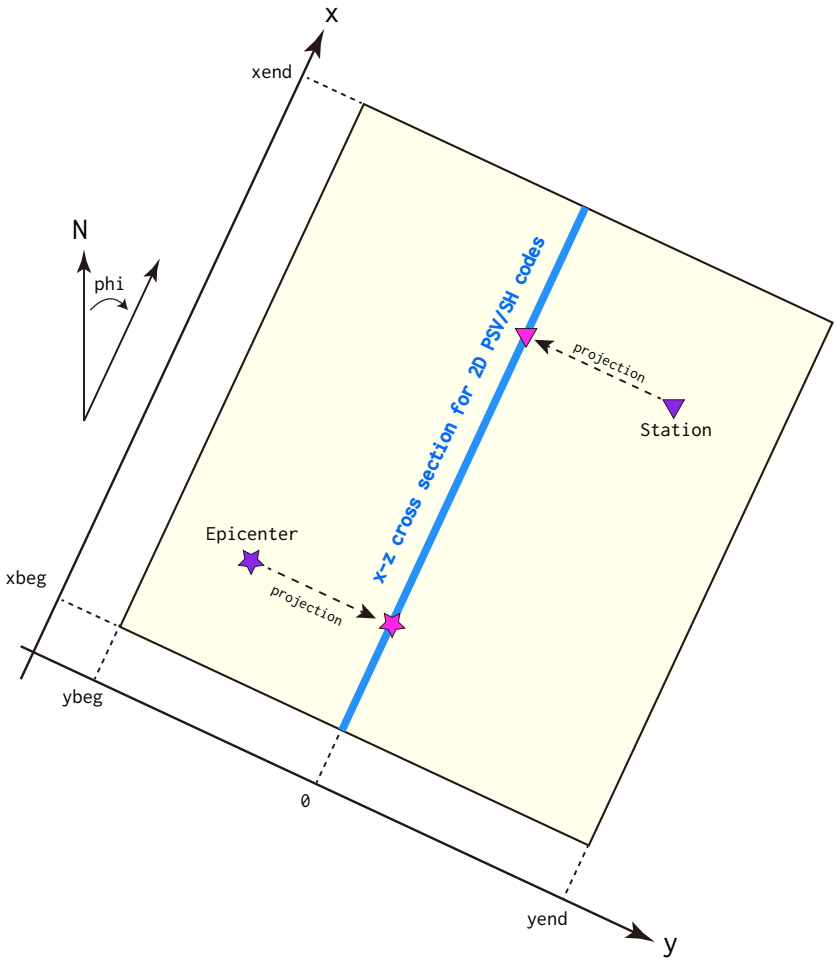

Fig. 2.11 Cross section for calculation in the 2D codes on the horizontal (x–y) plane. All stations and epicenters are projected onto the x–z cross section.

`stftype`
: Source time function type. Same with the case of the moment tensor source.

`ntdec_w`
: Temporal decimation factor of the output waveforms. Same with the case of the normal waveform output.

2.12 About Two-Dimensional Codes

`OpenSWPC` contains P-SV (`swpc_psv`) and SH (`swpc_sh`) codes. These codes work with the same parameter file. In these 2D codes, the simulation will be performed along the $x - z$ cross section of $y = 0$. The parameters related to the y-direction will be omitted. MPI partition therefore will be 1D, only along the x-direction. Notice that the all station and source outside the cross section will be projected onto the cross section as schematically shown in the Figure 2.11. For the plane wave incidence, `pw_strike`, `pw_rake` will be fixed according to the types of the code. Only the dip angle (`pw_dip`) is changeable.

2.13 Other Parameters

`stopwatch_mode`

Measure the computation times at major subroutines, and export the accumulated times to `(odir)/(title).tim`. This function is used for benchmarking and performance tuning.

`benchmark_mode`

If this flag is `.true.`, the fixed homogeneous medium and single point moment tensor source will be selected irrespective the parameter specification. This is used for validation and performance measurement.

`ipad, jpad, kpad`

Expand the Fortran array sizes along x-, y-, and z-directions. Some computer architecture, the computation speed is quite sensitive to the array size. In such case, slightly changing the array size by these parameters may contribute to improve the performance. The expanded array will not be used for simulation. Thus, the simulation result is not affected by changing this option.

