

OpenSWPC:

A Seismic Wave Propagation Code

Tutorial

Version 1.0j (2016-05-15)

Takuto Maeda

目次

1	OpenSWPC の特徴	2
2	準備	2
2.1	ソースコードとバイナリの配置	2
2.2	構造モデルと観測点情報ファイルの生成	2
3	データセットについて	3
3.1	構造モデル	3
4	座標系と主要パラメタ	4
4.1	計算座標系と地理座標系	4
4.2	主要パラメタ	6
4.3	震源リストファイル	9
4.4	観測点リストファイル	9
5	実行	10
5.1	小型サンプルジョブ（ベンチマークテスト）	10
5.2	大規模シミュレーション用のパラメタセット	12

本文書では、東京大学地震研究所の計算機システム EIC の利用を想定した環境下において、配布物に同梱されている実行ファイルとサンプル入力ファイルを用いた OpenSWPC による地震動シミュレーションを動作させるための最小限の解説を行う。

1 OpenSWPC の特徴

OpenSWPC は 2 次元・3 次元のデカルト座標系において粘弾性体の運動方程式を差分法によって解くことで地震動を計算する MPI 並列コード群であり、以下のような特徴がある。

- 簡単に MPI による並列シミュレーションが行える。
- ほとんどの動作制御をパラメタファイルの編集だけで行うことができ、コード改造の必要が少ない。
- 構造は GMT で用いられる `grd` ファイル形式の不連続面を入力ファイルとして用いる。
- 広域の構造情報から、特定領域のみを切り出してシミュレーションすることができる。
- 結果は SAC 波形と水平・鉛直断面のスナップショットとして保存できる。
- スナップショットの可視化・変換ツールも同梱されており、ポストプロセッシングが簡単である。
- EIC 環境でコンパイル済みの実行ファイルとサンプル入力ファイルが用意されており、3 次元地震波計算を簡単に実施できる。

2 準備

2.1 ソースコードとバイナリの配置

EIC 上でファイルをコピーする。

```
1 cp -rif /work/maeda/share/OpenSWPC .
```

EIC 以外の環境であれば、github リポジトリからクローン、あるいは release 版をダウンロードし、解凍する。

次に、データディレクトリの場所を環境変数 `${DATASET}` に設定する。地震研究所 EIC の場合はログインシェルが `csh` なので、ホームディレクトリの `${HOME}/.cshrc` に以下の行を追記する（ファイルが無ければ作成する）。

```
1 setenv DATASET ${HOME}/OpenSWPC/dataset
```

上記の例ではホームディレクトリ直下にファイルがあると仮定されている。必要に応じて変更されたい。なお、設定内容は次回ログイン時から有効になる。今すぐ有効にするには、

```
1 source ~/.cshrc
```

とする。なお、環境変数が適切に設定されてさえいれば、この `${DATASET}` ディレクトリは他のコードと同じ場所に置いておく必要はなく、どこに移動しても良い。

2.2 構造モデルと観測点情報ファイルの生成

`dataset/vmodel`/ディレクトリで構造モデルを生成する。

```
1 cd dataset/vmodel
2 ./gen_JIVSM.sh
3 rm -f *.zip
```

スクリプト `gen_JIVSM.sh` は必要な外部データファイルをダウンロードし、GMT を用いて構造モデルを自動生成する。構造モデル作成には 30 分程度の時間を要する。スクリプト実行後、`jivsm` と `ejivsm` にそれぞ

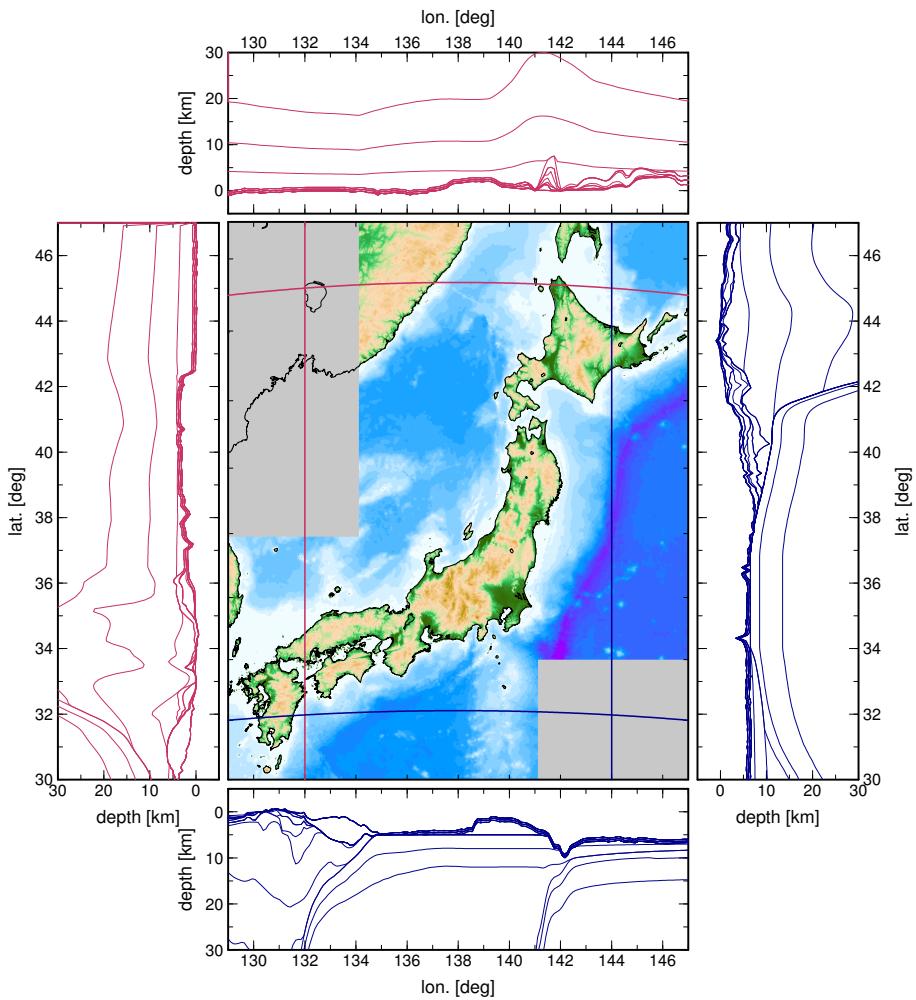


図1 JIVSM モデルと eJIVSM モデル. 中央地図で標高値による色が塗られた部分が JIVSM で構造が定義されている領域, 灰色部分が eJIVSM モデルによって拡張される未定義領域である. 地図上の青・赤線に沿った構造モデル不連続面の深さ断面を上下左右にそれぞれ示す.

grd ファイルが格納されていることを確認する.

