
README file for the BROAD-BAND TOOLBOX code (broad-band synthetics through high frequency coda generation)

- Martin P. Mai, Banu Mena, Walter Imperatori - Version 1.3, January '09

1) INTRODUCTION

The following tutorial explains how to set up and obtain synthetic seismograms by means of the Broad-Band Toolbox code. This code allows you to compute hybrid broad-band synthetics using pre-computed low-frequency seismograms. The high-frequency contributions are based on a S-S multiple scattering model proposed by Zeng et al. (1991,1993). High- and low-frequency seismograms are then combined, accounting for both amplitude and phase matching, using the methodology proposed by Mai and Beroza (2003). Moreover, in the current release, an additional feature allows the user to combine pre-computed low-frequency contributions with EXSIM pre-computed high-frequency ones (Motazedian and Atkinson, 2005).

The whole package is contained in the tar file *bb_toolbox_vx.x.tar.gz* (where *x.x* indicates version number). To unpack type '> tar -zxvf *bb_toolbox_vx.x.tar.gz*'. The resulting folder will contain two sub-directories (*/src* and */validation*), whose content will be illustrated in the following, and this README file in pdf format.

1a) CONVENTIONS

In this brief tutorial the following conventions are assumed:

- folders are preceded by slash symbol (*/*) and reported in *italics*;
- your bin folder (*~/bin*) is already included in your PATH environmental variable;
- files (source, input, etc.) are reported in *italics*;
- terminal command lines are preceded by the '>' symbol.

2) SOURCE CODE AND COMPILING

The source code is located under */src*. It is written in both Fortran95 and C, and consists of fifteen files:

- *main_bbtool.f90* – main program;
- *module_bbtool.f90* – global variables and modules;
- *module_interface.f90* – explicit interfaces for external subroutines and functions;
- *geometry.f90* – source-station related quantities computation;
- *io.f90* – input/output facilities;
- *error.f90* – error and warning conditions;
- *fourier.f90* – frequency-domain operations based on FFT/IFFT;
- *interpolation.f90* – spline and polynomial interpolation;
- *coda.f90* – S-to-S multiple scatter model equations resolution;
- *scattering.f90* – scattering parameters computation;
- *convolution.f90* – convolution of source-time function with scattered coda;
- *random.f90* – random numbers and probability distribution functions generators;
- *composition.f90* – low and high frequency seismograms coupling;
- *source.f90* – source-time function definition;
- *ray3DJHfor.c* – ray-tracing subroutine;

Beside these, there is a script called *compile.csh*. Let us first see briefly how the code works.

The code can be roughly divided into three chunks:

- In the first one, input parameters are read (see next paragraph), P and S-wave arrivals for the given velocity model and fault-stations configuration are computed using the ray-tracing code written by J.Hole, and scattering parameters and proper source-time function (hereafter stf) are set up. The code can handle both 1D and 3D velocity models, as well as point and extended sources. Several stfs are available (see next paragraph), whose characteristics are based on source-scaling relations; moreover, if extended fault models are specified, the initial stf is scaled in amplitude for each station according to the empirical Green's function (EGF) approach described by Irikura and Kamae (1994);
- The second section represents the core of the Broad-Band Toolbox code where, looping over each station, scattered seismograms are computed using the technique of Zeng et al. (1991, 1993), then convolved with the stf and finally combined with the low frequency seismograms according to the procedure described in Mai and Beroza (2003);

- The third and last section is concerned on output operations, such as hybrid broad-band synthetics and log-file disk storage.

As mentioned before, the user can choose to run the code also only to merge low- and high-frequency seismograms previously computed. Although explicitly written to handle EXSIM high-frequency seismograms (containing only S-waves contributions), the code can be easily extended to work with other high-frequency computation packages. In this case, the above workflow description changes into:

- A first chunk where input parameters (see next paragraph) and seismograms are read, and P and S-wave arrivals for low-frequency synthetics are computed, using the ray-tracing code;
- In the second part, looping over each station, the two contributions are combined according to the procedure of Mai and Beroza (2003), accounting for any time-shift between low- and high-frequency synthetics;
- The third and last section remains basically the same, concerning output operations, such as hybrid broad-band synthetics and log-file disk storage.