Chapter 3

Related Tools

3.1 Snapshot data handling

3.1.1 `read_snp.x`

Snapshot file in both format of `NetCDF` and orignally-defined binary can be extracted or visualized by the program `read_snp.x`.

```
1  read_snp.x -i snapfile [-h] [-ppm|-bmp] [-pall] [-mul var | -mul1 var -
       mul2 var ...] [-bin] [-asc] [-skip n]
```

`-h`

Print the header information defined in the snapshot as the following example.

```
 1  > ../bin/read_snp.x -i swpc_3d.xz.ps.snp -h
 2
 3  [binary type]   : STREAMIO
 4  [code type]     : SWPC_3D
 5  [header version]:          3
 6  [title]         : swpc_3d
 7  [date generated]: 1408015126
 8                    2014-08-14T11-18-46
 9  [coordinate]    : xz
10  [data type]     : ps
11  [ns1]           :        256
12  [ns2]           :        256
13  [beg1]          :       -63.87500
14  [beg2]          :        -9.87500
15  [ds1]           :         0.25000
16  [ds2]           :         0.25000
17  [dt]            :         0.05000
18  [na1]           :         20
19  [na2]           :         20
20  [nmed]          :          3
21  [nsnp]          :          2
22  [clon]          :       143.50000
23  [clat]          :        42.00000
```

`-ppm/-bmp`

Visualize and export the image files in ppm or bmp format. The ppm or bmp directory will be automatically created under the current directory, and image files with sequential numbers will be stored in it. If the snapshot file is displacement or velocity, absolute values of vertical and horizontal amplitude will be colored by red and green. Forthe ps file, absolute values of divergence and rotation vector will be colored in the same manner. If the absolute value option is specified, the black-red-yellow-white color palette (similar to the "hot" color palette in GMT) will be adopted. For cross sections along the surface (ob, fs), the topography color map will be overlaid. For other cross-sections, velocity structure on the section will be overlaid.

`-pall`

Visualize including absorbing boundary region. This option works only if this is used together with `-ppm/-bmp`. In default, the absorbing boundary region will be clipped.

`-mul var|-mul1 var -mul2 var ...`

Multiply `var` to the amplitude for visualization. Adjust the visualized color by changing this values. Optionally by specifying `-mul1`, `-mul2` etc., one may change the weight of amplitude by compnents.

`-abs`

Visualize the absolute value of the vector. It only work with velocity or displacement snapshot.

`-bin|asc`

Export to the snapshot data to binary (`-bin`) or ascii (`-asc`) files. The data file will be created in the automatically created `bin` or `asc` directories. The binary formatted data can be directly used from `GMT` such as `xyz2grd` module by appending `-bis` option.

`-skip` *n*

Skip the first *n* snapshot for visualization or data exports.

3.1.2 `diff_snp.x`

This program takes the difference between two snapshot, and export it to the another snapshot file.

```
1  > diff_snp.x snap1 snap2 diffile
```

The output file format (`NetCDF` or binary) depends on the input file format.

3.2 Supporting Parameter Setting

3.2.1 `fdmcond.x`

The grid width in space and time in the finite difference method is controlled by the stability condition. The wavelength condition will affect the allowed maximum frequency radiated from the source.

A tool `fdmcond.x` will help to determine these parameters with satisfying the conditions.

By specifying several parameters from grid width, maximum frequency (`fmax`), rise time (`Tr`), minimum and maximum velocity in the medium (`vmin, vmax`), the program suggest the other parameters.

Example

```
1
2  >  ./fdmcond.x
3
4  ------------------------------------------------------------------------
5                              FDM CONDITION
6  ------------------------------------------------------------------------
7
8
9   Model Dimension ? --> 3
10    2) 2D
11    3) 3D
12
13
14  Source Type ? --> 3
15    1) Triangle
16    2) Herrmann
17    3) Kupper
18
19
20  Parameter Combination ? --> 5
21    1) dh   (space grid),  fmax (max freq.),  vmax (max vel.)
22    2) dh   (space grid),  Tr   (rise time),  vmax (max vel.)
23    3) dh   (space grid),  fmax (max freq.),  dt (time grid)
24    4) dh   (space grid),  Tr   (rise time),  dt (time grid)
25    5) dh   (space grid),  vmin (min vel.),   vmax (max vel.)
26    6) dh   (space grid),  vmin (min vel.),   dt (time grid)
27    7) fmax (max freq.) ,  vmax (max vel.),   dt (time grid)
28    8) Tr   (rise time) ,  vmax (max vel.),   dt (time grid)
29    9) vmin (min vel.)  ,  vmax (max vel.),   dt (time grid)
30
31
32  Assumed Parameters:
33    dx     =   0.25
34    dy     =   0.25
35    dz     =   0.25
36    vmin   =   0.3
37    vmax   =   8.0
38
39  Derivaed Parameters:
40    dt    <=   0.01546
41    fmax  <=   0.17143
42    Tr    >=  13.41667
```