次に, 防災科学技術研究所 (http://www.hinet.bosai.go.jp/st_info/detail/dataset.php) から観測点一覧を CSV 形式でダウンロードし, dataset/station に格納する. その上で, 同ディレクトリのスクリプト gen_stlst_hinet.sh を実行することで, Hi-net 観測点位置情報ファイルが作成される.

3 データセットについて

3.1 構造モデル

OpenSWPC では 3 次元不均質構造の表現の一つとして成層構造を採用している. 一つ一つの層は境界面の深さ (あるいは標高) 値と物性値 (P・S 波速度, 密度, Q 値) から定義され, 境界面の深さより下がその物性値であることを示す. 深さが緯度経度によって変化するこれらの層を多数重ねることにより, 地表の地形形状からプレート境界までを一元的に表現できる. OpenSWPC は netcdf 形式 (GMT grd ファイルフォーマット) の緯度・経度・深さ (m) の任意のファイルを入力とすることができる. 特に標準モデルとして, 長周期地震動予

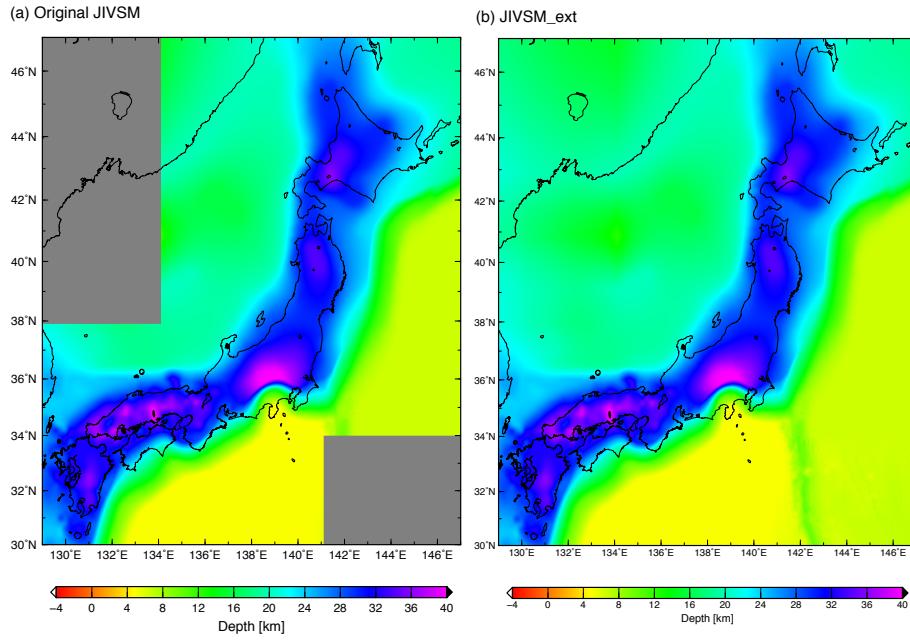


図 2 JIVSM と JIVSM_ext モデルのモホ面深さの比較. (a) の灰色部分はモデル未提供領域.

測地図 2012 年試作版に付属の全国 1 次地下構造モデル (JIVSM; [Koketsu et al. \(2012\)](#))^{*1}を採用した。このモデルは、日本列島を東日本と西日本の一部重複した領域に分割して提供されている。しかし、そのままでは統一的な取り扱いに不便であるうえ、日本海の北西部ならびに太平洋南西部にデータが存在しない領域がある。そこで、JIVSM のモデルをなめらかさを維持したまま融合し、かつ緯度経度 [129/147/30/47] 範囲をすべて覆うように補外した eJIVSM モデル (図 1) も提供する。このモデルは [Maeda et al. \(2014\)](#) で用いられたものと等価である。

このモデルでは内部境界面は補外により、海底地形は別の海底地形データから再現されている。海底地形からの境界面への相対深度について補外を行い、それを別途海底地形データに足し込むという処理によって構造モデルを作成した。そのため、例えばモホ面 (図 2) は補外領域まで含めて海底地形の形状をあるていど反映した深度分布になる。また、標高と海底地形は NOAA の ETOP01 に差し替えられている。

スクリプト `gen_JIVSM.sh` によってモデルを作成すると、これらのファイルは NetCDF フォーマットで \${DATASET}/vmodels/JIVSM および \${DATASET}/vmodels/eJIVSM に格納されている。詳しくは User's Guide 参照のこと。

4 座標系と主要パラメタ

4.1 計算座標系と地理座標系

水平方向には図 3 のような座標系を取る。デフォルトでは xyz がそれぞれ北・東・下向き正の右手系で、それぞれ $x_{\text{beg}} \leq x \leq x_{\text{end}}$, $y_{\text{beg}} \leq y \leq y_{\text{end}}$, $z_{\text{beg}} \leq z \leq z_{\text{end}}$ の領域をとる。この領域が各方向 dx , dy , dz の空間刻みで nx , ny , nz 個のグリッドに離散化される。実際には、 x_{beg} , y_{beg} , z_{beg} と nx , ny , nz をパラメタとして与え、それらから x_{end} , y_{end} , z_{end} は自動的に計算される。