As stated above, the code considers explicitly possible time-shifts between the two contributions. This may happen quite often, due to non-identical source-station(s) geometrical conditions, fault parameterization or velocity models, etc. To overcome this inconvenience, our code estimates these time-differences for each station and shifts high-frequency seismograms accordingly. This means that arrival times for the low-frequency contributions are always taken as a reference. Arrival times for the high-frequencies are detected using a first-break detection algorithm.

Depending on the size of your spatial domain and the grid-step adopted (see next paragraph and appendix A), the ray-tracing code may require large cpu-time: therefore we decided to parallelize it using OpenMP directives. In this way it is possible to have high performing code also on simple multi-core workstations. Parallelization of the loop over stations has been implemented as well, thus resulting in a sensibly faster code also when lots of stations are used.

This code doesn't require specific libraries and has been tested on Linux and Mac machines using Intel Fortran and C compilers, freely available for Linux operating systems on the web¹. It can be compiled using the script *compile.csh*, after setting properly the compiler flags for your machine (these can be important because strongly influence code run-time speed; please refer to the compiler documentation for further details). As output, an executable named *BBToolbox.exe* is produced.

1 <http://www.intel.com/>

After copying this executable file into the ~/bin folder and setting up all the input files needed (paragraph 3), the user can start jobs by calling directly the executable file and following the instructions or by using his/her own shell script.

Note: the OpenMP parallelization affects sensibly computation times only when large computational domains and/or large number of stations are used.

3) INPUT

Under the */validation* folder, one can find some sample input files. Basically the code requires a main input file with the following entries described below:

```
-----  
/* MODALITY FLAG: [0] LF-HF MERGING, [1] LF-SCATTERING, [2] LF-ISOCHRONE */  
1  
/* OUTPUT DIRECTORY */  
~/run/broadband_test/  
/* VELOCITY MODEL FILE (3D MODEL OR 1D MODEL) */  
velmodel.par  
/* STATIONS FILE REPORTING [X-Y] COORDINATES, FILENAMES AND PARAMETERS */  
stations.dat  
/* OPTIONAL STATIONS FILE REPORTING FILENAMES - ONLY FOR MODALITY = 0 */  
hf_stations.dat  
/* FAULT MODEL TYPE: [POINT], [EXTENDED FAULT-MODEL FILE] */  
extended coord.txt  
/* HYPOCENTER COORDINATES [X-Y-Z] IN KM */  
265.0 183.0 10.0  
/* GRID DEFINITION [X-Y-Z] FOR RAYTRACING: "NEAR-SIDE", GRID-SPACING (IN KM) */  
190.0 130.0 0.0 5.0  
/* GRID DEFINITION [X-Y-Z] FOR RAYTRACING: "FAR-SIDE" (IN KM) */  
480.0 250.0 50.0  
/* SCATTERING PARAMETERS FILE */  
scattering.dat  
/* TARGET MATCHING FREQUENCY & BANDWIDTH FOR SEARCH */  
0.35 0.15  
/* EVENT MAGNITUDE */  
7.7  
/* DOMINANT SOURCE MECHANISM [SS RS NS AL] */  
ss  
/* SOURCE TIME FUNCTION [TRI BOX YOF DREG LIU USER-DEF] */  
dreg  
/* VERBOSE MODE [ON OFF] */  
on  
-----
```

Entry 1: flag for operational modality. In the current release only the following values will be accepted: 0 – broad-band synthetics obtained by merging of pre-computed low-frequency and high-frequency contributions; 1 – broad-band synthetics computed using Zeng’s multiple S-to-S scattering theory. (Isochrone method will be available in a next release).

Entry 2: output directory for broad-band seismograms.

Entry 3: velocity model file. This model is used to compute P- and S-wave arrival times for low-frequency (i.e. deterministic) contributions. Plain ASCII files are assumed to contain 1D models (see appendix B). To avoid huge files, 3D models can be passed only as binary files. Please note that the code recognizes binary files only if they end with the *.bin* extension. In both cases V_p , V_s and ρ are given in km/s and g/cm^3 . See appendix B for information about formats.

