Broadband User Guide v16.5.0

From SCECpedia

Version 16.5.0

If you find errors in this document or sections to be unclear, please contact software @ scec.org. If possible, please send a description of how to reproduce the problem. This will help SCEC developers more quickly identify and resolve the problem you have identified.

Contents

Installing the Broadband Platform

Installing the Broadband Platform involves obtaining a copy of the code and building the required executables. You can either download the platform from the Broadband Platform website (see below) or check the code out of SCEC's Subversion repository. Most users should download the platform.

System Dependencies

The current version of the Broadband Platform is designed to run on standard 64-bit Linux machines and on Mac OS X. On Linux, testing has been performed on SCEC's development servers running Fedora Core 22 (kernel version 4.0.6-300.fc22.x86_64). On Mac OS X, testing was done on versions 10.9 (Mavericks) and 10.10 (Yosemite). In this guide we outline how to install the platform into your own account on a Linux or Mac computer using the simplest approach.

Software Dependencies

The Broadband Platform has certain software dependencies.

Required:

- Python v2.7.9+ with
  - Matplotlib 1.4.3
  - Numpy 1.9.2

    - Scipy 0.14.1
    - Pyproj 1.9.2

- GNU compilers (gcc, gfortran) v4.5.1

The following instructions describe how to download and install the Broadband Platform Release 16.5.0 in a Linux or Mac computer.

Mac OS X Installation

Mac OS X users should refer to our BBP on OS X Guide 16 5 for installing the above dependencies to their system. After completing the steps on that guide, please return to this page and continue reading these installation instructions to finalize the Broadband Platform setup.

Broadband Platform Easy Installation

This section provides a brief overview of how the Broadband Platform can be installed on your local Linux or Mac OS X computer. It covers the Easy Installation method, where users download and run a shell script that downloads the other components based on user responses. Users are welcome to set up the Broadband Platform manually on their systems using the old installation instructions described in the Broadband_Platform_Manual_Installation_16_5_0 page.

Starting with this version of the Broadband Platform, we provide an easy installation script that can guide users through downloading and installing the Broadband Platform on a Linux or Mac OS X computer that meets the system requirements for the Broadband Platform. For this method, users need to download the from the following location:

- Download the BBP 16.5.0 Easy Install Script (a bash script) (http://hypocenter.usc.edu/research/bbp/versions/16.5.0/easy_install_bbp_16.5.0.sh).

Once that script is downloaded, users should:

```
$ chmod +x easy_install_bbp_16.5.0.sh
$ ./easy_install_bbp_16.5.0.sh <directory>
```

The script will automatically create a "bbp" subdirectory, where it will install the Broadband Platform. To install the Broadband Platform in the current directory, users can simply type:

```
$ ./easy_install_bbp_16.5.0.sh .
```

```
====== Welcome to Broadband Platform 16.5.0 installation script =====

 Using destination directory: /home/sarah

=> Main Broadband Platform Source Distribution
==> Downloading...
==> Compiling...
==> Installed!
```

The easy installation script will automatically download and compile the main source distribution. Following that, it will ask users to select which regions should be installed along with the Platform. Please note that saying "Yes" to a region, will automatically install any validation events available for that region. At a minimum, and in order to run Unit and Acceptance tests, users should say "Yes" to the "LA Basin" region.

```
Please select what velocity models (regions) you would like to install:

==> Would you like to install the Northern California region?
1) Yes
2) No
#?2

==> Would you like to install the LA Basin region?
1) Yes
2) No
#? 1

...
...
...

=> Installing Broadband Platform Velocity Model Packages
==> LA Basin
==> Completed!
=> Installing Broadband Platform Validation Packages
==> Northridge
==> GMPEs
==> Completed!

=> All Done!
```

At the end of the installation, the easy install script will print instructions for the user to add several lines to their .bash_profile file:

```
Please add the following lines to your .bash_profile:

export BBP_DIR=/home/sarah/bbp/16.5.0/bbp
export BBP_GF_DIR=/home/sarah/bbp/bbp_gf
export BBP_VAL_DIR=/home/sarah/bbp/bbp_val
export PYTHONPATH=/home/sarah/bbp/16.5.0/bbp/comps
export BBP_DATA_DIR=/home/sarah/bbp/bbp_data
export PATH=/home/sarah/bbp/16.5.0/bbp/comps:/home/sarah/bbp/16.5.0/bbp/utils/batch:$PATH
```

After installing the Broadband Platform on their systems, users should confirm the code is built correctly by running Unit tests and then Acceptance tests before starting to use the code for research purposes. For detailed information, please refer to the **Broadband User Guide v16.5.0**.

Broadband Platform Source Distribution Directory Structure

The platform consists of several top-level directories:

- 16.5.0 - Contains the source code, executables, scripts, and tests
- bbp_gf - Contrains the Green's Functions for the several regions supported by the Broadband Platform
- bbp_val - Contains the Validation Events that can be used with the Broadband Platform
- bbp_data - Contains the inputs, outputs, logs, and other temporary files generated by Broadband

simulations. Please note that indata, logs, outdata, run, tmpdata, and xml sub-directories will be created when the Broadband Platform is first run, so they do not need to be created by the user.

The 16.5.0 directory has the bbp and doc sub-directories, containing the software distribution and some documentation and examples respectively. The bbp sub-directory contains the following folders:

- comps - Broadband modules required to run work-flows.
- mod_data - Data files used by the various simulation modules.
- plot - Scripts and data required to generate plots.
- src - Broadband science modules source code and supporting files.
- tests- Set of unit and acceptance tests.
- utils - Includes the batch utilities used for cluster operation

And the $BBP_DATA_DIR directory includes the following sub-directories:

- indata - Input data for simulations
- logs- Log files for simulations
- outdata - Output data from simulations
- run - Directory where user input files can be placed
- tmpdata - Temporary files generated during Broadband simulations
- xml - XML files with simulation parameters

In general, you will be interacting with the run directory for input files, comps to run the platform, tests to test the platform, and outdata to examine data products.