OpenSWPC は並列差分法により地震動を計算する。全モデル空間を図 4 のように 2 次元的に `nproc_x` ×

^{*1} http://www.jishin.go.jp/main/chousa/12_choshuki/dat/index.htm

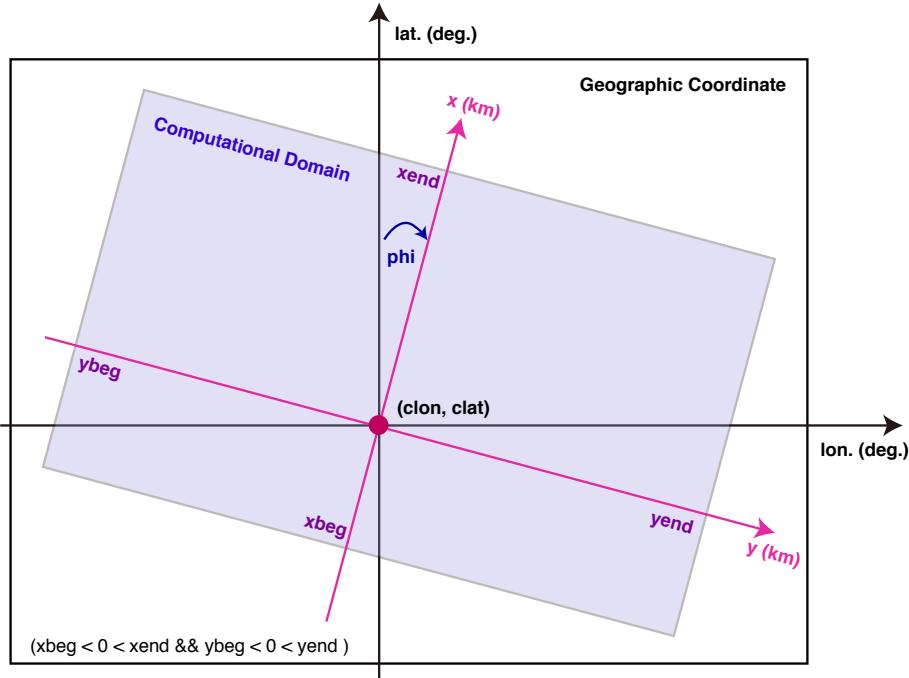


図3 地理座標系と計算座標系（水平断面）の関係

`nproc_y` 個に分割する。一つのブロックは一つのノード (CPU もしくは CPU コアなどの並列計算の単位) が担当して数値計算を行い、毎離散時間ステップに隣接ノードと通信を行うことで波動場の連続性を担保する。ユーザーが与える必要があるのは分割数 `nproc_x`, `nproc_y` だけである。また、水平方向の全グリッド数 `nx, ny` は分割数 `nproc_x`, `nproc_y` で割り切れる必要は無い。剩余がある場合には、端の計算を担当するノードが数グリッド多く計算を行う。このことの計算速度への影響は極めて限定的である。

近年の CPU の多くは CPU 内に複数の CPU コアをもつ構造をしており、单一 CPU 内部のコア間では、メモリが共有されている。OpenSWPC では、CPU 内部のコア間については OpenMP によるメモリ共有型の並列化を、CPU 間においては MPI による明示的な通信を伴う並列化をそれぞれ行うハイブリッド並列計算に対応している^{*2}。これにより、比較的少ない分割数で大規模な計算を実施できる。なお、CPU コア一つ一つを MPI で独立に扱う Flat-MPI にも対応可能である。

計算はすべてデカルト座標系で行われる。デカルト座標系の座標中心 ($x = 0, y = 0$) の地理座標系における緯度経度を `clon, clat` とし、そこを中心としたガウス・クリューゲル変換により地理座標系とデカルト座標系を結びつける。具体的には図 5 に例示されたような以下の手続きが内部的に行われる（ただし、実際のコードでは順序が一部異なっている）。

- (i) まずデカルト座標系において等間隔のグリッドを生成する
- (ii) 一つ一つのグリッド点を `clon, clat` を中心とした座標変換により緯度経度座標に変換する
- (iii) 地形や速度構造モデルに当てはめる。上述の変換されたグリッド点の位置における標高や不連続面の深さを構造モデルの `grd` データから双3次補間によって推定し構造モデルを構築する
- (iv) デカルト座標に戻り、距離座標系からグリッド番号座標に変換する（内部的にはなにもしない）
- (v) 予め与えられた並列数 `nproc_x, nproc_y` により各計算ノードにモデルを分割する

特に z 方向については、 $z = 0$ が静水面（海水面）の高さであることが仮定されているため、`zbeg` は負の値

^{*2} 2015-07-14 版以降に実装されている

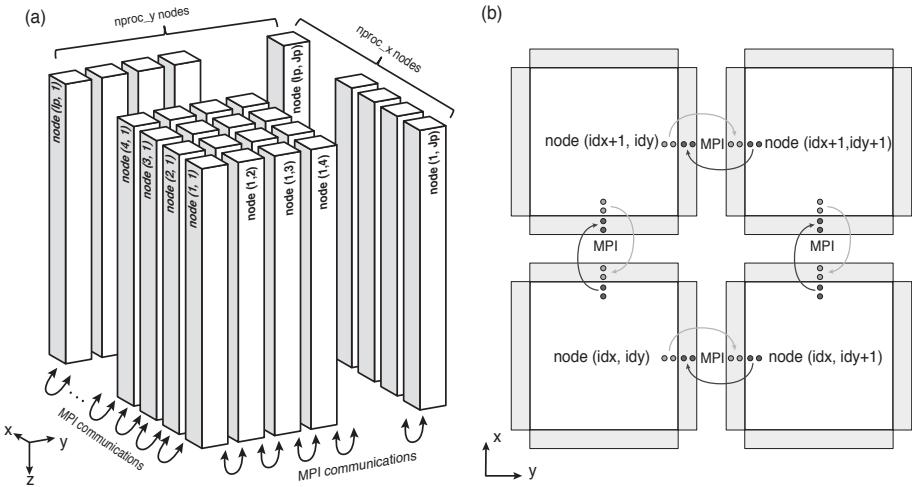


図 4 3 次元計算における並列計算のための領域分割 (modified from [Maeda et al., 2013](#))

に取ることが推奨される。 x, y 方向についてはこのような制限は無いが、地理座標系への座標変換の歪みを小さくするために、できるだけ `clon`, `clat` が計算モデルの中心付近にあることが望ましい。もし `xbeg`, `ybeg` のパラメタ指定が省略された場合には、`clon`, `clat` が中心になり、座標変換の歪みが最小になるように自動調整される。なお、パラメタ `phi` によって座標系を回転させることもできる。ただし、この場合は出力される地震波形の成分も回転後の x, y 方向成分となり、南北東西方向に自動回転はされない。そのため、観測記録との比較には注意が必要である。

