BBTOOLBOX

Separation of Scattering and Intrinsic Attenuation by Envelope Inversion

Version 1.7 beta

User's Manual

Walter Imperatori
Swiss Seismological Service, ETH Zurich
walter.imperatori@sed.ethz.ch
https://github.com/flurinursli/bbtoolbox

Contents

1 Introduction

BBToolbox is a numerical package to efficiently compute broadband synthetic seismograms. As other similar packages [e.g., 1, 2, 3, 4, to cite a few], it merges external pre-computed deterministic low-frequency (typically in the 0-1/2 Hz frequency band) and (semi-)stochastic high-frequency timeseries, the latter often based on fast numerical techniques suitable for 1D media as frequency-wavenumber integration, ray theory, etc. The low-frequency synthetics represent the deterministic component of a wave field, for which source rupture and propagation characteristics are controlled by larger-scale features (e.g. macroscopic layering of the crust, low-resolution 3D sedimentary basins, first-order fault plane geometry and slip distribution). On the other hand, high-frequency synthetics approximate the contribution of smaller-scale features that we can describe only in a statistical sense (e.g. velocity heterogeneity in the crust, irregularity of both fault plane and rupture process).
BBToolbox can model timeseries generated by either point- or extended-sources up to 20-30 Hz by impinging on the isochron theory [5], the radiative transfer theory [6] and the RIK source model of [7].
The code is fully parallelized, leveraging on MPI and OpenMP directives, and can be run on desktop machines or supercomputers for extensive scenario simulations.
The current release of *BBToolbox* builds upon older versions dating back to [8] and [9], although the source code was completely rewritten and many new features were added. A first preliminary version was presented during the PSHA Workshop (Lenzburg, 2017). However, it should be noted that several of these features haven't been fully tested yet and a thorough validation is still in progress. We therefore recommend extra caution when interpreting results.

1.1 Overview of the code

The program flow of *BBToolbox* is schematically shown in Figure 2.1 below. Users are recommended to read this section carefully in order to correctly understand how the code works and the role of each program unit.

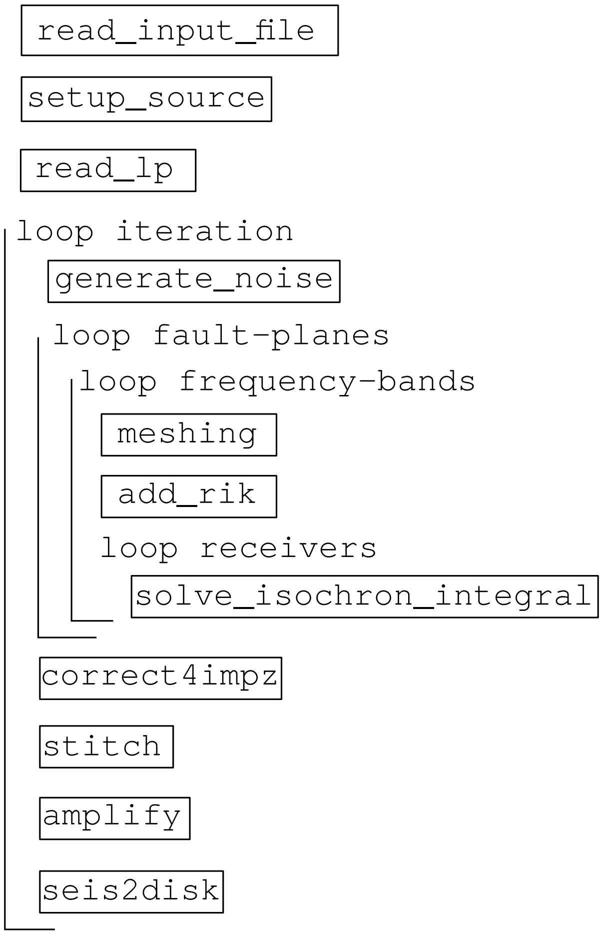


Figure 1.1: Schematic workflow (top to bottom) and subroutine calls in *BBToolbox*.

The workflow is virtually the same for both point- and extended-source calculations, with subroutine `add_rik` being the only exception since it is not called in the former case. A series of loop are present: the first, *loop iteration*, encompasses code called for each set of simulations where the initial seed number is different. This seed number controls the appearance of coda waves and several extended-source properties. The second and third loop (*loop fault-planes*, *loop frequency-bands*) repeat the same instructions for each fault plane and frequency band defined in input. Typically a user defines scattering parameters for a set of frequency bands (e.g. 1-2, 2-4, 4-8.). The last loop *loop receivers* iterates over all receivers.
Note that if the code is compiled with MPI enabled, the desired number of iterations

Subroutine	Description
`read_input_file`	Read input file, broadcast parameters to MPI tasks.
`setup_source`	Setup source model, read u.d. rupture file and broadcast parameters.
`read_lp`	Load pre-computed long-period timeseries (define max simulation time).
`generate_noise`	Generate spatially-correlated noise used to simulate coda waves.
`meshing`	Discretize source with a triangular mesh based on max resolving frequency.
`add_rik`	Define kinematic source parameters according to the RIK model (extended-source only).
`solve_isochron_integral`	Carry out isochron integration, set integration time-step.
`correct4impz`	Remove bandpass filtering effect when calculations occur at several frequency-bands.
`stitch`	Join long-period and short-period timeseries at matching frequency.
`amplify`	Apply u.d. station-specific amplification functions (optional).
`seis2disk`	Store broadband and short-period timeseries to disk.

Table 1.1: Description of subroutines shown in Figure 2.1.

is split among available MPI processes. Table 1.1 contains a short description of each subroutine listed in Figure 2.1.

1.2 Hardware and Memory Requirements

BBToolbox has, generally speaking, modest memory requirements. However, memory consumption can increase quickly with the maximum resolving frequency and fault plane(s) dimension. In any case most multi-core workstations should suffice to obtain solutions in acceptable time.
More consistent resources are recommended when a considerable number of scenarios (e.g. iteration) must be explored for large earthquake: in this case users should take advantage of the MPI parallelization by running the code on HPC clusters.