Adding aliases

You may find it helpful to add aliases, so you can quickly and easily move to different broadband directories with a single command. We recommend creating aliases for the comps, run, and outdata directories.

If you are using the Bash shell, you can create aliases by adding the following lines to ~/.bash_profile:

```
alias comps='cd $BBP_DIR/comps'
alias run='cd $BBP_DATA_DIR/run'
alias outd='cd $BBP_DATA_DIR/outdata'
```

Log out and log back in. You'll notice that now you can type the alias command as a shortcut to change directories:

```
$ pwd
/home/sarah
$ comps
$ pwd
/home/sarah/bbp/16.5.0/bbp/comps
```

This can be a useful way to navigate around the broadband platform directories.

Running the Tests

The Broadband Platform contains two kinds of tests: unit and acceptance tests.

Unit tests are designed to test each module separately, using a set of input files, and compare the results against known outputs. They verify that each module has been built and is working correctly. Acceptance tests verify that the modules are working correctly together as a simulation method. They test the Platform end-to-end using different combinations with known inputs and compare the results. Each method (for both user-defined and validation events) are tested as checks against integration errors. Both unit and acceptance tests require the LABasin Greens' Functions package to be installed. The acceptance tests also require users to install the Northridge validation package.

Running Unit Tests

In order to run the unit tests, users should go to the tests directory and type:

```
$ cd /home/sarah/bbp/16.5.0/bbp/tests
```

We recommend you run the Unit and Acceptance tests in the background using the following command. On a Linux of Mac computer type:

```
nohup ./UnitTests.py > tests.log 2>&1 &
```

This will output stdout to a file called tests.log. The UnitTests also produce a log file in the BBP_DATA_DIR with additional details about each method tested. Examine the tests.log file to determine whether the tests complete successfully. When they do you should see a message like the following:

You can run the Unit tests directly, with the following command. However you must keep the terminal window open until the tests complete.

```
$ ./UnitTests.py
```

The tests should begin and will take between 30-60 minutes to run, depending on your computer speed. At the end of each test, a "ok" should be printed if the test was successful. At the end, the program will print the number of tests that passed and the number of tests that failed. If a test has failed, first check that you have built the executables.You can rerun just the specific test that failed (test_<module>.py). If the test is still failing, also verify that you have the ref_data directory, since it contains the input and reference files. If you're looking for more information about the failure, you can consult the Unit Tests log file in $BBP_DATA_DIR/logs/unit_tests.log . If you can't determine the reason for the failure, contact support (mailto:software@scec.org).

Please note that one of the SDSU components, BBToolbox, does not pass its Unit Test when running on a Mac OS X system and the Platform will automatically skip this test. This is a known limitation of the current BBP release. You will see a message similar to:

```
*** Mac OS X detected: skipping SDSU BBToolbox unit test.
```

This reduces the available number of ground motion methods that can be run on the Mac. On Linux, there are six methods available (GP, SDSU, UCSB, EXSIM, SONG, CSM). On a Mac, there are five methods available (GP, UCSB, EXSIM, SONG, CSM).

Once the unit tests all pass, proceed to the acceptance tests. If there are any failure or errors while running the unit tests, consult the Troubleshooting section at the end of this user guide for know issues and their solutions.

Running Acceptance Tests

Make sure the unit tests pass before moving on to the acceptance tests. To run the acceptance tests, users should:

```
$ cd /home/sarah/bbp/16.5.0/bbp/tests
```

To run the acceptance tests in a terminal background process, on a Linux computer, the following command will run the tests as a background process, and redirect stderr and stdout to a "tests.log" file.

```
$ nohup ./AcceptTests.py > tests.log 2>&1 &
```

You can run the Acceptance tests directly in a terminal window, but you must keep the terminal window open until all tests complete.

```
$ ./AcceptTests.py
```

There will be one test for each method in the platform, for both the validation mode (historical events) and the scenario modes (hypotethical earthquakes). These tests will take somewhere between 2 and 4 hours to run, depending on how fast the computer is.

When they're complete, the console will either print "OK" or how many tests failed. Acceptance test failures indicate that the modules are not integrated correctly. Like with the unit tests, verify that you have the ref_data directory. If a certain acceptance test fails, you can get more information by consulting the acceptance test logs in $BBP_DATA_DIR/logs/acceptance_test_logs/<test that failed>.log . If you can't determine the reason for the failure, contact support (mailto:software@scec.org).

Here is an example output from the tests.log file created by running this command line above.

-bash-4.1$ more tests.log

```
nohup: ignoring input
==> Number of tests to run: 14
test_user-CSM (__main__.BBP2G_Acceptance_Tests) ... ok
test_user-EXSIM (__main__.BBP2G_Acceptance_Tests) ... ok
test_user-GP (__main__.BBP2G_Acceptance_Tests) ... ok
test_user-SDSU (__main__.BBP2G_Acceptance_Tests) ... ok
test_user-SONG (__main__.BBP2G_Acceptance_Tests) ... ok
test_user-UCSB (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-CSM (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-EXSIM (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-GP (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-GP_seis (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-SDSU (__main__.BBP2G_Acceptance_Tests) ... ok
```

```
test_valid-northridge-SDSU_seis (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-SONG (__main__.BBP2G_Acceptance_Tests) ... ok
test_valid-northridge-UCSB (__main__.BBP2G_Acceptance_Tests) ... ok

----------------------------------------------------------------------
Ran 14 tests in 11999.620s

OK
```

Please note that since the SDSU method will produce errors on Mac OS X systems, three of the tests above will be skipped. This is expected on a Mac OS X computer.

Performing Simulations

The platform supports two kinds of simulations, validation events and user-defined events. Validation simulations are performed using a historical event, and are directly compared to observed seismograms using goodness-of-fit. User-defined events are run using a rupture description provided by the user which may not necessarily be a historical earthquake.