4.2 主要パラメタ

OpenSWPC の挙動はテキスト形式の入力パラメタファイルによって行われる。ここでは最低限のパラメタのみ示す。すべてのパラメタ項目についての詳細は User's Guide を参照されたい。パラメタは 1 行にひとつずつ、等号 (=) でパラメタ名と値を関係づける。パラメタファイル内の記述の順序は問わず、!ならびに#で始まる行や文字列はコメントとして扱われる。パラメタのうち、ディレクトリ名には環境変数が利用可能である。環境変数は \${} で括って記入する。波括弧が無いと環境変数であると認識されないので注意すること。

```

1
2  !! -----
3  !! Control
4  !! -----
5
6  title          = 'swpc'           !! exe title: used for output filenames
7  odir           = './out'          !! output directory
8
9  !! -----
10 !! Model/Grid Size and Area
11 !! -----
12
13 nproc_x        = 4               !! parallelization in x-dir
14 nproc_y        = 6               !! parallelization in x-dir
15 nx             = 384            !! total grid number in x-dir
16 ny             = 384            !! total grid number in y-dir
17 nz             = 384            !! total grid number in z-dir
18 nt             = 1000           !! time step number
19
20 dx             = 0.5            !! grid width in x-dir
21 dy             = 0.5            !! grid width in y-dir

```

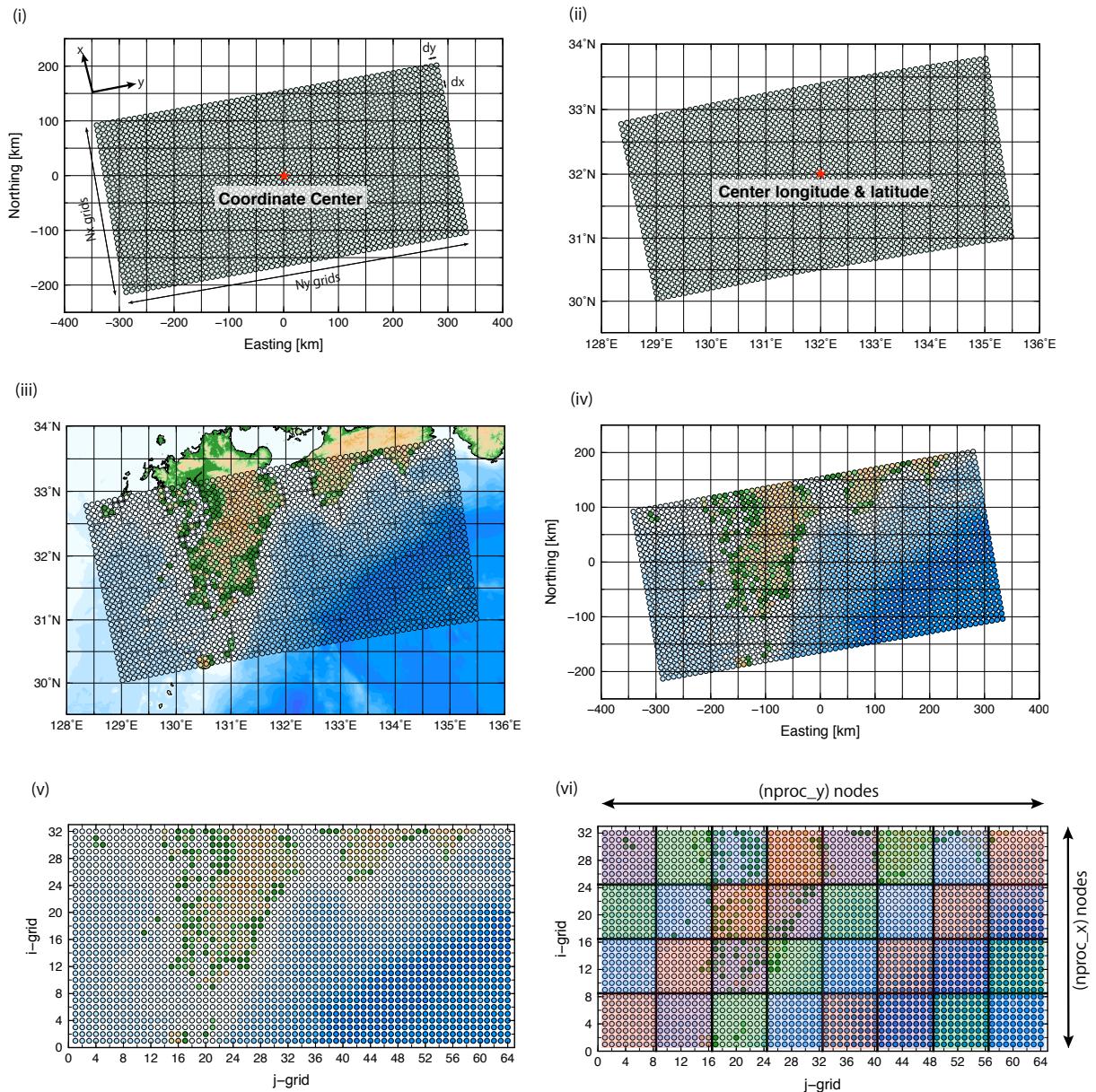


図 5 OpenSWPC による構造のモデル化の模式図。

```

22      dz          = 0.5           !! grid width in z-dir
23      dt          = 0.02          !! time step width
24
25      vcut        = 1.5           !! minimum velocity
26                                !- smaller velocity will be raised
27
28      xbeg        = -96.0         !! minimum in x-dir
29      ybeg        = -96.0         !! minimum in y-dir
30      zbeg        = -10.0          !! minimum in z-dir
31      tbeg        = 0.0           !! start time
32
33      clon       = 139.7604      !! center longitude
34      clat       = 35.7182      !! center latitude
35      phi         = 0.0           !! horizontal coordinate rotation
36                                !- measured clockwise from north