Entry 4: ASCII file containing stations coordinates (in km, in the Broad-Band Toolbox reference system as described in appendix A), their names, format specifications and other parameters. See below for details.

Entry 5: optional ASCII file for high-frequency seismograms. See below for description. This entry is necessary only if flag for operational modality is set to 0.

Entry 6: fault model flag. Valid specifications are: point – point source model; extended – extended source model. In this last case another entry consisting of an ASCII file is expected along the same line: this file must contain sub-faults (1 x 1 Km) coordinates (x y z), by which the fault plane has been discretized. First line contains the total number of subfaults. Coordinates are referred to sub-faults centers.

Entry 7: hypocenter coordinates in km. Note that depth is positive.

Entries 8 and 9: spatial domain extension of the grid for the ray-tracing subroutine (in km). The user has to define two opposite corners of the volume and the grid step. Please note that this volume must contain all the receivers, the hypocenter and the sub-faults (if provided). More in appendix A.

Entry 10: scattering parameters file. This will be described in detail in the following.

Entry 11: target frequency and bandwidth are the parameters for spectral matching, i.e. the optimal matching frequency is sought in the range [target-bandwidth target+bandwidth].

Entries 12 and 13: magnitude and dominant source mechanism of the earthquake used previously for the long-period calculations. These entries are used in the computation of the appropriate stf, based on source-scaling relations. The following dominant mechanisms are available: strike-slip (ss), reverse-slip (rs), normal-slip (ns) and unknown (al).

Entry 14: source-time function (stf) to be used during convolution with the scattering Green's function; the user can choose between boxcar (box), triangular (tri), yoffe (yof), dreger (dreg), liu (liu) and user defined stf in Rob Graves' format (an example is shown through the *example_udf_stf.txt* file located under */validation*).

Entry 15: enable/disable verbose output (scattered seismograms, convolved scattered seismograms and source-time function are also stored on disk).

Here below we describe input files for stations, corresponding to entries 4 and 5 in the main input file:

```
-----  
/* STATIONS FILE FOR BROAD-BAND COMPUTATION CODE (P.M. MAI & K.B.OLSEN) */  
/* STATIONS COORDINATES ARE IN THE X-Y SYSTEM REPORTED IN FIG.1 OF APPENDIX A */  
  
/* INPUT DIRECTORY */  
~/data/synth/lf_files/
```

```

/* FILES FORMAT [RGF BIN CMP 3SF] */
  RGF

/* FILES EXTENSION OR BINARY FILE NAME */
  .fp .fn .fv

/* X      Y      NAME  Vs  Rho  Kappa */
  346.0  230.0  s01_m1  1.8  -1   -1
  425.0  236.0  s02_m1  -1   -1   -1

```

Entry 1: directory where input low-frequency seismograms are located.

Entry 2: format specification for these files. Only the following formats can be read: RGF – Rob Graves’ format; BIN – binary format; CMP – Compsyn (Spudich and Xu, 2003) format; 3SF – plain ASCII file format. For their description see appendix C.

Entry 3: files extension or binary file-name. If RGF or CMP formats have been specified, then the code will expect three entries corresponding to the low-frequency files extensions (i.e. one component for each file). In the 3SF format case, only one extension is expected (i.e. three components one the same file). For BIN format, the user must provide the whole file-name, including its extension.

Entries 4 ... N: stations coordinates in the Broad-Band Toolbox system (appendix A), their name (please be sure that, except for the binary case, NAME+EXTENSION must give the complete name of the file referred to that station), S-wave speed at the site (Km/s; Vs30 if available), density (g/cm^3) and kappa (adimensional) for the site. If -1 is specified, then values defined internally in the code will be used. This means that Vs and rho values will be derived from the input velocity model (that is, they will be the same for each station if a 1D model has been specified), while for kappa a common constant value will be adopted instead.

In the example above, for the first station, Vs is 1.8 Km/s, while rho assumes a value based on the velocity model. For the second station, Vs and rho are both derived from the velocity model. Kappa value is the same for both stations.