When you run a simulation, the platform assigns an ID to it. This ID can be used to track the simulation and locate the output data products.

To supply input files to the platform, put them in the run directory. This directory is located in the $BBP_DATA_DIR directory and it is created by the platform when it is first ran. Extensions are important - the platform recognizes station lists (.stl), SRF files (.srf), and simple source descriptions (.src). For example, when selecting a station file, the Broadband Platform will list all the .stl files in the run directory and will prompt the user to select one.

Methods

To perform a simulation, a user selects a method, including optional modules. Currently, the Broadband Platform supports the following methods:

- Graves & Pitarka (GP)
- SDSU
- UCSB
- Composite Source Model (CSM - Under development)
- EXSIM
- SONG

Modules

The Broadband Platform consists of a series of modules. There are two main types of modules, science modules and utility modules. Science modules are those for which the platform has multiple implementations, provided by different coding research groups. Utility modules only have 1 implementation and are used by all simulations. A schematic of the available modules and their flow relationships is shown below in the following sections.

Science Modules

All simulations must include a module that creates synthetic seismograms. In some methods, there are separate low-frequency and high-frequency modules, while in other methods, these two steps are done by a single module. Rupture generation and site response are optional science modules. Users may select the following different implementations of each of these modules:

- Rupture generation: GP, UCSB, SONG
- Low-frequency: GP, UCSB, CSM, EXSIM
- High-frequency: GP, UCSB, SDSU, CSM, EXSIM
- Site response: GP, UCSB, SDSU

Utility Modules

Two spectral response utility modules are automatically run after the seismogram synthesis is completed and the optional site response module. Additionally, users may select an optional goodness-of-fit (GoF) utility module to run at the conclusion of the simulation. The Broadband Platform currently supports both the GP and SDSU GoF modules and the users can select to run one, or both (or none) of these modules.

Validation Simulations

To run a validation simulation, go to the data/run directory and run run_bbp.py. The platform will ask you a series of questions. Answer 'y' to "Do you want to perform a validation run?"

```
$ cd $BBP_DATA_DIR/run
$ run_bbp.py
Welcome to the SCEC Broadband Platform version 16.5.0.
================================================================================

Please select the Broadband Platform mode of operation:
   * Validation - Simulates a historical event
   * Scenario   - Runs a user-defined hypothetical event

Do you want to perform a validation simulation (y/n)? y
================================================================================

Please select a validation event from the list below:

(1) Northridge
(2) Loma Prieta
?
...
```

No input files are required by the user. However, you may wish to customize the validation simulation by selecting an alternate source description (src file) or a reduced station list to speed up the computations. You can put your own source description andor station list into the run directory (the format is described in section 6.3) or you can tell the platform where each file is located by using an absolute path. Note that any stations which do not have observed seismograms will not be included in the automatically generated goodness-of-fit comparison.

In addition to the low-frequency modules which compute seismograms using 1D Green's Tensors, certain methods allow validation events to be also run using precomputed seismograms to supply the low-frequency.

User-defined Simulations

To run a user-defined simulation, two input files are required, a source description (src file) and a station list (stl file). For certain methods, the source description can either be in SRF format or a simple source description (the format is described in section 5.1). Other methods will always require a simple source description (src file). To run a user-defined simulation, run run_bbp.py.

```
$ cd $BBP_DATA_DIR/run
$ run_bbp.py
Welcome to the SCEC Broadband Platform version 16.5.0.
================================================================================

Please select the Broadband Platform mode of operation:
   * Validation - Simulates a historical event
   * Scenario   - Runs a user-defined hypothetical event

Do you want to perform a validation simulation (y/n)? n
================================================================================

The Broadband Platform provides the following velocity models, which also include
several method-specific and region-specific parameters.

Please select a velocity model (either number or name is ok):

(1) NOCAL
(2) Mojave
(3) LABasin
(4) CentralJapan
(5) Canada1k
(6) WesternJapan
(7) CEUS1k
?
...
```

You may then choose the method you would like to run:

```
The Broadband Platform includes several scientific methods that can be
used to calculate synthetic seismograms.

Choose a Method to use in this Broadband scenario simulation:
(1) GP (Graves & Pitarka)
(2) UCSB
(3) SDSU
(4) EXSIM
(5) CSM (Composite Source Model) - Beta Version
(6) Song
?
```

Logging

During the run, log files will be produced in logs/<simulation ID>/<simulation ID>.<module name>.log. If the platform fails, this is a good place to look to determine the error. Additionally, any fatal errors will be recorded in fatal_error.log.

Metadata capturing all the executable calls is located in tmpdata/<simulation ID>/metadata.txt for careful tracing of exactly what was called. Both the log files and metadata can be useful if troubleshooting an issue.

Data Products

The platform produces a variety of data products. All data products are located in outdata/<simulation ID>. The ouput folder has a index-<simulation ID>.html file with a listing of all the data products in the output folder. You can open the index file on your browser window and then click through all data products. Image files can be displayed by clicking on the link to the file in the index.html page. The .bbp and .list files will be displayed as text files in the browser. On Mac OS X, you can see these data products by opening the outdata folder in Finder and double clicking on the specific file. On most Linux systems, you can show images using display:

```
$ display <PNG file>
```

Make sure you have X11 forwarding enabled.

Station Map

To help visualize the stations in relationship to the fault, the platform produces a PNG file displaying station locations with red circles and the fault plane with a black line. You can find this file in outdata/<simulation ID>/station_map.png.

The Broadband Platform also produces a station_map.kml file that can be opened in Google Earth and will show the fault location and stations with their labels. This file is useful to identify the individual stations on the map.

Seismograms

When running the Broadband Platform, you have the option to generate plots of velocity and acceleration seismograms for each station. Plots of these files can be found in outdata/<simulation ID>/<simulation ID>.<station>_<velocity or acceleration>_seis.png.