```

```

37      !! -----
38      !! Snapshot Output
39      !! -----
40      !!
41
42      xy_ps%sw      = .false.          !! P&S amp. for xy section
43      xz_ps%sw      = .true.           !! P&S amp. for xz section
44      yz_ps%sw      = .false.           !! P&S amp. for yz section
45      fs_ps%sw      = .false.           !! P&S amp. for free surface
46      ob_ps%sw      = .true.            !! P&S amp. for ocean bottom
47
48      xy_v%sw       = .false.           !! 3-comp. velocity for xy section
49      xz_v%sw       = .true.            !! 3-comp. velocity for xz section
50      yz_v%sw       = .false.           !! 3-comp. velocity for yz section
51      fs_v%sw       = .false.           !! 3-comp. velocity for free surface
52      ob_v%sw       = .true.            !! 3-comp. velocity for ocean bottom
53
54      xy_u%sw       = .false.           !! 3-comp. displacement for xy section
55      xz_u%sw       = .true.            !! 3-comp. displacement for xz section
56      yz_u%sw       = .false.           !! 3-comp. displacement for yz section
57      fs_u%sw       = .false.           !! 3-comp. displacement for free surface
58      ob_u%sw       = .true.            !! 3-comp. displacement for ocean bottom
59
60      z0_xy         = 7.0              !! depth for xy cross section
61      x0_yz         = 0.0              !! x-value for yz cross section
62      y0_xz         = 0.0              !! y-value for xz cross section
63
64      ntdec_s       = 5                !! time decimation of snapshot
65      !- (specify 1 for no decimation)
66
67      idec          = 2                !! x-decimation for snapshot
68      jdec          = 2                !! y-decimation for snapshot
69      kdec          = 2                !! z-decimation for snapshot
70
71      !! -----
72      !! Waveform Output
73      !! -----
74
75      sw_wav_v       = .true.           !! velocity trace output at stations
76      sw_wav_u       = .false.          !! displacement trace output at stations
77      ntdec_w        = 5                !! time decimation of waveform output
78      fn_stloc       = '${DATASET}/station/hinet.stlst' !! station location file
79
80      !! -----
81      !! Earthquake Source
82      !! -----
83
84      !! Moment tensor source format:
85      !!   xym0ij: x, y, z, tbeg, trise, mo, mxx, myy, mzz, myz, mxz, mxy
86      !!   xym0dc: x, y, z, tbeg, trise, mo, strike, dip, rake
87      !!   llm0ij: lon,lat,z, tbeg, trise, mo, mxx, myy, mzz, myz, mxz, mxy
88      !!   llm0dc: lon,lat,z, tbeg, trise, mo, strike, dip, rake
89      !!   xymwij: x, y, z, tbeg, trise, mag, mxx, myy, mzz, myz, mxz, mxy
90      !!   xymwdc: x, y, z, tbeg, trise, mag, strike, dip, rake
91      !!   llmwij: lon,lat,z, tbeg, trise, mag, mxx, myy, mzz, myz, mxz, mxy
92      !!   llmwdc: lon,lat,z, tbeg, trise, mag, strike, dip, rake
93      !! Body force source fomrat:
94      !!   xy: x, y, z, tbeg, trise, fx, fy, fz
95      !!   ll: lon, lat, z, tbeg, trise, fx, fy, fz
96
97      stf_format      = 'xym0ij'
98

```

```
99 fn_stf      = "./in/source.dat"    !! Source grid file
```

4.3 震源リストファイル

震源や観測点は別のリストファイルとして与えられ、そのファイル名がパラメタファイルに記述されている。震源は場所が緯度経度かデカルト座標か、震源メカニズムがモーメントテンソルか(strike, dip, rake)か、規模が地震モーメントかモーメントマグニチュードか、によって様々なフォーマットで記述可能である。ここでは緯度経度・(strike, dip, rake)・モーメントマグニチュードの場合の入力ファイルの例を示す。#で始まる行はコメントとして無視される。

```
1 #
2 # format 'llmwdc': lon/lat, moment magnitude, double couple angles
3 #
4 #   lon   lat   z   tbeg   trise   mag   strike   dip   rake
5 # -----
6 139.76  35.72  10.0   0.0     4.0     5.0     0.0    45.0   90.0
```

緯度経度の他、震源時間関数(三角形形状をなめらかにしたもの)の破壊開始時間と継続時間(=モーメント関数のライズタイム)をtriseによって指定する。継続時間がシミュレーション結果の卓越周期に大きく影響する。短くしすぎるとシミュレーションの限界を超え、数値分散が現れる。適切なパラメタの選択方法については、専用のツールが提供されている(User's Guide参照)。

4.4 観測点リストファイル

観測点は以下のようなリストファイルで与える。リストファイルには緯度経度深さ情報を記述する。ただし、深さについてはグリッドが地表の上に飛び出すのを防ぐためのオプションを指定できる。以下の例では"obb"オプションによって海底(地表)面の1グリッド下に観測点が配置される。これは、差分法のスタガードグリッドの配置のため、変数によっては出力位置が地表面(あるいは海底面)より上側に飛び出すことを避けるための設定である。特段の理由が無ければ、地表観測点はこの"obb"オプションで出力することを推奨する。オプション詳細についてはUser's Guideを参照のこと。

なお、計算領域の外に位置する観測点は自動的に出力から除外される。したがって、シミュレーション毎に観測点リストを再作成する必要はなく、常に全国全観測点のリストを指定しておくのが簡単である。

```
1 # hinet.stlst
2 #
3 # station location data by geographical format.
4 # lines starting from '#' and blank lines are omitted.
5 #
6 # zsw: controls station depth
7 #   'dep': use the depth
8 #   'fsb': locate one-grid below from the free surface/sea surface
9 #   'obb': locate one-grid below from the ground surface/seafloor
10 #   'oba': locate one-grid above from the ground surface/seafloor
11 #   'pac': on the pacific plate interface
12 #   'phs': on the philippine sea plate interface
13 #
14 #   lon       lat       dep      stnm   zsw
15 # -----
16 129.4353  33.2116  0.1780  N.HRDH  obb
17 129.7652  33.2122 -0.0270  N.SBNH  obb
18 129.8026  32.9553  0.0980  N.KKIH  obb
19 129.8102  33.1256  0.1910  N.SBSH  obb
20 129.8625  32.6999  0.0460  N.NGSH  obb
```

21	129.8798	33.2656	0.0730	N.IMRH	obb
22	129.8877	33.5080	0.0840	N.CZEH	obb
23	130.1048	33.1805	0.1920	N.SRIH	obb
24	130.1191	32.1554	0.1880	N.NSMH	obb
25	130.1408	32.9103	0.1720	N.TKKH	obb
26	130.1811	32.3152	0.1410	N.SNWH	obb
27	130.2077	31.9940	0.1610	N.AKNH	obb
28	130.2329	33.4098	-0.2710	N.FJIH	obb
29	130.3602	31.8374	0.0870	N.SENH	obb
30	130.3647	32.1704	0.0960	N.MMTH	obb
31	130.4049	33.3652	0.1460	N.HSFH	obb

5 実行

EIC 環境でコンパイルしたバイナリファイルが提供されているため、改造せずに動かす限りは再コンパイル不要である。サンプル入力ファイルとジョブスクリプトの例が `SEISM/examples/` に格納されている。他環境で動作させるためには、シミュレーションコードに用いるものと同じ Fortran コンパイラでビルドされた NetCDF ライブラリが必要である。

5.1 小型サンプルジョブ（ベンチマークテスト）

`example/input.inf` に比較的小規模なテスト計算用の入力ファイル例がある。以下主要なパラメタを抜粋する。

```

1 nproc_x          = 4           !! x-方向並列数
2 nproc_y          = 8           !! y-方向並列数
3 nx              = 256          !! x軸全グリッド数
4 ny              = 256          !! y軸全グリッド数
5 nz              = 256          !! z軸全グリッド数
6 nt              = 2000         !! 時間軸グリッド数
7
8 dx              = 0.5          !! x方向グリッド間隔
9 dy              = 0.5          !! y方向グリッド間隔
10 dz             = 0.5          !! z方向グリッド間隔
11 dt              = 0.02         !! t方向グリッド間隔
12
13 benchmark_mode = .true.