The optional stations file (entry 5 in the main input file) looks like:

```

/* OPTIONAL STATIONS FILE FOR BROAD-BAND COMPUTATION CODE (P.M. MAI & K.B.OLSEN) */
/* STATIONS NUMBER MUST BE THE SAME AS THAT ON FIRST STATIONS FILE */

/* INPUT DIRECTORY */
~/data/synth/hf_files/

/* FILES FORMAT [RGF BIN CMP 3SF] */
  3SF

```

```
/* FILES EXTENSION OR BINARY FILE NAME */
  .3sc
```

```
/* NAME */
  ITGG
```

This file must be provided only if the code is operating in modality 0, i.e. merging of pre-computed low- and high-frequencies contributions. In this case, additionally to the input folder, format and extension(s) specifications following the restrictions above, the user must supply only the station name. Note: also here one has to observe the rule that, excluding the binary case, NAME+EXTENSION must return the complete name of the file to be opened. Obviously, the number of stations here must be equal to that on the mandatory stations file.

Another important secondary input file is that containing scattering parameters. It has been introduced to give greater flexibility to our code. Here below an example is shown. Please note that generally parameters can assume default values or user-defined values, and some of them can be drawn from different probability distribution functions (uniform and normal).

```
/* SCATTERING PARAMETERS FOR THE BROAD-BAND TOOLBOX */
/* -----THE FOLLOWING PARAMETERS CAN BE CHANGED----- */

4518 \* iseed - seed number for scattering wavelets -> [-1]:random value, [0]:default value, [int]:user-defined
0 \* s_seed - seed number for medium properties -> [-1]:random value, [0]:default value, [int]:user-defined
180.0 \* Q - Q for the coda waves -> [0.0]:default value, [1.0]:drawn from normal distribution, [real]:user-defined
0.03 \* scatcoeff - scattering coefficient -> [0.0]:default value, [1.0]:drawn from uniform distribution, [real]:user-defined
0.04 \* kappa - kappa at the site -> [0.0]:default value, [1.0]:drawn from uniform distribution, [real]:user-defined
0.45 \* fdec - see equation  $Q=Q_f*f*fdec$  -> [0.0]:default value,[real]:user-defined
0.2 \* hpass - highpass corner of the cosine filter -> [0.0]:default value,[real]:user-defined

/* -----WARNING: THE FOLLOWING PARAMETERS SHOULD NOT BE CHANGED----- */

500 \* ncoda - number of points computed for the coda envelope -> [0]:default value,[int]:user-defined
512 \* nfcoda - the corresponding fft points -> [0]:default value,[int]:user-defined
1500 \* nscat - number of scattering wavelets -> [0]:default value,[int]:user-defined
0.001 \* merr - tolerance of the coda envelope -> [0.0]:default value,[real]:user-defined
0.03 \* abscoeff - absorption coefficient -> [0.0]:default value, [1.0]:drawn from uniform distribution, [real]:user-defined
0.1 \* trans - trans. bandw. of the filter (begins at hpass-trans, ends at hpass) -> [0.0]:default value,[real]:user-defined
```

Entry 1: starting seed number for random number generator used during the scattered seismograms computation. Seed number changes automatically for each station.

Entry 2: starting seed number for random number generator optionally used to compute some medium properties (kappa, scattering coefficient, etc. See next lines).

Entry 3: attenuation value. This Q is the Q_0 term in the equation $Q=Q_0f^{f^{dec}}$, where f is frequency. Default, user-defined or drawn from normal distribution values are possible.

Entry 4: scattering coefficient parameter: elastic attenuation of the medium due to scattering effects. It mainly controls the S-wave coda decay and the “strong motion duration” of the resultant synthetics. Default, user-defined or drawn from uniform distribution values are possible.

Entry 5: kappa parameter: site specific high-frequency attenuation term, which mainly determines the high-frequency diminution of the spectrum. Default, user-defined and drawn from uniform distribution values are possible. Note: only kappa values not specified by the user in the mandatory station file (-1 flag) are affected at this stage.

Entry 6: frequency decay term (fdec) in the equation $Q=Q_0f^{f^{dec}}$. Together with the above attenuation parameter, this factor controls the frequency-dependent attenuation. Default or user-defined values are possible.