3.2.2 mapregion.x

A geographical region of simulation will be automatically determined by the parameters such as `clon, clat, phi, xbeg, ybeg, nx, ny, dx, dy`. The program `mapregion.x` read the parameter file, and export the outer edge of the region in longitude and latitude.

```
1  > mapregion.x -i input.inf -o region.dat
```

If the option `-o` is omitted, the result will be printed to the standard output on the screen. This program also will estimate the total memory usage to the standard error output.

3.2.3 mapregion.gmt4, mapregion.gmt5

These scripts uses the `mapregion.x` to visualize the region by using GMT4 or GMT5. In default, these scripts plot only around Japanese Islands.

3.3 Velocity Structure

3.3.1 qmodel_tau.x

Calculate frequency dependence of Q^{-1} and body wave dispersion from the input parameter file.

```
1  > qmodel_tau.x -nm [nm] -i [prm_file] -f0 [min_freq] -f1 [max_freq] -nf [
       ngrid]
```

It discretize frequency range from `min_freq` to `max_freq` into `ngrid`, and export $Q^{-1}(f)$ and physical dispersion. The latter is normalized to 1 at the reference frequency. The parameters regarding to the viscoelastic body is read from the input parameter file, however the number of bodies `nm` should be specified separately, since it is hard-coded in the program.

3.3.2 grdsnp.x

From the input parameter file, calculate and print the discontinuity of input `NetCDF` file in the Cartesian coordinate for simulation (x, y, depth) to the standard output. This program is used to confirm coordinate transformation, the detailed digital model, and to visualize the model in the computational domain.

```
1  > grdsnp.x -i [prm_file] -g [grd_file]
```

3.4 Generation of Random Media

3.4.1 gen_rmed3d.x

Generate three-dimensional random medium file.