```

このモデルではグリッド間隔 0.5km で、一辺 $256 \times 0.5 = 128$ km の 3 次元領域を、経過時間を 0.02 秒刻みで経過時間 40 秒まで計算する。`benchmark_mode` は動作確認用の特別なモードで、他のパラメタを無視して均質媒質の水平方向中心に等方震源がある場合を仮定して運動方程式を解く。

EIC で計算を実行するには、プログラムを実行するジョブスクリプトを書き、バッチシステムに投入して実行を待つ必要がある。本サンプルに対応したジョブスクリプトが `example/eic_bench.job` にある。

```

1#!/usr/bin/csh
2
3#!/bin/csh
4#$ -jc C           # Job Class
5#$ -ac n=32         # Total Core
6#$ -ac P=32         # Total Core
7#$ -cwd             # Execute at the current directory
8#$ -j y             # Combine stderr & stdout
9#$ -o swpc_out.txt # Ouput file
10#$ -N SEISM        # Job name
11
12mpiexec_mpt -np 32 dplace -s1 ./bin/swpc_3d.x -i example/input.inf

```

ジョブの投入は SEISM のディレクトリにて

```
1 > qsub example/eic_bench.job
```

によって行う。正常に投入されれば

```
1 Your job XXXX ("SEISM") has been submitted
```

のような表示が現れる。ジョブの進捗状況は `qstat` コマンドによって確認できる。

ひとたびジョブが実行状態になれば、数分で計算は終わる筈である。計算が終わると、結果出力ファイル `swpc.out.txt`（このファイル名はジョブスクリプトで指定したものである）が作成され、そこに以下のようないい結果が出力されている。

```
1 Original PE host file
2 eich3 32 eich.q@eich3 <NULL>
3 New PE host file
4 eich3 32 eich.q@eich3 <NULL>
5 Now exucuting UGE job (debug)
6 [info{0}] m_output.F90(394): station st02 is out of the region
7
8 -----
9   SEISM_3D (BENCHMARK MODE)
10 -----
11
12   Grid Size          :      256 x  256 x  256
13   MPI Partitioning  :          4 x     8
14   Total Memory Size :      3.638 [GiB]
15   Node Memory Size  :      0.114 [GiB]
16   Stability Condition c :      0.980 (c<1)
17   Wavelength Condition r :      7.000 (r>5-10)
18   Minimum velocity    :      3.500 [km/s]
19   Maximum velocity    :      6.062 [km/s]
20   Maximum frequency   :      1.000 [Hz]
21
22 -----
23
24   it=0000050, 0.433 s/loop, eta 000:14:04, ( 5.01E-05  5.01E-05  4.95E-05 )
25   it=0000100, 0.436 s/loop, eta 000:13:49, ( 1.75E-05  1.75E-05  1.05E-05 )
26   it=0000150, 0.435 s/loop, eta 000:13:24, ( 1.02E-05  1.02E-05  5.36E-06 )
27   it=0000200, 0.434 s/loop, eta 000:13:01, ( 6.64E-06  6.64E-06  4.31E-06 )
28
29 (中略)
30
31   it=0001950, 0.442 s/loop, eta 000:00:22, ( 5.48E-09  5.48E-09  3.43E-09 )
32   it=0002000, 0.441 s/loop, eta 000:00:00, ( 4.94E-09  4.94E-09  3.17E-09 )
33
34 -----
35
36   Total time          :      882.803 s
37   Score                :      1.188E+06  1.678E+07  5.243E+05      32      4      8
38
39 -----
40 -----
```

出力のうち、`it=`で始まる行は、一定時間ステップ（この例では 50 ステップ）に一度、計算時間に関わる情報と地表面における 3 成分地震波速度振幅の最大値を出力している。

5.1.1 出力ファイルと可視化

計算が正常に終了すれば、入力パラメタの指定に応じた出力ディレクトリ（この場合は `./out/`）が作られ、その下にスナップショットファイル (`*.nc`) と SAC 形式の波形ファイル (`./out/wav/*sac`) が出力される。ス

ナップショットファイルは計算の途中に少しずつ追記されていくが、波形ファイルは計算終了時にまとめて出力される。ここでは NetCDF 形式のファイル出力を用いるが、パラメタの選択により、外部ライブラリを必要としない独自バイナリ形式で出力することもできる。

波形ファイルは、example/stloc.dat で定義されている場所の速度波形が格納されている。ただし、この観測点位置ファイルに含まれる観測点のひとつは、計算領域の外側にあるため、

```
1 [info{0}] m_output.F90(394): station st02 is out of the region
```

の警告メッセージが表示され、その波形は出力されていない。

出力波形ファイルはそのまま SAC で利用できる。計算座標系は北・東・下がそれぞれ正であったが、上下動方向の SAC 波形ファイルは、慣例に従って上が正になるように変換されている。観測点の位置（緯度経度）や震源リストファイルで与えられた震源位置がヘッダに格納されている。震源が複数ある場合には震源リストの最初の行の位置が SAC ヘッダの震源情報として採用される。また、プログラムを実行した日時および時間が kztime として設定される。開始時刻ヘッダ b は常に 0 である。サンプルに用いた震源（ベンチマークモードでは入力ファイルによらず自動的に定まる）の破壊開始時刻が 0 であるので、この場合はファイルの開始時刻が震源の破壊時刻と一致する。一般的なシミュレーションの場合には、入力する震源情報によって破壊開始時刻を制御することができ、その時刻は SAC の o ヘッダに格納されている。