Entry 7: lower band limit for coda waves computation. Default or user-defined values are possible.

The remaining section of this file contains parameters that normally don't require to be changed.

Entry 8: number of points for the coda envelope. Default or user-defined values are possible.

Entry 9: frequency points for the coda envelope. Default or user-defined values are possible.

Entry 10: number of scatterers. Default or user-defined values are possible.

Entry 11: tolerance of the coda envelope. Default or user-defined values are possible.

Entry 12: absorption coefficient. Default, user-defined and drawn from uniform distribution values are possible.

Entry 13: transition bandwidth for the high-pass filter. This value controls the steepness of the filter, since it starts at [hpass – trans] and ends at [hpass]. Default or user-defined values are possible.

Besides all these files, it is necessary to provide also the low-frequency seismograms (previously computed) in velocity (m/s). If modality flag is set to 0, high-frequency seismogram in velocity (m/s) will be needed as well. Note that inside the code there is no check for units consistency. Moreover it is assumed that ALL the low-frequency and high-frequency files have the same number of points and same time-sampling. If this condition is not respected, an error message will be generated and the code will stop abruptly.

Warning: before running your computations, be sure that all the needed input files are in the current directory.

4) OUTPUT

Most of the output files are stored in a user-specified directory (note: you have to create this folder before running the code). Basically these are the hybrid broad-band seismograms (velocity, in m/s), having extension *.hyb* and root-name as the low-frequency ones (however, if binary input has been selected, one single binary file named *BBhyb.bin* will be produced in output). Their format is described in appendix C. Additionally, if verbose mode is allowed in the main input file and modality flag is set to 1, here are stored also the scattered seismograms (*.ocd* extension), the scattered seismograms convolved with the source-time function (*.ccd* extension) and one file with the source-time function (*.stf* extension). Also in this case, all files have root-names equal to stations names specified in the mandatory stations file. Note that only one *.stf* file is produced, even if an extended source model has been selected. This because, according to the EGF approach quickly reported in paragraph 2, all the source-time functions have the same shape but different amplitudes. Sample output files can be found under */validation*.

In any case, in the current directory a log-file called *run.log* is saved instead. An example is shown here below.

*** LOG-FILE FOR BROADBAND HYBRID COMPUTATION CODE (P.M.MAI & K.B.OLSEN) ***

Current date and time: 09/02/2009 12:05:07.929

Code running in modality: 1

----- RAYTRACING PROPERTIES -----

Domain dimensions [x y z] : [290 120 50]

Domain discretization (pts) : [59 25 11]

----- MEDIUM PROPERTIES -----

Num. of layers for 1D model : 201

Resampled 1D model has been written on a separated log-file

----- SOURCE PROPERTIES -----

An extended-source model of 3072 subfaults has been used

Hypocenter position at : [75.000 53.000 10.000]

----- STF PROPERTIES -----

Npts in STF : 984

Time length : 15.000 Time-step : 0.015

----- MATCHING PROPERTIES -----

Target frequency : 0.350 Bandwidth for search : 0.150

----- SCATTERING PROPERTIES -----

Source density : 2.780 Source S-wave speed : 3.700

Maximum frequency : 20.000 Start time for coda waves : 0.000

Scattering wavelets : 1500 Number of pts coda envelope : 500

Scattering fft points : 512 Seed number for parameters : 461

Coda tolerance : 0.001
Absorption coefficient : 0.030 Scattering coefficient : 0.030
Q0 value : 180.000 Q frequency decay factor : 0.450
High-pass corner frequency : 0.200 Bandwidth for filter : 0.100

----- TIME-SERIES PROPERTIES -----

Npts in time-series : 16384
Time length : 249.997 Time-step : 0.015

----- STATIONS PROPERTIES -----

Total number of stations : 2

Station number: 1 [s01_m1]
Kappa value : 0.040
Matching frequencies [P N V]: 0.216 0.268 0.200
Site density : 2.670 Site S-wave speed : 3.170
Average S-wave speed : 3.646 Hypocenter distance : 94.181
P-wave travelttime : 15.433 S-wave travelttime : 26.409

.
.