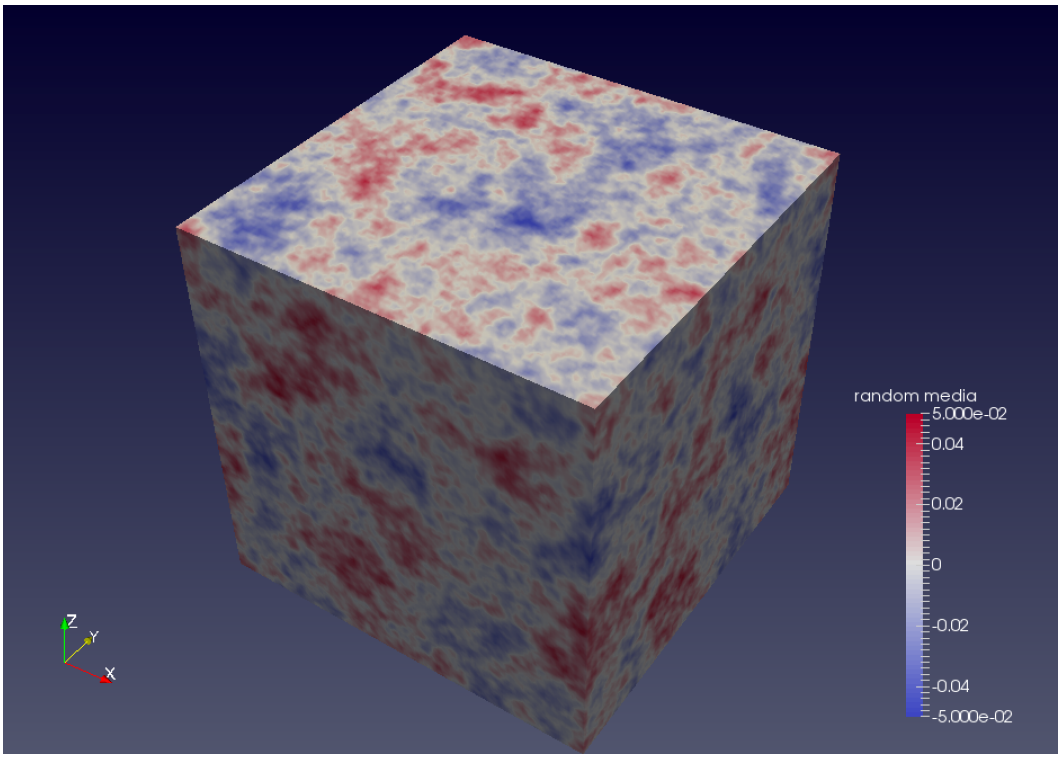

Fig. 3.1 Example of visualization of 3D random medium by `ParaView` software

```
1  gen_rmed3d.x  [-o outfile] [-nx nx] [-ny ny] [-nz nz] [-epsil epsilon] [-
       kappa kappa]  [-dx dx] [-dy dy] [-dz dz] [-ax ax] [-ay ay] [-az az]  [-
       ptype ptype] {-seed seed_number}
```

`-o outfile`
: Name of the output file

`-nx nx -ny ny -nz nz`
: Numbers of grid in x-, y-, and z-directions. They must be the power of 2.

`epsil epsilon`
: Root Meen Square (RMS) of the velocity fluctuation ε.

`-ax ax -ay ay -az az`
: Characteristic scale in x-, y-, and z-directions in km unit.

`-dx dx -dy dy -dz dz`
: Grid width in x-, y-, and z-directions. They should be identical with the simulation parameters.

`-ptype ptype`
: Choice of the power spectrum density functions (PSDFs) of random media model in wavenumber space. 1:Gaussian, 2:Exponential, 3:von Kármán

`-kappa kappa`
: A parameter κ for teh von Kármán-type PSDF.

`-seed seed_number`
: Specify the seed number of the random variable generation (Optional). If this option is

not specified, the seed number is automatically generated based on the execution date and time.

The random media file will be stored in the `NetCDF` format. Various software such as texttttParaView[*1] (Figure 3.1) and `Panoply`[*2] can be used for visualization.

3.4.2 `gen_rmed2d.x`

Generate 2D random media file. The usage is same with `gen_rmed3d.x`, with parameters regarding to the y-direction are omitted.

3.5 Miscellaneous Tools

3.5.1 timvis scripts

Four scripts, `timvis.gmt4`, `/timvis.gmt5`, `timvis_abs.gmt4`, `/timvis_abs.gmt5` are used for visualization of elapsed time of computation obtained with the input parameter `stopwatch_mode = .true.` by using GMTs version 4 and 5.

3.5.2 Geographic Coordinate Conversion

Fortran programs `ll2xy.xxy2ll.x` will project and inversely-project between the geographic coordinate and the Cartesian coordinate with the same algorithm with the `OpenSWPC`. These tools are provided after `OpenSWPC` version 3.0 or later.

[*1] http://www.paraview.org
[*2] http://www.giss.nasa.gov/tools/panoply

Chapter 4

Etcetera

4.1 Hints for Parameter Settings

The 3D simulation is bounded by the total memory size. The code requires

$$m_{\mathrm{MP}} = 116 + 24\mathtt{NM} = 188 \quad (\mathtt{NM} = 3) \quad \text{byte} \tag{4.1}$$

of memory for the case of mixed precision (`MP=DP`) and GNZ viscoelastic body of `NM=3`. Note that this is a coarse estimate excluding the effect of absorbing boundary.

The computation time can be roughly estimated by parameter $\mathtt{n_G}$ which is defined as the number of spatial and/or temporal grids that one CPU can process within a second. This value depends on the CPU as shown in the Table 4.1. The total computation time can be estimated by

$$t_{\mathrm{comp}} = \frac{\mathtt{nx} \times \mathtt{ny} \times \mathtt{nz}}{\mathtt{n_G} \times \mathtt{ncore}} \times \mathtt{nt} \quad [\mathrm{s}], \tag{4.2}$$

where the `ncore` is the number of CPU cores used in the computation. If the estimated time exceeds that provided by the computer system, it is recommended to make the model size smaller and/or to use the checkpointing/restarting.