空間スナップショットの可視化には read_snp.x プログラムを用いる。出力ディレクトリに移動して、以下のように可視化を行う。

```
1 > cd out
2 > ./bin/read.snp.x -i swpc.xz.ps.nc -ppm -mul 1e10
3   ppm/swpc.xz.ps.000001.ppm
4   ppm/swpc.xz.ps.000002.ppm
5   ppm/swpc.xz.ps.000003.ppm
6   (中略)
7   ppm/swpc.xz.ps.000399.ppm
8   EOF detected
```

これにより ppm ディレクトリ以下にスナップショットの連番画像ファイルが ppm 形式で作成されている（図 6）。画像形式は ppm の他に windows bmp も選択できる（-bmp オプション）。-mul で指定している数字は可視化の重みであり、振幅（SI 単位系）にこの重みが乗じられた値が RGB 値の計算に用いられる。計算結果出力ファイルに現れる振幅最大値の概ね 10 倍から 100 倍程度を指定するうまく可視化できことが多い。プログラム read.snp.x は環境変数\${DATASET} に定義されたディレクトリ以下のフォント画像データを用いる。環境変数が定義されていないと経過時間が表示されない。

OpenSWPC のスナップショット出力には div/rot (P 波・S 波) と 3 成分速度振幅があるが、そのどちらもこのプログラムで可視化できる。また、特定の成分にだけ色を付けたり、あるいは GMT 等で扱うためのデータを出力することもできる。詳しくは User's Guide を参照のこと。

5.2 大規模シミュレーション用のパラメタセット

前節までのサンプルパラメタを修正することにより様々な解像度や規模のシミュレーションが実施可能である。しかし、現実的な 3 次元シミュレーションを行おうとすると EIC の計算機規模の上限近い性能が必要になり、パラメタの設定によっては有限時間内に計算が終わらない、あるいはそもそもメモリが足りず計算が起動しないといったことが起こりうる。

そこで、EIC で現実的な時間内で実行可能な OpenSWPC のパラメタを 3 種類同梱した。SEISM/example/input_eic_?.inf がそれであり、それぞれ図 7 の 3 領域（North, Central, West）に対応する。それぞれの領域において、F-net CMT 解を参考にしたサンプルの震源情報 source.?.dat が用いられる。この入力ファイルに対応するジョブ

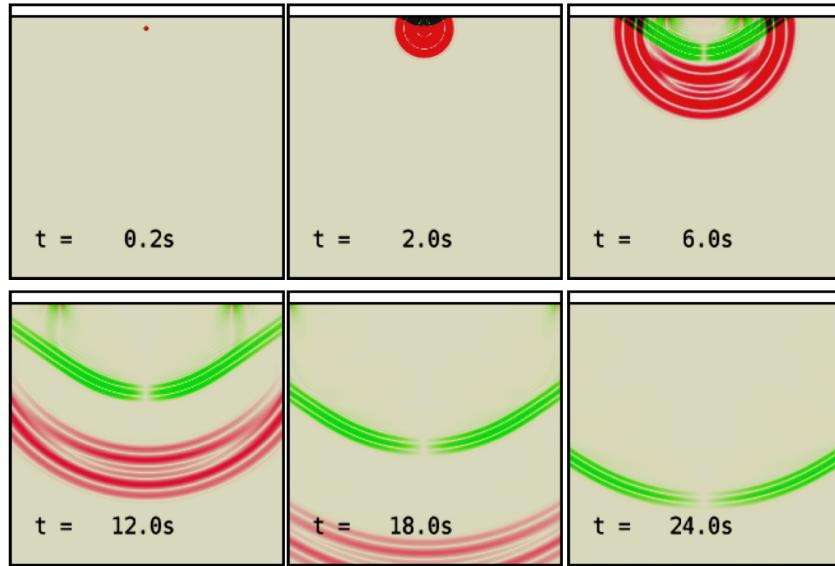


図6 `read.snp.x` による可視化結果例. 赤色が地震波速度の `div` を, 緑が `rot` ベクトルの絶対値をそれぞれ示す.

スクリプトのサンプルが `example/eic_?.csh` にある. ジョブの投入は, 小型サンプルジョブと同様に,

```

1 > qsub example/eic_N.job
2 > qsub example/eic_C.job
3 > qsub example/eic_W.job

```

である. ベンチマークテストの場合と異なり, これらの計算は速度構造をファイルから読み込む. そのため, 事前に環境変数 \${DATASET} が正しく設定され, かつ構造モデルが生成されている必要がある.

これらのジョブを投入すると, EIC で利用可能な最大規模である 36 CPU (432 CPU コア) を使った計算が実行される. 経過時間 600 秒分の計算に概ね 4 時間半程度を要する. ただし, 大規模な G クラスの計算ジョブは最大同時実行数には制限があるため, ジョブ投入後即座に計算が開始されるとは限らない.

計算結果として, スナップショットファイルと計算領域内にある Hi-net 観測点位置での速度波形が出力される. 図 8 に計算波形の例を示す. スナップショットファイルの出力サイズは合計で 10 GB ほどである. 特にスナップショットファイルは, 空間並びに時間間引きを行わないと出力サイズが莫大になることがあるので注意すること. 地震波形についてはさほどサイズは大きくならないが, ファイル数が非常に多くなりがちである.

これらのサンプルパラメータでは, 広い領域のシミュレーションを実現するため, 下記のようなパラメタセットを用いている.

- 最小速度 1.5 km/s
- 水平解像度 0.5km, 鉛直 0.5km
- 最小周期 5 秒程度

長周期地震動のシミュレーションであれば, これらのパラメタをそのまま用い, 震源ファイルの位置情報とメカニズムを変更するだけで実用的な計算が可能である. 最小速度をより大きく取ることでより高周波の地震動を再現できる. ただしその代償として低速度の堆積層が十分に再現できなくなる恐れがある. あるいは, グリッド数をそのままに空間グリッド間隔を狭め, 領域そのものを狭くすることによって, 高解像度・高周波のシミュレーションを行うことも可能である. グリッド間隔等のパラメタの選択には, 専用のプログラム