Computation terminated in: 3.023611 secs

This file reports most of the parameters used during the computation, those referred to the medium, those in common for all the stations and those variable for each station. All the lines are self-explanatory. A similar log file is printed out for a modality flag equal to 0.

In conjunction with this, if a 1D velocity model has been specified and verbose output is allowed, other two output files are produced, containing the P- and S-wave speed profiles as interpolated from the input model using the ray-tracing grid-step.

5) VALIDATION

The */validation* folder contains some validation tests that the user should run before executing his/her own jobs. The user can also use the sample files enclosed as a reference while creating personalized input files. There are two sub-directories, named */point* and */extended*; each of these has its own input and output files. Moreover three Matlab scripts are given, named *CompSeisB.m*, *ScatSeis.m* and *sfft.m*. The first two can be used to plot respectively the low-frequency and the hybrid broadband seismograms and their spectra (both in velocity and acceleration), and the verbose-mode output files (scattered seismograms, convolved scattered seismograms and source-time function). The *sfft.m* script is needed only to compute the amplitude and phase spectra (using a Fast Fourier Transform algorithm).

To validate, simply copy all the input files in another directory and then run your calculations. After that, compare your results with those contained in the proper */validation* sub-folders. Just as an example, here we report a sample plot showing how our broadband seismograms appear:

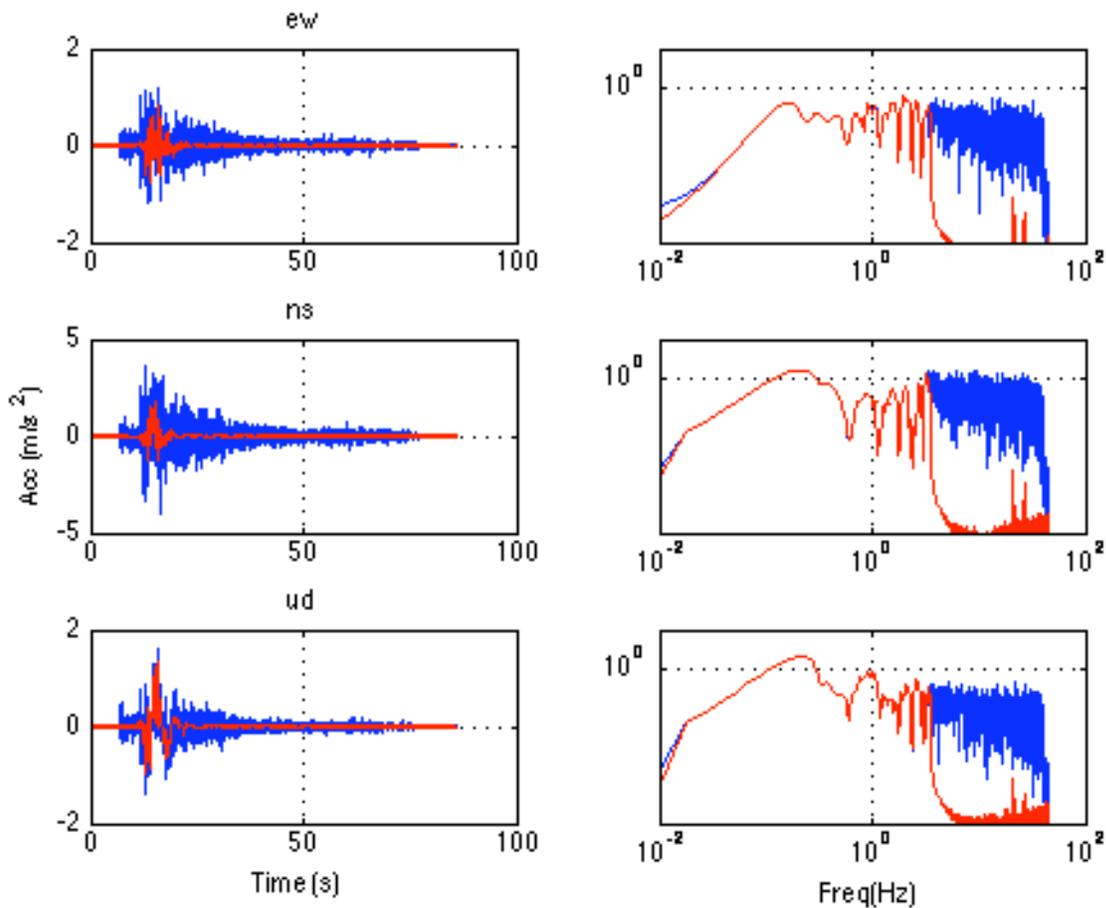


Fig. 1 – Example of broadband synthetics (in blue) and their spectra. In red, the low-frequency components are shown.

APPENDIX A

The volume defining the spatial domain of the Broad-Band Toolbox must be specified in the main input file (paragraph 3). Below a sketch, showing the adopted convention:

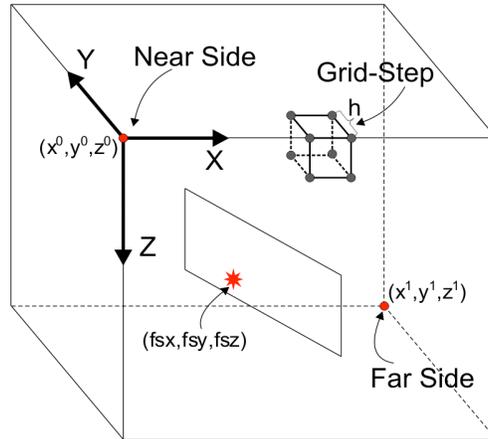


Fig.1 – Volume defining the spatial domain in our code (see also paragraph 3). The elemental cube delimited by eight grid-nodes, with side-length equal to the grid-step, represents how the whole medium is discretized by the ray-tracing subroutine. The total number of grid nodes is given by $n_x * n_y * n_z$, each one representing the number of points along the relative direction. The plane inside the volume represents a fault-plane with the hypocenter (red star) located on it.

As already mentioned, remember to allow your volume to include all the stations and, eventually, the sub-faults composing the extended fault model.

APPENDIX B

The input 1D velocity model should be defined through an ASCII file composed of two header lines followed by one line for each depth level in this format: $z \ V_p \ V_s \ \rho$ (respectively in km, km/s, km/s and g/cm^3). For an example see file *velmodel.dat* under */validation/point*. Note that to introduce a sharp discontinuity, the user has to specify two different property values at the same depth, otherwise a gradient will result (this convention is equal to that followed in the Compsyn package).

The input 3D velocity model is instead a binary file with three entries for each record (V_p , V_s and ρ , always in km/s and g/cm^3 respectively). First, values along the x-direction must be specified, then follow y-direction values and last z-direction ones (starting from the surface, i.e. $z=0$). In practice our code reads the binary file through the following nested loop:

```
do k=1,nz
  do j=1,ny
    do i=1,nx
      read(unit, rec=(k-1)*ny*nx+(j-1)*nx+i) vp(i,j,k),vs(i,j,k),rho(i,j,k)
    enddo
  enddo
```

```
    enddo
enddo
```

Note that the total number of records in the binary file must be equal to the number of points contained in the volume under study (see appendix A). It is a task left to the user to verify the consistency between the input binary file and volume box dimensions.

APPENDIX C

Here below we describe the formats used in our code for input/output files containing time-series. As reported before, the Broad-Band Toolbox is able to read several formats, also binary.

The first format is the RGF (Rob Graves' format): it consists of one plain ASCII file for each component of the ground motion (i.e. three files for each station). Two header lines (the first of which skipped by the code, the second reporting number of points and time-step) are followed by numerical values arranged into six columns, with time increasing from the left to the right; each row is separated from the others by a single space line:

```
s01  ver
22728  0.011
  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000
  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000
  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000  0.00000E+000
                                     :
                                     :
```

The CMP format is identical to the Compsyn output files format. These are ASCII files (again, one for each component) having a long header section (100 lines) and followed by numerical amplitude values. For further details, see the Compsyn package documentation.

The last ASCII format is the 3SF: one single file containing the three components and the time vector. There are no header lines.