The raw seismogram data is available in outdata/<simulation ID>/<simulation ID>.<station>.vel.bbp (velocity) and outdata/<simulation ID>.<station>.acc.bbp (acceleration). Its format is described in section 6.4.

Response Spectra

The Respect and RotD50 codes, run at the end of each simulation, calculates the response spectra for each station. The raw Respect and RotD50 data is located at

```
outdata/<simulation ID>/<simulation ID>.<station>.rsp
outdata/<simulation ID>/<simulation ID>.<station>.rd50
```

in the format described in section 6.5 and section 6.6.

Goodness-of-fit

If you run a goodness-of-fit module, several additional data products are produced. Two goodness-of-fit modules are available in Broadband.

- GP Goodness-of-fit:

The goodness-of-fit comparison is performed by comparing the response spectra of a set of calculated seismograms to seismograms from another simulation or observed seismograms. For each station involved in the comparison, a plot comparing the response spectra calculated by the Respect module can be found in outdata/<simulation ID>/<comparison label>_<simulation ID>_<station>_rsp.png. Similarly, a plot comparing the response spectra calculated by the RotD50 module can be found in outdata/<simulation ID>/<comparison label>_<simulation ID>/<station>rotd50.png. A plot showing the seismograms on top and bottom can be found at outdata/<comparison label>_<simulation ID>_<station>_overlay.png. The seismogram comparison plot also includes a comparison of the arias duration. Finally, two goodness-of-fit plots are generated, one using the data from Respect, and a second one with data from RotD50. These plots can be found at gof-<comparison label>-<simulation ID>_r0-<cutoff distance>.png and gof-<comparison label>-<simulation ID>_r0-<cutoff distance>-rotd50.png.

Note that at least 3 stations must be run for goodness-of-fit to be valid. If fewer than 3 stations are run, no goodness of fit calculation will be performed.

- SDSU Goodness-of-fit:

The SDSU goodness-of-fit module can compute several GoF measures. A detailed list of these GoF measure is listed here: SDSU MO-Gof. The goodness-of-fit comparison is performed by comparing the synthetic seismogram to the observed seismogram for each of the sites. For each GoF measure computed, a summary file labeled GOF_<measure>.list is generated with the GoF values for each of the sites. A weighted sum of GoF measures called site-fit is available in gof_summaray.txt file. If the simulation was run with three or more sites in the station list, GoF map plots will be available in the output directory as <simulation ID>_<Gof measure>_map.png. If PGA or PGV GoF values are computed a set of overlay plots showing the observed and synthetic seismograms on top and bottom will be available as outdata/gof_plots/<simulation ID>_<station>_match_<format>_overlay.png files.

Rupture files

When a user-defined event is simulated, the user has the option to run a rupture generator. This generator produces an SRF file, found in outdata/<simulation ID>/*.srf. This file can be put in the run directory and used in future runs. Additionally, the platform produces a plot of the cumulative slip on the fault surface, outdata/<simulation ID>/<SRF prefix>.png.

Platform Modes

The platform can be run in multiple modes. The default is interactive mode, in which the user is prompted to answer a series of questions. Once all the information has been gathered, the simulation run begins.

For a large number of runs, or if the user is repeating a specific run, this can be tedious. The platform provides two other ways to describe a run, with an option file or an XML description.

An option file provides responses to all the questions that the platform poses. The format is described in section 6.7, but it's basically a text file, 1 entry per line, with support for comments. It can be fed to the platform using the -o option.

The platform will also accept XML files containing a full description of a run. The schema for these files is given in section 6.8. These files are also produced by the platform after every simulation, and placed in xml/<simulation ID>.xml. So if you want to rerun a simulation, you can point the platform to the XML file from that simulation using the -x option. Note that a new simulation ID will be assigned to the run, so there is no risk of overwriting previous simulation results.

Available Options

To get a list of the current available options, run run_bbp.py with the -h flag.

```
$ ./run_bbp.py --help
Usage: run_bbp.py [options]

Options:
 -h, --help                                show this help message and exit
 -x XML_FILE, --xml-file=XML_FILE          Run using XML description of workflow
 -s SIM_ID, --sim-id=SIM_ID                Force a sim id
 -o OPTFILE, --option-file=OPTFILE         File containing responses to interactive platform prompts
 -v, --version                             Broadband platform version
 -g, --generate-only                       Generates the XML description but does not run the platform
 -l LOG_FILE, --log=LOG_FILE               Directs output to a file, use to run BBP in background
 -m, --no-xml                              Do not generate xml
 -r RESUME_MODULE, --resume=RESUME_MODULE  Resume workflow from a certain module
 -e END_MODULE, --end=END_MODULE           End workflow after a certain module
 --expert                                  Turns on expert mode
```

Examples

Below are some examples that you can try using the sample files in the examples directory. You will need to have the Northridge Validation Package installed to use this example, which also requires that the LABasin Green's Functions package be installed in the Broadband Platform. Make sure all the tests pass before you try this.

You should be in the run directory when you start these examples. If you set the aliases defined above, type:

$ run

Otherwise, type a path to the run directory. The Run directory provides a collection point, a staging area, when you are assembling the input files to be used in a simulation:

```
$ cd $BBP_DATA_DIR/run
```

Running a Northridge Full Validation Simulation

You don't need to move any files for this. Notice that we will be using the EXSIM method, which will generate results quickly.

```
$ run_bbp.py
Welcome to the SCEC Broadband Platform version 16.5.0
```

```
================================================================================

Please select the Broadband Platform mode of operation:
   * Validation - Simulates a historical event
   * Scenario   - Runs a user-defined hypothetical event

Do you want to perform a validation simulation (y/n)? y
================================================================================

Please select a validation event from the list below:

(1) Loma Prieta
(2) Northridge
? 2
================================================================================

The Broadband Platform includes several scientific methods that can be
used to calculate synthetic seismograms.

Choose a Method to use in this Broadband validation simulation:
(1) GP (Graves & Pitarka)
(2) UCSB
(3) SDSU
(4) EXSIM
(5) CSM (Composite Source Model) - Beta Version
(6) Song
? 4
================================================================================
SRC file: /home/sarah/bbp/bbp_val/Northridge/common/nr_v14_02_1.src
================================================================================
STL file: /home/sarah/bbp/bbp_val/Northridge/common/nr_stats-ver_9_2013.stl
Running ExSim
...
```