1.3 Compilers

BBToolbox is written in standard Fortran and C++, therefore it can be compiled using most compiler suites, provided the required libraries are already installed (see Section 1.4). To date, we have successfully built *BBToolbox* using the following compilers:

- **GCC**, version 7.4, 9.2, 10.1, `https://gcc.gnu.org/`

GCC can be obtained for free. Other suites may be used after the `Makefile.inc` file is modified accordingly (see Section 1.5).

1.4 External Libraries

In order to build *BBToolbox*, the following external libraries must be present in your system:

- **FFTW** (the Fastest Fourier Transform in the West), version 3.0 or higher (version 3.3.8 recommended), `http://www.fftw.org/`
- **LAPACK** (Linear Algebra Package), `http://www.netlib.org/lapack/`
- **TRNG** (Tina's Random Number Generator), `https://www.numbercrunch.de/trng/`
- **SCARF3D** (Scalable Three-Dimensional Random Fields generator), `https://github.com/flurinursli/scarf3d`
- **PROJ4** (generic coordinate transformation software), `http://proj.org`

FFTW is a widely distributed collection of fast routines to compute the discrete Fourier transform (DFT). LAPACK is another popular collection of subroutines for linear algebra. TRNG is an efficient random numbers generator, while SCARF3D is used to perturb the geometry of fault planes. PROJ4 is required to move from geographic to cartesian coordinates.

Optionally, the following library must be available:

- **MPI** (the Message Passing Interface), standard 3 or higher, `https://www.mpi-forum.org/`

MPI provides support for message passing on parallel machines and it is required to build *BBToolbox* on multi-node systems to reduce computational time for a large number of scenario simulations. A popular open-source implementation is OpenMPI (`https://www.open-mpi.org/`). Instructions on how to install these libraries can be found on

Variable	Meaning	Accepted values
`FC/C++`	(MPI) Fortran/C++ wrappers.	(any)
`OPT`	Compiler optimization flags.	(any)
`OMP`	Compiler-dependent flag to enable OpenMP parallelism.	(any)
`MPI`	Enable calls to the MPI library.	`y/n`
`ERROR_TRAP`	Enable some error checking at run-time.	`y/n`
`PERF`	Enable detailed performance measurement (requires MPI).	`y/n`
`MKL`	Enable Intel MKL version of FFT and LAPACK routines.	`y/n`
`INCL_FFTW`	Path to FFTW' *include* folder.	(any)
`LINK_FLAGS`	Flags to link against BLAS, LAPACK and FFTW libraries.	(any)
`PROJ4_PATH`	Path to the PROJ4 library.	(any)
`TRNG4_PATH`	Path to the TRNG4 library.	(any)
`SCARF3D_PATH`	Path to the SCARF3D library.	(any)

Table 1.2: Environmental variables in `Makefile.inc` and their meaning.

the respective websites, although some are present by default in many systems.

In the current version, we recommend building *BBToolbox* using MPI (building without MPI has not been tested yet. Moreover this library is required if timing is enabled (see 1.5)).

1.5 How to compile

BBToolbox can be built using **make**. The process is controlled by environmental variables defined in the provided `Makefile.inc` and summarized in Table 1.2.
The `Makefile.inc` already provides good optimization flags for GCC and Intel compilers. These can be further fine-tuned or changed to allow the use of other compiler suites. Enabling the OpenMP parallelism is recommended on most systems as it reduces considerably the time needed to compute coda waves. Note that `LINK_FLAGS` is compiler- and system-dependent. In general, we advice to disable the `PERF` flag when building *BBToolbox* as it introduces internal communication overhead. On the other hand, `ERROR_TRAP` should be enabled until the code is fully tested.

Once all flags are duly set, to build *BBToolbox* type:

```
make
```

The resulting executable, named `toolbox.exe`, will be located in the current folder. To build *BBToolbox* in debugging mode type:

```
make debug
```

In this case the code will produce a large amount of ASCII and binary files that can be used to visualize perturbed fault geometry, slip, rise-time, receiver-based coda envelopes and short-period only timeseries.
At the moment we recommend building *BBToolbox* in debugging mode since this will make much easier finding and fixing bugs.

2 Using BBToolbox

In this chapter we explain how to use *BBToolbox* and illustrate the output files it produces. The program is distributed with an heavily commented sample input file, named `input.par`. Here below we describe more in detail the meaning of each entry. A few scripts to read the output files can be found in Section 2.3.

2.1 The input file

Users control the behavior of *BBToolbox* by setting a series of parameters in the input file, which can have any name. Tables 2.1 through 2.4 list all the parameters, including a brief description and allowed values. **Note that the parsing routine is case-sensitive**. Parameters can be given in any order, also in the same line, but must be introduced by specific keywords. Long statements can be broken in multiple lines by appending the `'+'` symbol. Comments are introduced by the `'#'` symbol, instead.

Variable	Description	Value
`fmax`	Maximum frequency of hybrid synthetics	> 0
`matching frequency`	Where long- and short-period waveforms are merged	> 0
`bandwidth`	Controls width of matching filter window	> 0
`seed`	Initial seed number, controls appearance (phase) of coda wave	> 0
`samples`	Number of simulations (each characterized by a different seed)	≥ 1
`model`	Select coda wave coherency model	'lw'/'hv'
`alpha`	Controls the 'lw' coherency model	> 0
`a, b, ak, f0`	Controls the 'hv' coherency model	

Table 2.1: Input parameters introduced by keyword `coda`.

Long- and short-period timeseries are merged based on the approach of [10]. Best results are obtained when both sets of timeseries have comparable Fourier phases around the matching frequency. Setting `bandwidth` to too small values will introduce ringing in the hybrid synthetics (in our experience values between 0.5 and 1 are a good choice).
The coherency models of Luco & Wang ('lw') and Harichandran & Vanmarcke ('hv'), including the relevant input parameters, are described in detail in [11].