4.2 Hints for Modification of the Codes

4.2.1 Definition of Your own Velocity Model

The velocity structure is defined by subroutine named `vmodel_*` called by the module `m_medium.F90`. These subroutines commonly has input/output parameters defined in the Table 4.2. By creating a Fortran subroutine that returns the medium parameters `rho, lam,`

Table 4.1 The performance parameter $\mathtt{n_G}$

Architecture Name	CPU	#core/CPU	$\mathtt{n_G}$
Mac Pro 2010	Xeon X5670 2.93GHz	6	6.7×10^6
EIC (ERI, UTokyko)	Xeon E5-2680 v3 2.5 GHz	12	7.0×10^6
The Earth Simulator (3rd gen.)	NEC SX-ACE	4	57×10^6

Table 4.2 Input/ouput specificaiton of subroutines for velocity models

Variable name	In/Out	Type	Description
io_prm	in	int	I/O number of input parameter file
i0,i1	in	int	Start/end indices of arrays in x-direction
j0,j1	in	int	Start/end indices of arrays in y-direction
k0,k1	in	int	Start/end indices of arrays in z-direction
xc(i0:i1)	in	real	x grid locations
yc(i0:i1)	in	real	y grid locations
zc(i0:i1)	in	real	z grid locations
vcut	in	real	Cut-off velocity
rho(k0:k1,i0:i1,j0:j1)	out	real	Mass density [g/cm^3]
lam(k0:k1,i0:i1,j0:j1)	out	real	Lame coefficient λ [g/cm^3]
mu(k0:k1,i0:i1,j0:j1)	out	real	Lame coefficient μ [g/cm^3]
qp(k0:k1,i0:i1,j0:j1)	out	real	Q_P
qs(k0:k1,i0:i1,j0:j1)	out	real	Q_S
bddep(i0:i1,j0:j1,0:NBD)	out	real	Discontinuity boundary depths [km]

mu, qp, qs at locations given in the input of subroutine xc, yc, zc, it is easy to add a new velocity model.

The topography and bathymetry will be investigated automatically in the m_medium module after the call of the vmodel_* routine. To make this investigation work properly, the medium parameter mu must be zero in the air and ocean columns, and lam must be zero in the air column.

Variables bddep(:,:,0) is assumed to be topography, and is used for snapshot output. The other bddep(:,:,1:NBD) will be used for fit the source and/or station location to the discontinuity depths. Give dummy values if these functions are not necessary.

4.2.2 Definition of Your own Source Time Function

The source time function is called as a Fortran's function with names of source__momentrate in the m_source.F90 based on the choice of stftype. The definition of the source time functions are described in share/m_fdtool.F90. It would be easy to add new source time function to here, and to add the call of the new function in the m_source module.

All of pre-defined source time functions takes two time parameters tbeg and trise. In the source code, they are stored in the array variable srcprm(:). If the new source time function requires more than three parameters, the users can expand the array srcprm(:) to store them.

4.2.3 Append New Controll Parameters

At many Fortran modules, the first set-up is performed by subroutines with names of (modulename)__setup at the first time of the computation. Some of the setup modules read parameters from input parameter file. This parameter read is done by the subroutine readini which is defined in shared/m_readini.F90.

Bibliography

Aki, K., and P. G. Richards (2002), *Quantitative Seismology: Theory and Methods*, 2nd eidtion ed., University Science Books.

Blanch, J. O., J. O. Robertsson, and W. W. Symes (1994), Modeling of a constant Q: methodology and algorithm for an efficient and optimally inexpensive viscoelastic technique, *Geophysics*, *60*, 176–184, doi:10.1111/j.1365-246X.2004.02300.x.

Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef (1985), A nonreflecting boundary condition for discrete acoustic and elastic wave equations, *Geophysics*, *50*(4), 705–708.

Koketsu, K., H. Miyake, and H. Suzuki (2012), Japan Integrated Velocity Structure Model Version 1, *Proceedings of the 15th World Conference on Earthquake Engineering*, p. Paper No.1773.

Maeda, T., and T. Furumura (2013), FDM Simulation of Seismic Waves, Ocean Acoustic Waves, and Tsunamis Based on Tsunami-Coupled Equations of Motion, *PAGEOPH*, *170*(1-2), 109–127, doi:10.1007/s00024-011-0430-z.

Maeda, T., T. Furumura, S. Noguchi, S. Takemura, S. Sakai, M. Shinohara, K. Iwai, and S.-J. Lee (2013), Seismic- and tsunami-wave propagation of the 2011 Off the Pacific Coast of Tohoku Earthquake as inferred from the tsunami-coupled finite-difference simulation, *Bulletin of the Seismological Society of America*, *103*(2B), 1456–1472.

Okamoto, T., and H. Takenaka (2005), Fluid-solid boundary implementation in the velocity-stress finite-difference method, *Zisin*, *57*, 355–364.

Robertsson, J. O., J. O. Blanch, and W. W. Symes (1994), Viscoelastic finite-difference modeling, *Geophysics*, *59*(9), 1444–1456, doi:10.1190/1.1443701.