This simulation should take about 30 minutes (as opposed to about 12+ hours if using some of the other methods). Once it's complete, the platform will tell you:

```
You can find results in $BBP_DATA_DIR/outdata/<simulation ID>
```

In that directory you will find:

- HTML Directory Listing (index-<simulation ID>.html)
- Velocity seismograms (<simulation ID>.<station>.vel.bbp)
- Acceleration seismograms (<simulation ID>.<station>.acc.bbp)
- Plots of velocity seismograms (<simulation ID>.<station>_velocity_seis.png)
- Plots of acceleration seismograms (<simulation ID>.<station>_acceleration_seis.png)
- Response spectra Respect files (<simulation ID>.<station>.rsp)
- Response spectra RotD50 files (<simulation ID>.<station>.rd50)
- Plots comparing simulated and observed seismograms (Northridge_<simulation ID>_<station>_overlay.png)
- Plots comparing simulated and observed Respect response spectra (Northridge_<simulation ID>_<station>_rsp.png)
- Plots comparing simulated and observed RotD50 response spectra (Northridge_<simulation ID>_<station>_rotd50.png)
- Overall Respect goodness-of-fit plots (gof-Northridge-<simulation ID>_r0-120.png)
- Overall RotD50 goodness-of-fit plots (gof-Northridge-<simulation ID>_r0-120-rd50.png)

Sample Northridge Validation simulation, custom stations

Validation runs can take a long time. The time needed to generate each low-frequency seismogram will generaly increase with the magnitude of the event, taking more than 1 hour per station for earthquakes with Mw > 7.2. Sometimes you might want to run with a reduced station list so the simulation will run faster.

Copy the station list from $BBP_DIR/doc/examples/northridge_3_stations into the run directory, which should be $BBP_DATA_DIR/run. Take a look at the format of the station file:

```
$ more northridge_3_stations.stl
#Required:  lon, lat, station name, Vs30
#Optional: low freq corner, high freq corner
#lon            lat     station Vs30    LF corn HF corn
-118.6417       34.5640 cast    450     0.120   111.11
-118.4180       34.0628 lacn    278     0.140   111.11
-118.8811       34.2886 moor    405     0.160   111.11
```

Now, run the platform, using a different station list. To do this, you will need to use the Broadband Platform "expert" mode. See below:

```
$ run_bbp.py --expert
Welcome to the SCEC Broadband Platform version 16.5.0.
================================================================================

Please select the Broadband Platform mode of operation:
   * Validation - Simulates a historical event
   * Scenario   - Runs a user-defined hypothetical event

Do you want to perform a validation simulation (y/n)? y
================================================================================

Please select a validation event from the list below:

(1) Loma Prieta
(2) Northridge
? 2
================================================================================

The Broadband Platform includes several scientific methods that can be used to
calculate synthetic seismograms.

Choose a Method to use in this Broadband validation simulation:
(1) GP (Graves & Pitarka)
(2) GP Seis (using precomp seismograms)
(3) UCSB
(4) SDSU
(5) SDSU Seis (using precomp seismograms)
(6) EXSIM
(7) CSM (Composite Source Model) - Beta Version
(8) Song
? 1
================================================================================

When starting a simulation from a source description (SRC) file, the Broadband Platform
workflow should include a rupture generator. Answer 'yes' here unless providing
a complex Standard Rupture Format (SRF) file.

Do you want to run the rupture generator (y/n)? y
================================================================================

Each validation package includes a default source description (SRC) file for a historical event.
Would you like to provide a different file instead of the default file provided?
Answer 'no' here if you would like to use the standard source file for this event.

Do you want to provide a custom source file (y/n)? n
```

```
================================================================================
SRC file: /home/sarah/bbp/bbp_val/Northridge/common/nr_v14_02_1.src
================================================================================

Station Selection
=================
Would you like to:
   (1) generate seismograms for all stations in the validation package
       OR
   (2) provide a custom list with a subset of the stations
? 2
Do you want to
   (1) select a BBP station list in /home/sarah/bbb/bbp_data/run
       OR
   (2) enter the path of a BBP station list file
? 1
```

You will see a list of station list files that you have in your run directory. For this example, we only have the northridge_3_stations.stl file that we copied from the examples directory. You could have more files in the run directory if you have already tried other unit and acceptance tests. For this example, just select the northridge_3_stations.stl file.

```
Here are the BBP station list files in the run directory.
Please select one:

(1) northridge_3_stations.stl
? 1
================================================================================
STL file: /home/sarah/bbp/bbp_data/run/northridge_3_sta.stl
================================================================================

Site Response
=============
Running a site response module is an optional step while running a
Broadband Platform simulation. It requires a station list file containing
the Vs30 values for each station location.

Do you want to run the site response module (y/n)? n
================================================================================
Do you want to generate velocity seismograms' plots (y/n)? y
================================================================================
Do you want to generate acceleration seismograms' plots (y/n)? y
================================================================================

The Broadband Platform can generate comparison plots of the validation data against
GMPEs to show how GMPEs match the recorded data for a certain event.

Do you want to generate a GMPE comparison plot (y/n)? y
================================================================================

Please select a GMPE set to use in the comparison (number of name are ok):

(1) CENA GROUP 1
(2) NGA-West1
(3) NGA-West2
? 3
================================================================================

Goodness-of-Fit Plot
====================
Running a goodness-of-fit (GoF) module is an optional  step while running a
Broadband Platform simulation. It creates a comparison plot showing how
well the calculated seismograms fit recorded data.

Do you want to run a goodness-of-fit module (y/n)? y
================================================================================
Users can optionally select a Goodness of Fit module to plot a comparison of how well
the simulated seismograms match the recorded data in a historical event.

Choose a Goodness of Fit (GOF) Module:
(1) GP
(2) SDSU
(3) Both
? 1
================================================================================

Additional Metrics
==================
Calculating additional metrics is an optional step on the Broadband Platform. It creates
additional plots and data files that can be used to study the simulation.

Do you want to calculate additional metrics (y/n)? n
```

Again, when the run completes, in about 15 minutes, you can find results in the output directory. You'll notice far fewer files, as only 3 stations were run instead of 133. The goodness-of-fit plots won't look very good - more stations are really needed to get an accurate plot. In the $BBP_DIR/doc/examples/northridge_3_stations/results directory you will find pre-calculated results for a few of the methods (GP and UCSB). You should obtain results equivalent to those.