Variable	Description	Value
`folder`	Location of pre-computed long-period timeseries	any
`format`	Format of files containing the timeseries	'sw4' 'txt'
`variable`	Quantity represented in the timeseries	'displacement' 'velocity' 'acceleration'
`amplification`	Location with station-specific amplification functions	any

Table 2.2: Input parameters introduced by keyword `input`.

The 'sw4' file format is pretty similar to the format adopted by SW4, a popular freely-available finite-difference code, and is characterized by a short descriptive header followed by four columns. The 'txt' format does not contain any header instead.

Variable	Description	Value
`folder`	Location where the hybrid timeseries will be stored	any
`format`	Format of files containing the timeseries	'sw4' 'txt' 'paz' 'mseed'
`variable`	Quantity represented in the timeseries	'displacement' 'velocity' 'acceleration'

Table 2.3: Input parameters introduced by keyword `output`.

The 'paz' format was developed for an internal project at SED (ETH). In the more widespread 'mseed' format each component of motion stored in a multi-column ASCII

format (SLIST) supported by, e.g., ObsPy (`https://docs.obspy.org/tutorial/code_snippets/export_seismograms_to_ascii.html`).
To specify the earthquake source two different commands are provided, depending on whether the source is approximated as a point (`source`, Table 2.4) or extended (`rupture`, Table 2.5). Note that these commands are mutually exclusive (they cannot coexist) and can be present only once in the input file. Users cannot use multiple `rupture` commands to specify multiple fault plane, rather a single file containing such complex geometries.

Variable	Description	Value
`x,y,z`[a]	Location of the source in cartesian coordinates	any
`lon,lat,z`[a]	Location of the source in geographical coordinates	any
`strike, dip, rake`	Mechanism of the source	any
`m0`	Desired scalar moment	> 0
`type`	Moment rate-function	'brune'
`freq`	Set m.r.f. corner frequency	> 0

[a] These two sets are mutually exclusive

Table 2.4: Input parameters introduced by keyword `source`.

The source position can be given in either cartesian or geographical coordinates (obviously these cannot be mixed). `At the moment, only the classic Brune's omega-squared moment rate-function is supported`. Its corner frequency (fc) is given by $fc = freq/(2 \cdot pi)$.

Users are expected to provide low-resolution rupture (or slip-only) models in input. These are typically retrieved in source inversion studies. Accepted formats are fsp and srf (see `http://equake-rc.info/SRCMOD/fileformats/`). If the input model contains only a slip distribution (and the on-fault hypocenter position), *BBToolbox* will trace automatically a low-resolution rupture front by taking into account the relative position of the fault planes: the rupture will jump from plane to another at their intersection point (or line). The code will then use the procedure introduced by [7] to build high-resolution kinematic rupture models to compute short-period synthetics up to the maximum frequency of interest (see Table 2.1). Although a RIK model naturally follows the input slip distribution, users can set a minimum correlation threshold to guarantee that the two slip distributions won't be too different. Note that setting `corr` close to unity will heavily increase the computation time.
Fault plane(s) roughness is introduced in the high-resolution rupture models by following [12]. This is mainly used to introduce small-scale perturbations in the fault geometry that, in turn, produce variations in terms of strike, dip and rake and also variations in

Variable	Description	Value
`file`	File describing the (low-resolution) rupture model	*.fsp, *.srf
`roughness`	Control roughness of high-resolution fault plane(s)	< 0
`corr`	Minimum low- and high-resolution slip models correlation	$[0, 1]$
`l0, aparam, vrfact`	RIK model parameters	
`seed`	Initial seed number for RIK model	> 0
`save`	Save the high-resolution rupture model to disk	'y'/'n'

Table 2.5: Input parameters introduced by keyword `rupture`.

the source-receiver distance (the latter may result in non-negligible travel-time changes at receivers close to the fault plane, see also [13] for a discussion). Users should set this parameter to very low values if roughness is not desired.

Variable	Description	Value
`lon, lat`	Position of reference point for geographical to cartesian coordinates conversion	any

Table 2.6: Input parameters introduced by keyword `origin`.

We stress out that the `origin` keyword must be used only if station and source positions are given in geographical coordinates.

BBToolbox can accept several 1D velocity models characterized by an arbitrary number of layers. Free-surface boundary conditions are applied on top of the first layer. Physical properties can be constant or present a linear gradient inside each layer. `Note that negative gradients are not accepted and will produce an error message`. Also, some velocity models will result in triplications of travel-time curves: `in this case calculations will be carried out but their accuracy hasn't been verified yet`. In any case users must be aware that *BBToolbox* introduces always a very small gradient inside each layer to avoid numerical errors during raytracing. In our experience this does not affect the resulting timeseries.

Coda waves follows the elastic isotropic scattering model (see [6]). The parameters con-

Variable	Description	Value
`vp, vs, rho`	Physical properties (m/s, kg/m^3) of a layer	> 0
`depth`[a]	Depth (m) of layer top	≥ 0
`vpgrad, vsgrad, rhograd`[a]	Gradient of each property inside a layer	≥ 0

[a] Optional parameter (default value is 0)

Table 2.7: Input parameters introduced by keyword `layer`.

Variable	Description	Value
`gpp, gps, gss`	P-to-P, P-to-S and S-to-S scattering parameters	≥ 0
`b`	Attenuation parameter	≥ 0
`frequency`[a]	Specify the frequency band for each parameter	> 0

[a] Bands are defined with the following notation: {1,2; 2,4; 4,8; ...}

Table 2.8: Input parameters introduced by keyword `attenuation`.

trolling this model are typically defined by inversion of observed envelopes. Users can introduce as many scattering models as desired (even one for each receiver) by specifying the `attenuation` keyword multiple times.

Multiple receivers can be specified by repeating the `rec` keyword as necessary. Note that parameter `file` is very important: the input file containing the long-period waveforms for the receiver must match this string (e.g. if `file = 'joke'`, the code will look for file 'joke.txt' inside the folder indicated in Table 2.2). The same holds for the output hybrid timeseries. Parameters `velocity` and `attenuation` are used to assigned the velocity and attenuation models to each receiver.