Sato, H., M. C. Fehler, and T. Maeda (2012), *Seismic Wave Propagation and Scattering in the Heterogeneous Earth: Second Edition*, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-23029-5.

Zhang, W., and Y. Shen (2010), Unsplit complex frequency-shifted PML implementation using auxiliary differential equations for seismic wave modeling, *Geophysics*, *75*(4), T141–T154, doi:10.1190/1.3463431.

Acknowledements

This project was supported by the Collaborative Research Program of the Earthquake Research Institute, the University of Tokyo (2015-B-01), and the Core-to-Core Collaborative Research Program of the Earthquake Research Institute, the University of Tokyo and the Disaster Prevention Research Institute, Kyoto University (2016-K-06). This code was developed through research collaborations of: Takashi Furumura, Shunsuke Takemura, Masaru Todoriki, Futoshi Mori, Nana Yoshimitsu, Hiroyuki Kumagai, Hanae Morioka, Aitaro Kato, Issei Doi, and Nozomi Kanaya.

Revision History

2015-06-04 First closed version for ERI/UT joint usage program
2015-06-10 Revision for the new Earth Simulator
2015-06-29 Added random media
2015-07-14 MPI/OpenMP hybrid parallel simulation mode
2015-12-04 Text revision
2016-01-14 Body force and reciprocity modes
2016-02-03 Ouput in `NetCDF` format
2016-05-05 (v1.0) Official open-source release
2016-06-19 (v2.0) Hybrid parallel simulation for 2D codes
2016-08-21 (v3.0) Improved reciprocity mode, geographic projection tools, csf waveform format

Copyright & License

This software is provided under the MIT license.

Copyright (c) 2013-2016 Takuto Maeda

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The author request that the user cite (at least one of) the following papers in any publications that result from the use of this software, although this is not an obligation.

1. Furumura, T. and L. Chen (2004), Large scale parallel simulation and visualization of 3D seismic wavefield using the Earth Simulator, *Computer Modeling of Engeneering and Sciences*, *6*(2), 153-168.
2. Furumura, T. and L. Chen (2005), Parallel simulation of strong ground motions during recent and historical damaging earthquakes in Tokyo, Japan, *Parallel Computing*, *31*, 149-165.
3. Furumura, T. Hayakawa, M. Nakamura, K. Koketsu, and T. Baba (2008), Development of long-period ground motions from the Nankai Trough, Japan, earthquakes: Observations and computer simulation of the 1944 Tonankai (Mw8.1) and the 2004 SE Off-Kii Peninsula (Mw7) Earthquakes, *Pure Appl. Geophys.*, *165*, 585-607.
4. Furumura, T. and T. Saito (2009), An integrated simulation of ground motion and tsunami for the 1944 Tonankai earthquake using high-performance super computers, *Journal of Disaster Research*, *4*(2), 118-126.
5. Maeda, T., and T. Furumura (2013), FDM simulation of seismic waves, ocean acoustic waves, and tsunamis based on tsunami-coupled equations of motion, *Pure Appl. Geophys.*, *170*(1-2), 109-127, doi:10.1007/s00024-011-0430-z.
6. Noguchi, S., T. Maeda, and T. Furumura (2013), FDM simulation of an anomalous later phase from the Japan Trench subduction zone earthquakes, *Pure Appl. Geophys.*, *170*(1-2), 95-108, doi:10.1007/s00024-011-0412-1.
7. Maeda, T., T. Furumura, S. Noguchi, S. Takemura, S. Sakai, M. Shinohara, K. Iwai, S. J.

Lee (2013), Seismic and tsunami wave propagation of the 2011 Off the Pacific Coast of Tohoku Earthquake as inferred from the tsunami-coupled finite difference simulation, *Bull. Seism. Soc. Am.*, *103*(2b), 1456-1472, doi:10.1785/0120120118.

8. Maeda, T., T. Furumura, and K. Obara (2014), Scattering of teleseismic P-waves by the Japan Trench: A significant effect of reverberation in the seawater column, *Earth Planet. Sci. Lett.*, *397*(1), 101-110, doi:10.1016/j.epsl.2014.04.037.
9. Noguchi, S., T. Maeda, and T. Furumura (2016), Ocean-influenced Rayleigh waves from outer-rise earthquakes and their effects on durations of long-period ground motion, *Geophys. J. Int.*, *205*(2), 1099-1107, doi:10.1093/gji/ggw074.