Sample User-defined Southern California simulation with source description

Next let's try running a user-defined event. Copy the files $BBP_DIR/examples/scenario_1_station/nr_one_stat.stl and $BBP_DIR/examples/scenario_1_station/user_eq.src to the $BBP_DATA_DIR/run directory. user_eq.src is a simple source description. Its format is outlined in section 6.1.

```
$ run_bbp.py
Welcome to the SCEC Broadband Platform version 16.5.0.
================================================================================

Please select the Broadband Platform mode of operation:
   * Validation - Simulates a historical event
   * Scenario   - Runs a user-defined hypothetical event

Do you want to perform a validation simulation (y/n)? n
================================================================================

The Broadband Platform provides the following velocity models, which also include several
method-specific and region-specific parameters.

Please select a velocity model (either number or name is ok):

(1) Mojave
(2) LABasin
(3) CentralJapan
(4) LOMAP
(5) WesternJapan
? 2
```

Since we are running a Southern California event, we select the LABasin velocity model, as it is the one that most closely matches the location of our rupture and stations.

```
================================================================================

The Broadband Platform includes several scientific methods that can be used
to calculate synthetic seismograms.

Choose a Method to use in this Broadband scenario simulation:
(1) GP (Graves & Pitarka)
(2) UCSB
(3) SDSU
(4) EXSIM
(5) CSM (Composite Source Model) - Beta Version
(6) Song
? 1

================================================================================

The source description (SRC) file contains a description of the hypothetical
(or scenario) earthquake, including information like location, geometry,
magnitude, and mechanism.

Do you want to
   (1) select a source description in /Users/fsilva/Work/svn/temp/bbp_sims/run
       OR
   (2) enter the path of a source description file
? 1
Here are the source description files in the run directory.
Please select one:

(1) user_eq.src
? 1
Do you want to
   (1) select a BBP station list in /home/sarah/bbp/bbp_data/run
       OR
   (2) enter the path of a BBP station list file
? 1

Here are the BBP station list files in the run directory.
Please select one:

(1) nr_one_stat.stl
(2) nr_three_stat.stl
? 1
Running Genslip
...
```

Since this run only includes one station, it will run in about 5 minutes. In the output directory you'll notice there are no goodness-of-fit or files, since this is not a validation simulation. However, there is also a map file (station_map.png and station_map.kml), showing the fault plane and the stations, and a plot of the rupture slip (user_eq.png). The SRF generated by the rupture generator is in user_eq.srf; this file could be used in future runs. The filenames of the rupture slip plot and SRF are taken from the rupture description filename.

File Formats

This section offers descriptions of various file formats in the broadband platform.

Simple source description

This is the format for the simple source description. It is not whitespace sensitive. The filename must end in .src for the platform to identify it as a source description.

```
MAGNITUDE = <moment magnitude>
FAULT_LENGTH = <length of fault>
DLEN = <DX, along length>
FAULT_WIDTH = <down-dip fault width>
DWID = <DY, along width>
DEPTH_TO_TOP = <depth of fault below surface>
STRIKE = <strike>
RAKE = <rake>
DIP = <dip>
LAT_TOP_CENTER = <latitude of top center point in fault plane>
LON_TOP_CENTER = <longitude of top center point in fault plane>
HYPO_ALONG_STK = <along strike location of hypocenter (0.0 = middle)>
HYPO_DOWN_DIP = <down dip location of hypocenter (0.0 = top)>
SEED = <random seed used in calculations>
CORNER_FREQ = <corner frequency to use>
```

SRF rupture description

Details of the Standard Rupture Format (SRF) are given in more detail in the accompanying document. The general fault surface is represented by a distribution of point sources, and it can support one or more planes. The platform produces SRFs as output from rupture generators, but a user can also supply an SRF file as input for user-defined simulations. SRF filenames must end with the suffix .srf for the platform to correctly identify them.

Station File List

The user may specify a list of stations, matching the following format:

```
#optional
%comment
# lines
<lon> <lat> <stat name> <Vs30> <LF corner> <HF corner> <stat info>
...
```

The station list filename needs to end in .stl for the platform to recognize it as a station list.

BBP seismograms

All intermediate and output seismograms are produced in a 4-column text format, whether velocity or acceleration.

```
# optional
% comment
# lines
<timestamp> <North/South> <East/West> <Up/Down>
...
```

Time is in seconds and motion is in cm/s (velocity) or cm/s/s (acceleration) unless otherwise specified in the header. Velocity filenames end in .vel.bbp; accelerations in .acc.bbp.

Respect Response Spectra

The Respect response spectra file is in a 4-column text format.

```
# optional
% comment
# lines
<period (sec)> <North/South> <East/West> <Up/Down>
...
```

The platform samples Respect response spectra at 112 periods, 0.01-20 sec and outputs pseudo spectral acceleration in units of g.

RotD50 Response Spectra

The RotD50 response spectra file is in a 4-column text format.

```
# optional
# comment
# lines
```

```
<period (sec)> <PSA North/South> <PSA East/West> <RotD50>
...
```

The platform samples RotD50 response spectra at 63 periods, 0.01-10 sec and outputs pseudo spectral acceleration in units of g.