Most of the parameters listed in Table 2.10 should be changed only by experience users to (hopefully) improve the quality of the short-period synthetic seismograms. We recommend to leave these parameters to values suggested in the table.

2.2 Execution

To run *BBToolbox*, a typical command looks like:

```
mpirun -np 32 -x OMP_NUM_THREADS=12 -bind-to socket ./inveta.exe input.par
```

Variable	Description	Value
`x,y,z`[a]	Location of the receiver in cartesian coordinates	any
`lon,lat,z`[a]	Location of the receiver in geographical coordinates	any
`file`	Label characterizing the receiver	any
`velocity`[b]	Velocity model assigned to receiver	> 0
`attenuation`[b]	Attenuation (scattering) model assigned to receiver	> 0

[a] These two sets are mutually exclusive
[b] Optional parameter (default value is 1)

Table 2.9: Input parameters introduced by keyword `rec`.

Variable	Description	Value
`pmw`[a]	Points per minimum wavelength	> 0
`avecuts`[b]	Average number of isochron cuts while computing integral	> 0
`waves`[c]	Legacy parameter	$[0, 2]$
`verbose`[d]	Control verbosity level	$[0, 2]$

[a] Recommended value: 4
[c] Recommended value: between 6 and 10
[c] This will be removed in a future version. Recommended value: 0
[c] Recommended value: 2

Table 2.10: Input parameters introduced by keyword `advanced`.

For instance, on a system featuring computing nodes equipped with 2 12-cores CPUs, the program would allocate 16 nodes.

2.3 Visualization

Folder `postprocessing/` contains a series of Matlab/Octave scripts to display the synthetic seismograms (in SW4 format) and to visualize features of the rupture models.

2.4 Sample application

This section contains a series of Figures showing sample output of *BBToolbox*. Check the provided `input.inp` file for a description of the parameters used to compute the synthetic seismograms. A more complete description will follow soon.

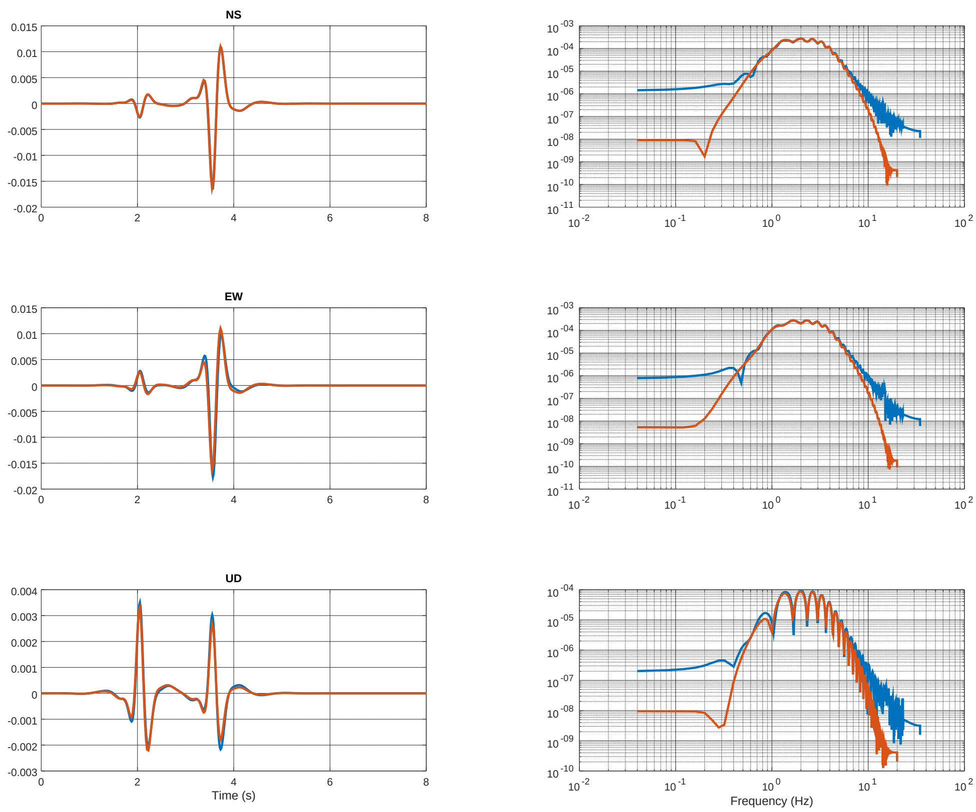
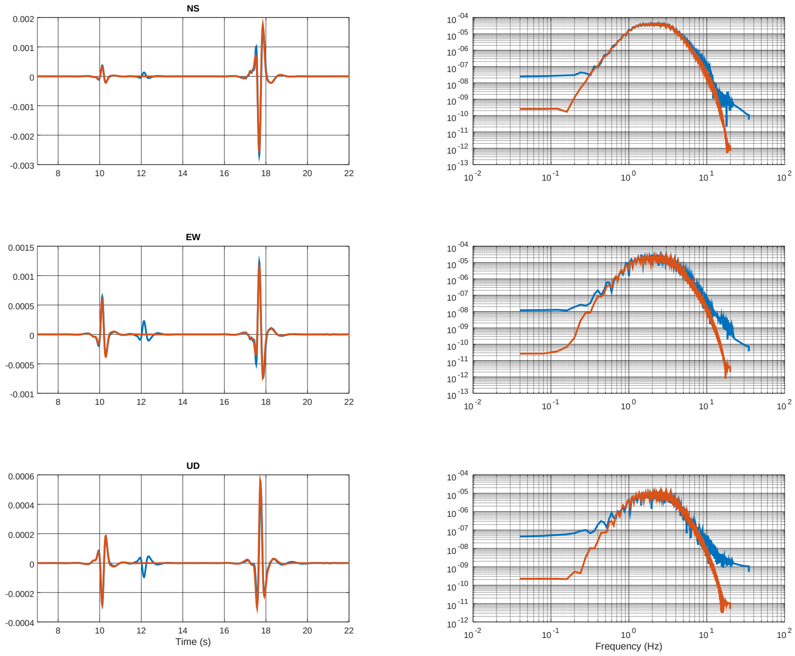
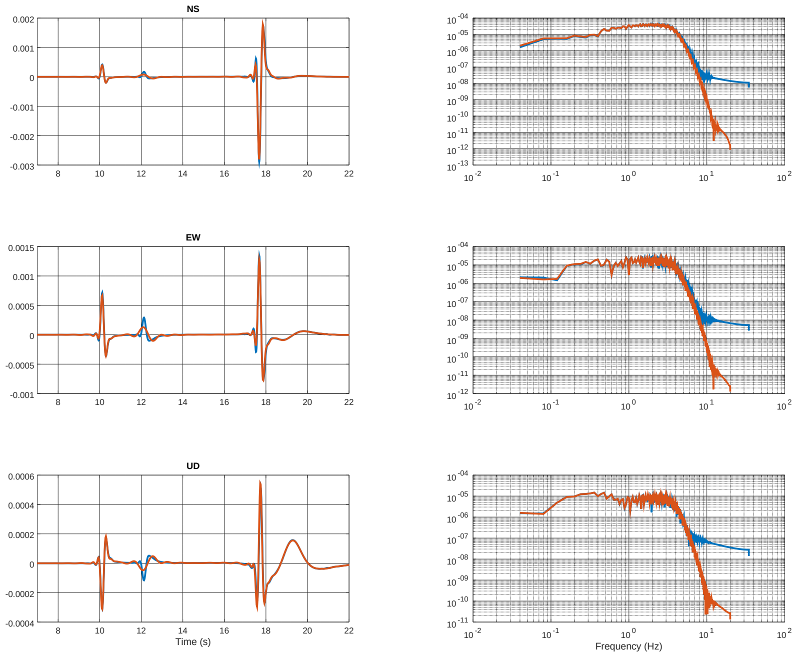