For waveforms passed through a binary file, these must be stored sequentially for each site, following the same order as specified in the station file. Our code reads this input in the following way:

```
do k=2,npts+1
  read(1,rec=(station-1)*npts +k) seis(k,1),seis(k,2),seis(k,3)
enddo
```

In practice each line contains the three components of motion at each time-step for each station. Note that the first record contains the number of points and time-step (constant for all the stations). The corresponding output binary file has one more entry (time value) in each record line.

WHAT TO DO IF THE BROAD-BAND TOOLBOX FAILS?

This section deals with some common problems the user may encounter when using our code. Most simple but typical problems are those related to code crashes due to incorrect input files. Our code has already some error-handling subroutines that may help the user to understand and correct it. In many cases, the compiler itself can give useful hints. We recommend the user to check carefully all the input files. Particular attention should be given to correct input seismograms formatting procedure, since it can cause of subtle mistakes.

A major problem can be represented by the ray-tracing subroutine failure. It has been noted that this piece of the code is rather sensible to the grid step adopted, resulting in abrupt segmentation-fault errors. We are trying to fix this bug and hope to remove it for the next release. What we suggest, in such cases, is to increase the grid spacing first, and eventually the spatial domain size then. Increasing the grid-step is not an important issue in many cases, especially when 1-D velocity models are used: in these circumstances, the loss of resolution in travel-time arrivals can be compensated by the bilinear interpolation performed inside the code.

Another problem related to the ray-tracing subroutine can be observable in the final hybrid seismograms, where the user may experience a noticeable mismatch between the real P- or S- waves arrival times and the computed ones. This is remarkable when the operational modality is equal to 1. Such errors are due to an incorrect geometry set-up, that is, relative positions of fault (hypocenter) and stations in the Broad-Band Toolbox reference system are wrong. We encourage the user to check always the correspondence between the original fault-stations system (when the low-frequency contributions are computed) and that introduced for our code.

Sometimes the user may find the final broad-band seismograms appear strange. This case includes also when amplitudes are too low or when the scattered contributions are too short in time than expected, etc. Causes can be multiple, but typically these problems may be due to an incorrect matching frequency. It is hard to define here what is correct or not, since the matching frequency should be fixed according to the low-frequency seismograms characteristics. These include not only the physical validity limit of the simulations, but also their spectral content. For instance, we have found that a constant rise-time overall the fault can cause holes in the spectra and this can give problems during the merging operations. An incorrect matching frequency often results in lower than expected amplitude values. Other reasons can be linked to bad choices for scattering parameters: at this purpose, the *run.log* file can result rather useful for a posterior checking. Inspection of the secondary output files (available only in verbose modality) can help as well.

IMPORTANT:

This is a research-purpose software, under constant development and improvement. It may contain errors.

Please report any bugs to Walter Imperatori, e-mail: imperatori@tomo.ig.erdw.ethz.ch

REFERENCES

Irikura, K., Kamae, K. (1994): Estimation of strong ground motion in broad-frequency band based on a seismic source scaling model and an empirical Green's function technique, *Annali di Geofisica*, Vol. XXXVII, 6, pp. 25-47.

Mai P., Beroza G. (2003): A hybrid method for calculating near-source, broadband seismograms: application to strong motion prediction, *PEPI*, 137, pp. 183-199.

Motazedian D., Atkinson G. (2005): Stochastic Finite-Fault Modeling Based on a Dynamic Corner Frequency, *BSSA*, Vol. 95, 3, pp. 995-1010.

Spudich P., Xu L. (2002): Documentation of software package COMPSYN sxv3.11: programs for earthquake ground motion calculation using complete 1-D Green's functions, published in a cd accompanying *International Handbook of Earthquake and Engineering Seismology*, IASPEI, Academic Press.

Zeng Y. (1993): Theory of scattered P-and S-wave energy in a random isotropic scattering medium, *BSSA*, Vol. 83, 4, pp. 1264-1276.

Zeng Y., Su F., Aki K. (1991): Scattering wave energy propagation in a random isotropic scattering medium. 1. Theory, *J. Geophys. Res.*, 96, pp. 607-619.