Option File

Option files contain responses to the prompts in the platform. You can use them with the -o option to run_bbp.py. The responses have to match the prompts exactly. For example, if you craft an option file assuming a certain order of station lists (stl) or source descriptions (src) files in the run directory and there is a different set of files there it will potentially cause the Broadband Platform to pick a wrong file or possibly abort. Whenever a list of method, files, velocity models, or validation events is presented, the user can use either name or number to select that option. Picking the name avoids problems with the ordering. Option files are recommended to perform a series of runs with different inputs or different modules. If you have a run you perform regularly, it's better practice to run the platform with the -g option and the option file to produce an XML description, which are more complete descriptions of the workflow.

```
#optional comments
%using either symbol
n #comments
y % can go
1 # here
LABasin # velocity model
...
```

Use with:

```
$ ./run_bbp.py -o <option file>
```

XML workflow description schema

Each time the platform is invoked, an XML file is produced describing the workflow, obeying the following schema:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name= "BBP_Run_Specification" minOccurs= "1" maxOccurs= "1">
  <xs:attribute name="version" type = "xs:string" use = "required" />
  <xs:complexType>
    <xs:sequence>
      <xs:element name= "Validation_Run" minOccurs= "0" maxOccurs = "1">
        <xs:attribute name= "event" type= "xs:string" use= "required" />
        <xs:attribute name= "input_station_file" type = "xs:string" />
        <xs:attribute name= "subset" type = "xs:boolean" use = "required" />
      </xs:element>
      <xs:element name= "BBP_Modules">
        <xs:complexType>
          <xs:sequence>
            <xs:element name = "BBP_Module" maxOccurs = "unbounded" >
              <xs:complexType>
                <xs:sequence>
                  <xs:element name= "name" type="xs:string" />
                  <xs:element name= "staged_files" >
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="file" type="xs:string" maxOccurs= "unbounded"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name= "arguments" >
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="argument" type="xs:string" maxOccurs="unbounded">
                          <xs:attribute name= "type" type= "xs:string" user= "required" />
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

These files are difficult to construct by hand, but can be generated using the -g flag when running run_bbp.py.

Advanced Users

Alternatively, if you would like access to the latest version of the platform, to get frequent but less thoroughly tested improvements, you can check out the platform from SCEC's Subversion version control repository on source.usc.edu. Only advanced users should follow this approach. If you are unsure, you should download the code as outlined in Downloading the Platform and skip this section.

You may need to request access from the SCEC system administrator, John Yu (johnyu@usc.edu (mailto:johnyu@usc.edu)). Make sure you have subversion in your path before beginning.

Once you have access, you'll need to decide if you want a stable tagged version, or the latest version in the repository. For a stable version, open a terminal window on the system of your choice and type:

```
$ svn checkout https://source.usc.edu/svn/broadband/tags/<stable version> <stable version>
```

For the latest version, type:

```
$ svn checkout https://source.usc.edu/svn/broadband/trunk trunk
```

This will check out the project to your local machine. It contains the source code, tests, and some example files. You also need to obtain a copy of the Broadband Platform Green's Functions packages. They are too large to be stored in version control. You can get the .tar.gz files for the Green's Functions from the same download site you obtained this guide.

Periodically you should check for updates to the code. To do so, go to your trunk directory and type

```
$ svn update
```

This will pull down any code updates that have been made in the repository. After you update, make the code again (detailed in section 1.6) so that any code changes are captured in the executables.

User Support

If you run into bugs with the platform, you can open a trouble ticket at the Broadband Trac site (http://northridge.usc.edu/trac/broadband).

Check to see if there has already been a ticket opened for the bug. If you are unable to get the platform to run, you can get direct user support by emailing software@scec.org (mailto:software@scec.org).

Appendix A: Troubleshooting

If you experience trouble building the platform or successfully running test and simulations, try the following solutions.

Build Errors

The instruction for installing Broadband Platform are listed in Section 1: "Installing the Second-Generation Broadband Platform" of the User guide. If after following all steps listed in this section of the user guide the build fails, check if the failure is listed in this section and try the solution to fix the issue you are facing.

Unit/Acceptance Test Warnings and Errors

Unit and Acceptance tests are provided to verify the Broadband platform and it's supporting modules built by the user are functioning as designed. Under certain circumstances, some of these test might fail or print a warning message. While some of these failures might indicate serious problems that will have to be addressed before the platform can used, it is acceptable to ignore some of the warnings. This section lists some issues and their solutions.

test_bbtoolbox

The test_bbtoolbox unit test will fail on Mac OS X. This is due to the raytracing code in the SDSU method not working on Mac OS X. This is a known issue and we hope to have this issue resolved in a future version of the Broadband Platform.

test_gof

In certain environments, the test_gof unit test will display a series of warning messages regarding the availability of certain fonts used in the plots. The user would see a number of messages like:

```
/usr/lib64/python2.7/site-packages/matplotlib/font_manager.py:1242: UserWarning: findfont: Font family ['STIX
Falling back to Bitstream Vera Sans
/usr/lib64/python2.7/site-packages/matplotlib/font_manager.py:1242: UserWarning: findfont: Font family ['STIX
Falling back to Bitstream Vera Sans
```

These messages do not indicate a failure of the unit test. At the end of this test you will still see the "ok" status, indicating that the test was successful.

test_user-SDSU

test_valid-northridge-SDSU

test_valid-northridge-SDSU_seis

All acceptance tests for the SDSU method will fail on Mac OS X due to the raytracing code in the SDSU method. This is a known issue and we hope to have this resolved in an upcoming version of the Broadband Platform.

Runtime Warnings and Errors

Below is a list of known issues in the current Broadband 16.5.0 release that can sometimes cause a simulation to fail.