Figure 2.1: Velocity synthetic seismograms (in m/s) for a point-source (strike = 20°, dip = 120°, rake = 30°) embedded in an homogeneous half-space at a receiver located 5 km away on the free-surface. The timeseries provided by *BBToolbox* are in red, while the finite-difference solution is in blue. Source depth is 5 km and its scalar moment amounts to 10^{15} Nm (roughly corresponding to M_w 3.9). The source time-function is a Brune pulse with corner frequency at 2 Hz. Physical properties of the medium are $V_p = 3500\, m/s$, $V_s = 2000\, m/s$ and $rho = 2700\, kg/m^3$. Both sets of synthetics were band-pass filtered in the range 1-4 Hz using a two-pass two-poles Butterworth filter.

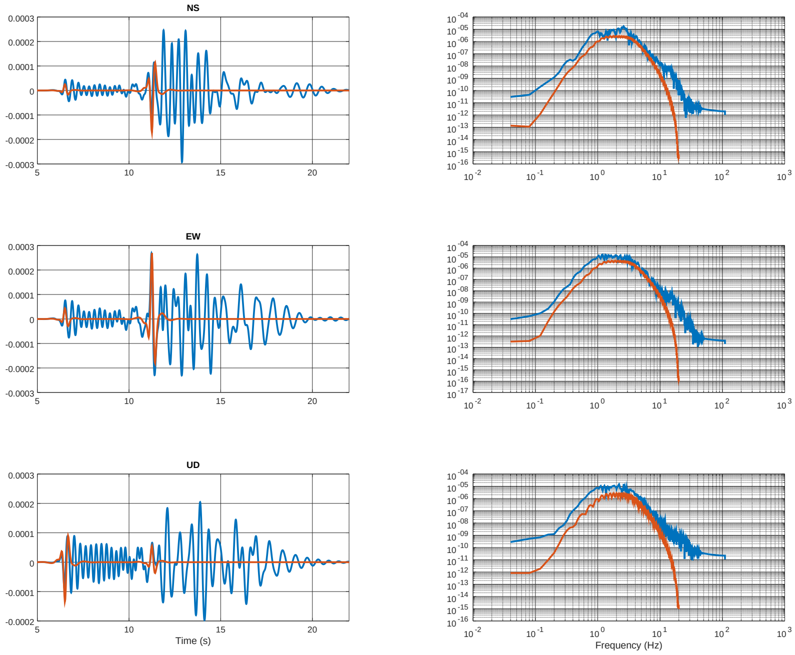


(a)

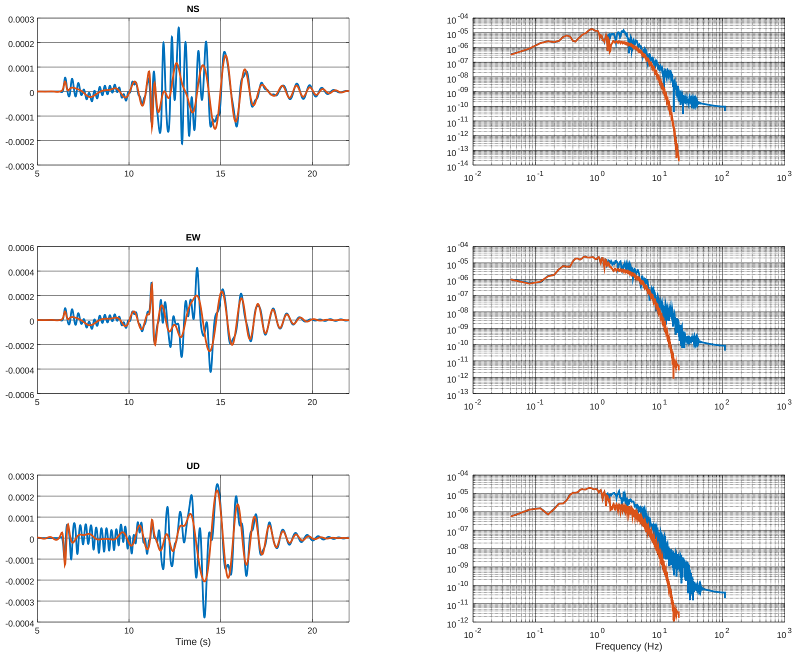


(b)

Figure 2.2: (a) Velocity synthetic seismograms as in Figure 2.1 but for an epicentral distance of 35 km. (b) Finite-difference (blue) and composite broadband synthetics (red) low-pass filtered at 4 Hz. Merging frequency is 1.5 Hz.