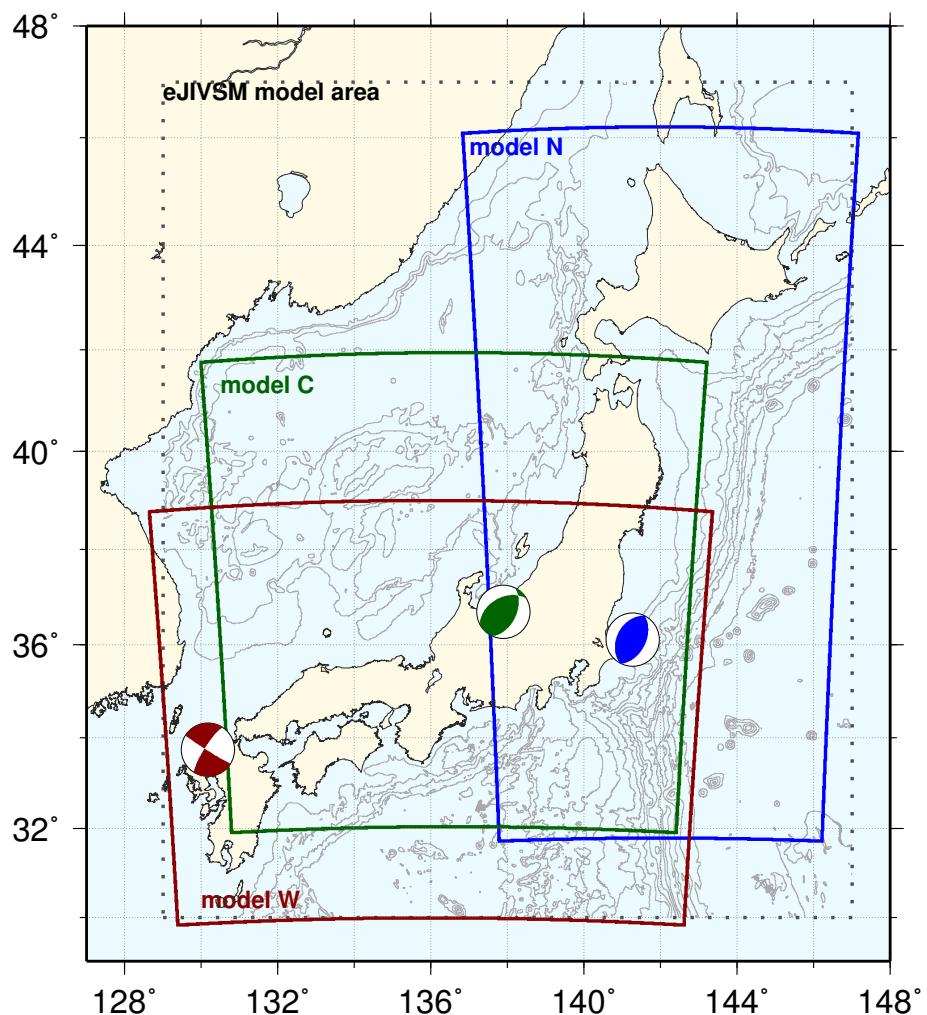


図7 大規模計算用の3モデルセットと、それぞれのパラメタセットで利用されるサンプル震源。

`fdmcond.x` が用意されている,

参考文献

- 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., 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.
- 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 and Planetary Science Letters*, 397(C), 101–110.

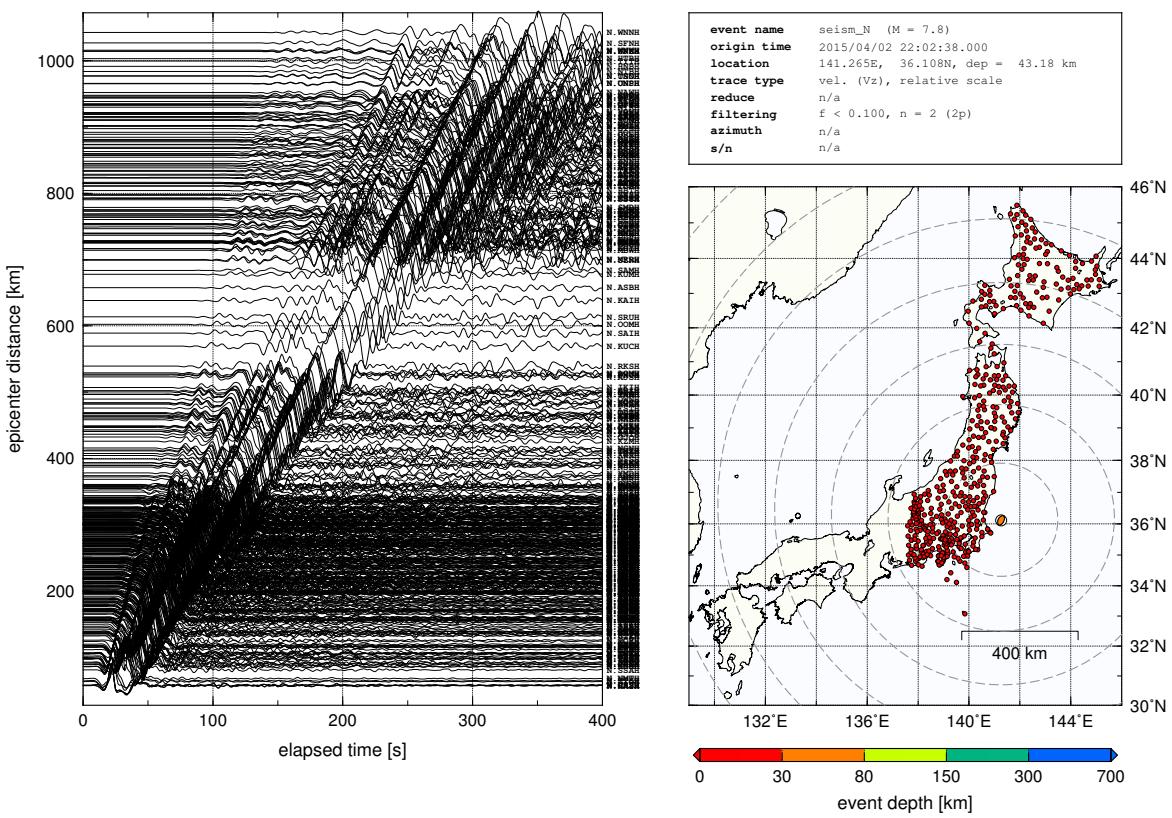


図8 サンプルファイル `input_eic.N.inf` による計算結果例。シミュレーション結果に $f \leq 0.1$ s のローパスフィルタを施した上下動速度波形のペーストアップ。