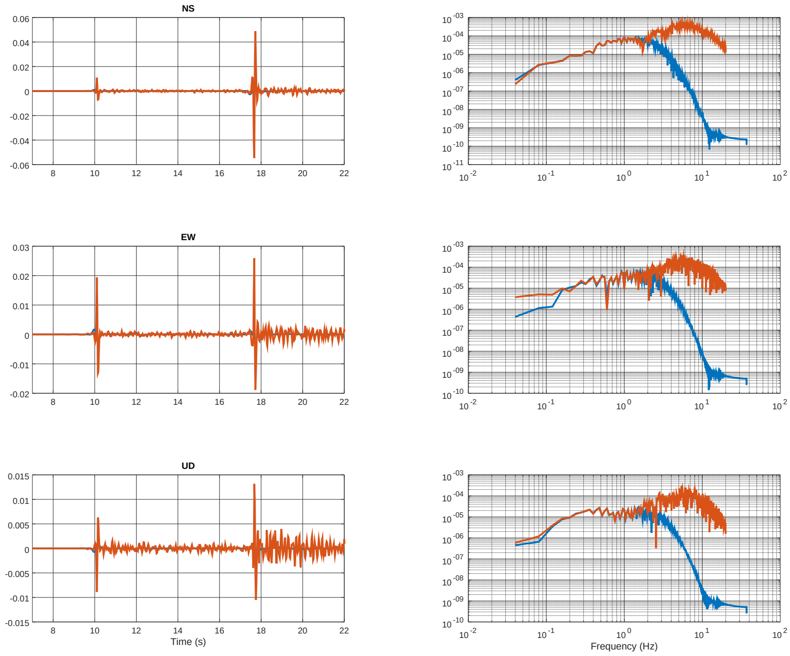


(a)

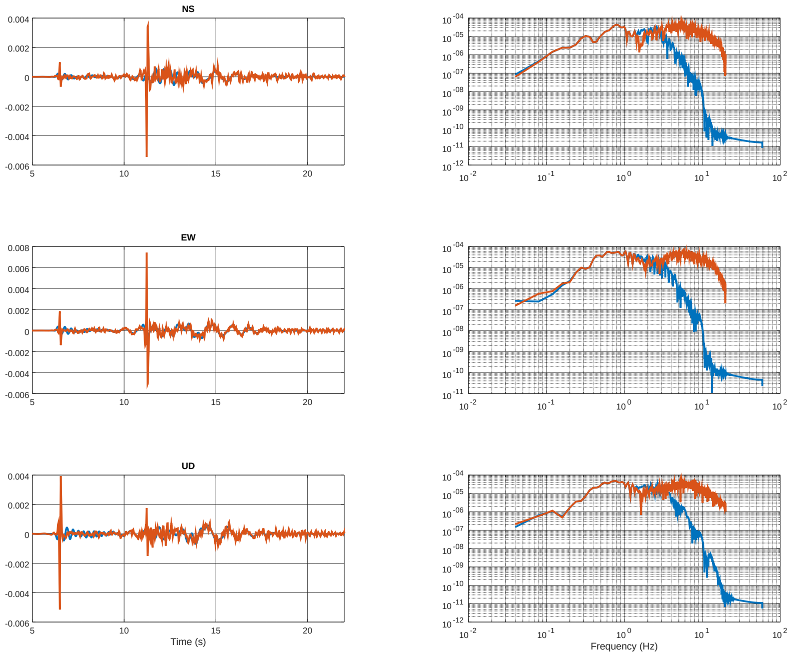


(b)

Figure 2.3: Velocity synthetic seismograms as in Figure 2.2 but for the Swiss reference model and a shallower point-source. The multiple reflections at the free-surface determined by a sharp velocity gradient appear in the composite broadband synthetics (b, in red).



(a)



(b)

Figure 2.4: Acceleration synthetic seismograms up to 10 Hz for the homogeneous (a) and Swiss reference model (b) of Figure 2.2 and 2.3, respectively. The finite-difference solution (limited to 4 Hz) is shown in blue.

Figures below refer to an extended-source test case based on the Hector Mine slip model of [14].

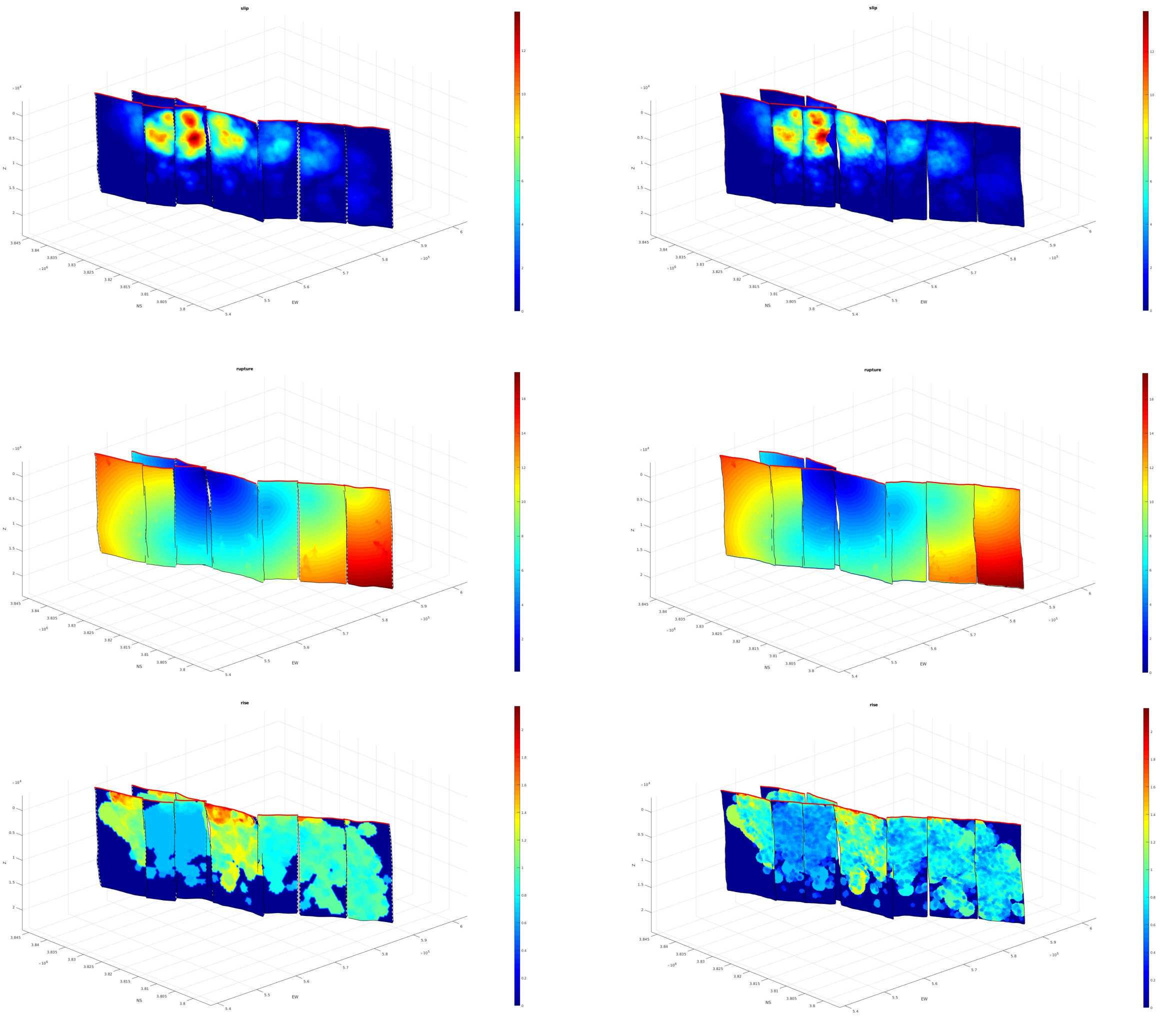


Figure 2.5: RIK rupture model in the frequency band 1-2 Hz (left) and 2-4 Hz (right) produced by *BBToolbox* based on the Hector Mine slip model of [14]. Top: slip; middle: rupture time; bottom: rise-time. Note that the low-resolution rupture front progression was computed by *BBToolbox* independently.

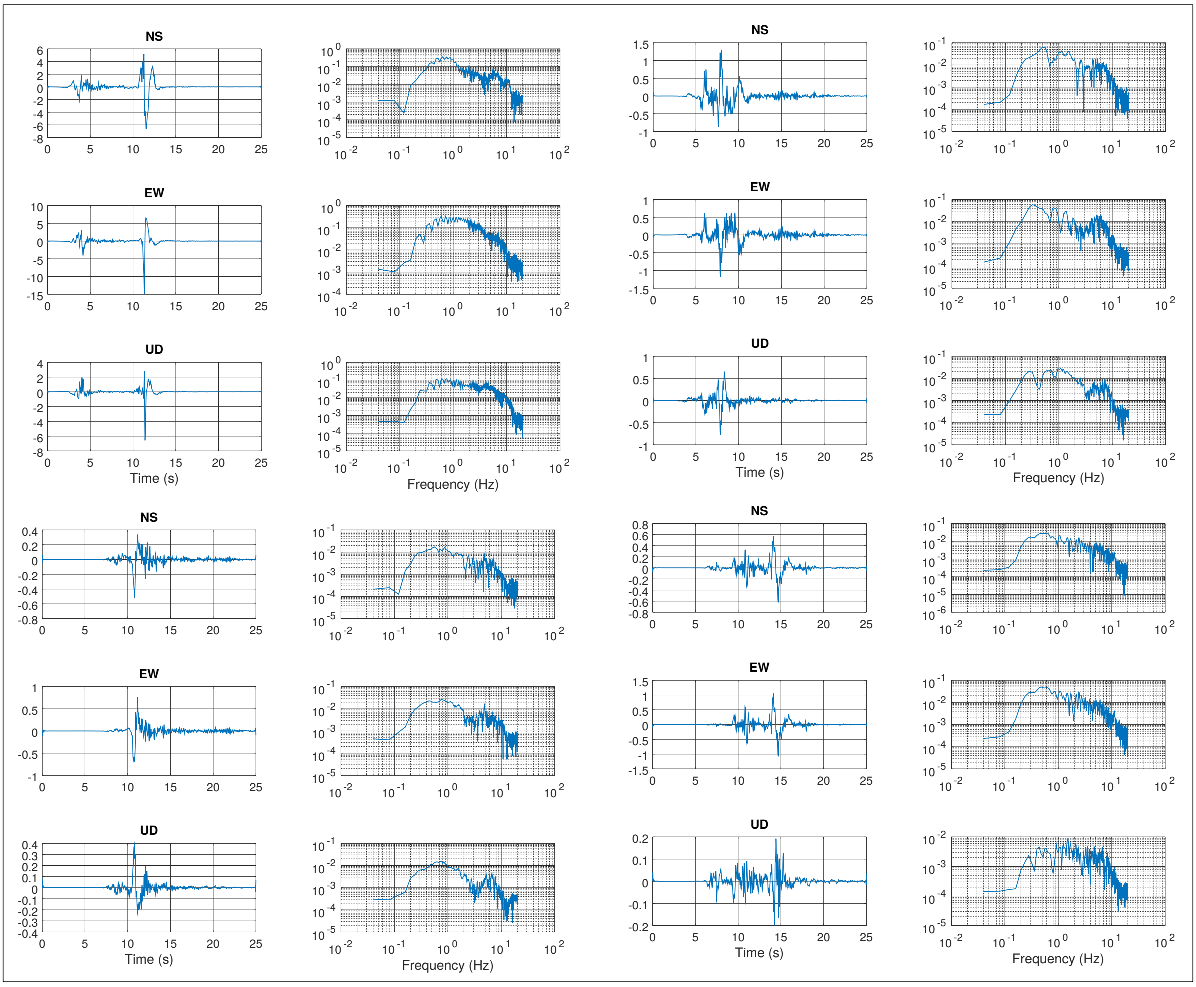


Figure 2.6: Acceleration synthetic seismograms (short-period component only) in the frequency range 0-10 Hz based on the rupture model shown in Figure 2.5. The four sets of timeseries refere to station STCH (top left), EBON (top right), ISBO (bottom left) and HECT (bottom right). See also Figure 1 of [15].

Figure 2.5 shows a current known bug in *BBToolbox*: there is a wrap-around effect in some timeseries. This depends on a convolution subroutine where zero-padding hasn't been performed properly. This bug can be fixed easily.

License

References

[1] Stephen Hartzell et al. "Calculation of broadband time histories of ground motion: Comparison of methods and validation using strong-ground motion from the 1994 Northridge earthquake". In: *Bull. Seismol. Soc. Am.* 89.6 (Jan. 1999), pp. 1484–1504. ISSN: 00371106.

[2] Pengcheng Liu, Ralph J. Archuleta, and Stephen H. Hartzell. "Prediction of Broadband Ground-Motion Time Histories: Hybrid Low/High- Frequency Method with Correlated Random Source Parameters". In: *Bull. Seismol. Soc. Am.* 96.6 (Dec. 2006), pp. 2118–2130. ISSN: 0037-1106. DOI: 10.1785/0120060036.

[3] Robert W. Graves and Arben Pitarka. "Broadband Ground-Motion Simulation Using a Hybrid Approach". In: *Bull. Seismol. Soc. Am.* 100.5A (Oct. 2010), pp. 2095–2123. ISSN: 0037-1106. DOI: 10.1785/0120100057.

[4] J. G. F. Crempien and Ralph J. Archuleta. "UCSB Method for Simulation of Broadband Ground Motion from Kinematic Earthquake Sources". In: *Seismol. Res. Lett.* 86.1 (Jan. 2015), pp. 61–67. ISSN: 0895-0695. DOI: 10.1785/0220140103.

[5] P Spudich and Ln Frazer. "Use of ray theory to calculate high-frequency radiation from earthquake sources having spatially variable rupture velocity and stress drop". In: *Bull. Seismol. Soc. Am.* 74.6 (Jan. 1984), pp. 2061–2082. ISSN: 0037-1106.

[6] Haruo Sato, Michael C. Fehler, and Takuto Maeda. *Seismic wave propagation and scattering in the heterogeneous earth: Second edition.* Vol. 9783642230. 2012, pp. 1–494. ISBN: 9783642230295. DOI: 10.1007/978-3-642-23029-5.

[7] František Gallovič. "Modeling Velocity Recordings of the M w 6.0 South Napa, California, Earthquake: Unilateral Event with Weak High-Frequency Directivity". In: *Seismol. Res. Lett.* 87.1 (Jan. 2016), pp. 2–14. ISSN: 0895-0695. DOI: 10.1785/0220150042.

[8] P. Martin Mai, Walter Imperatori, and Kim B. Olsen. "Hybrid Broadband Ground-Motion Simulations: Combining Long-Period Deterministic Synthetics with High-Frequency Multiple S-to-S Backscattering". In: *Bull. Seismol. Soc. Am.* 100.5A (Oct. 2010), pp. 2124–2142. ISSN: 0037-1106. DOI: 10.1785/0120080194.

[9] Meeke C. van Ede et al. "Hybrid Broadband Seismograms for Seismic Shaking Scenarios: An Application to the Po Plain Sedimentary Basin (Northern Italy)". In: *Pure Appl. Geophys.* 177.5 (May 2020), pp. 2181–2198. ISSN: 0033-4553. DOI: 10.1007/s00024-019-02322-0.

[10] P. Martin Mai and G.C. Beroza. “A hybrid method for calculating near-source, broadband seismograms: application to strong motion prediction”. In: *Phys. Earth Planet. Inter.* 137.1-4 (May 2003), pp. 183–199. ISSN: 00319201. DOI: 10.1016/S0031-9201(03)00014-1.

[11] Aspasia Zerva. *Spatial Variation of Seismic Ground Motions.* Vol. 1. CRC Press, Apr. 2016, p. 474. ISBN: 9780429121784. DOI: 10.1201/9781420009910.

[12] Zheqiang Shi and Steven M. Day. “Rupture dynamics and ground motion from 3-D rough-fault simulations”. In: *J. Geophys. Res. Solid Earth* 118.3 (Mar. 2013), pp. 1122–1141. ISSN: 21699356. DOI: 10.1002/jgrb.50094.

[13] P. Martin Mai et al. “Accounting for Fault Roughness in Pseudo-Dynamic Ground-Motion Simulations”. In: *Pure Appl. Geophys.* 174.9 (2017), pp. 3419–3450. ISSN: 14209136. DOI: 10.1007/s00024-017-1536-8.

[14] Sigurjón Jónsson et al. “Fault slip distribution of the 1999 Mw 7.1 Hector Mine, California, earthquake, estimated from satellite radar and GPS measurements”. In: *Bull. Seismol. Soc. Am.* 92.4 (2002), pp. 1377–1389. ISSN: 00371106. DOI: 10.1785/0120000922.

[15] Duncan Carr Agnew et al. “Coseismic displacements from the Hector Mine, California, earthquake: Results from survey-mode global positioning system measurements”. In: *Bull. Seismol. Soc. Am.* 92.4 (2002), pp. 1355–1364. ISSN: 00371106. DOI: 10.1785/0120000928.

Index