SDSU Method: BBToolbox exits with code 174

The current version of SDSU's BBToolbox code can sometimes fail with an exit code of 174. This will cause the Broadband simulation to stop immediately and a message similar to the one below will be recorded in the bbtoolbox log file (which is located in the simulation's log directory at $BBP_DATA_DIR/logs/<simulation_id>:

```
*** Welcome to the Broad-Band Toolbox (v1.6) ***

Initialising the code, please type input filename ...
rl=          131
rl=          131
Starting slug3d: by J. Vidale, 1988, UCSC
Starting punch: by J. Hole, 1993, UBC-Stanford
-- travel-times written to :time3d_P.out
-- selected velocity file  :vel3d_P.bin
WARNING: Computing only to max radius = 1001
Completed radius = 10
Completed radius = 20
Completed radius = 30
Completed radius = 40
Completed radius = 50
Completed radius = 60
Completed radius = 70
Completed radius = 80
Completed radius = 90
Completed radius = 100
Completed radius = 110
wavefront done
Starting slug3d: by J. Vidale, 1988, UCSC
Starting punch: by J. Hole, 1993, UBC-Stanford
-- travel-times written to :time3d_S.out
-- selected velocity file  :vel3d_S.bin
WARNING: Computing only to max radius = 1001
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image              PC                Routine            Line        Source
BBtoolbox.exe      000000000043A28B  Unknown               Unknown  Unknown
BBtoolbox.exe      0000000000403A8B  Unknown               Unknown  Unknown
BBtoolbox.exe      000000000040372C  Unknown               Unknown  Unknown
libc.so.6          000000337641ECDD  Unknown               Unknown  Unknown
BBtoolbox.exe      0000000000403629  Unknown               Unknown  Unknown
```

This message reflects an issue in the raytracer code used by BBToolbox that will cause the code to fail with certain geometries. The SDSU team is currently working on a fix to this issue and we expect a future version of the Platform to include the fix. Meanwhile, the user can try to change the problem geometry in order to avoid the error. This can be done by overriding the geometry that is calculated based on the stations and rupture (using a larger x/y/z grid will not affect the simulation results but can sometimes solve this raytracer issue). To see what is the current problem geometry, users can look for the following line in the .bbpar file (located in the $BBP_DATA_DIR/indata/<simulation_id> directory:

```
/* GRID DEFINITION [X-Y-Z] FOR RAYTRACING: "FAR-SIDE" (IN KM) */
130.0 140.0 125.0
```

To specify a different geometry to be used by BBToolbox's raytracer, users can include the following keys in the SRC file:

```
GRID_X = xxx
GRID_Y = yyy
GRID_Z = zzz
```

Where xxx, yyy, and zzz are the new dimensions for the raytracer to use (specified in km).

SDSU Method: BBToolbox exits with code 139

The current version of SDSU's BBToolbox code can sometimes fail with an exit code of 139. We are working with the modelers to resolve this issue, but it appears related to memory management issues in the code. We have found that it is sometimes useful to increase the stack size (if one is set). This should be done before the simulation starts, with the command below (for bash):

```
$ ulimit -s unlimited
```

SDSU Method: BBToolbox fails unit test with *** buffer overflow detected *** error

The current version of SDSU's BBToolbox code has an internal issue that in some systems trigger built-in compiler checks and cause the code to abort. This is known to happen on more recent Ubuntu/XUbuntu installations, but can also happen in other systems where these checks are enabled by default. When running the BBToolbox unit test users will see a message like:

```
*** buffer overflow detected ***: /home/sarah/bbp/16.5.0/bbp/src/sdsu/bin/BBtoolbox.exe terminated
======= Backtrace: =========
/lib/x86_64-linux-gnu/libc.so.6(+0x7338f)[0x2afe8fe5138f]
/lib/x86_64-linux-gnu/libc.so.6(__fortify_fail+0x5c)[0x2afe8fee8c9c]
/lib/x86_64-linux-gnu/libc.so.6(+0x109b60)[0x2afe8fee7b60]
/home/sarah/bbp/16.5.0/bbp/src/sdsu/bin/BBtoolbox.exe[0x44351b]
/home/sarah/bbp/16.5.0/bbp/src/sdsu/bin/BBtoolbox.exe[0x4020e4]
/home/sarah/bbp/16.5.0/bbp/src/sdsu/bin/BBtoolbox.exe[0x4036bd]
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf5)[0x2afe8fdffec5]
/home/sarah/bbp/16.5.0/bbp/src/sdsu/bin/BBtoolbox.exe[0x401d79]
```

Until this problem is fixed, one solution is disabling the built-in check. That can be done by editing the makefile in the /home/sarah/bbp/16.5.0/bbp/src/sdsu/bin directory so that the first line looks like:

```
UFLAGS = -w -Wall -U_FORTIFY_SOURCE
```

and the line under the BBtoolbox.exe line looks like:

```
${FC} ${MODULES} ${CODES} -O3 -U_FORTIFY_SOURCE ray3DJHfor.o -o BBtoolbox.exe
```

Then, type:

```
$ make clean
$ make
```

UCSB Method: Site Response module

In the current version of the Broadband Platform, the UCSB site response module is producing unusually high amplitudes. This is likely due to a unit mismatch between the new versions of the source and wave propagation codes, and the old version of the site response module. Therefore, the UCSB method site response module should not be used in this release. We are currently working on a fix and will have a solution implemented by the next release of the Platform.

Related Entries

- CME Project
- SEISM Project
- Broadband Platform
- SCECpedia wiki Home Page (http://scec.usc.edu/scecpedia/Main_Page)

Acknowledging

Please support the Broadband Platform project by acknowledging the use of this software. Acknowledgements and citations help us obtain additional resources for continued development of the platform. If you use the Broadband Platform software for work resulting in an academic publication, we would appreciate it if one, or more, of the following paper is cited.

References for specific computational methods included in the Broadband Platform (v15.3.0 and later, including v16.5.0) and for the validation procedures developed by the Broadband Platform include:

1. Anderson, J. G (2015) The Composite Source Model for Broadband Simulations of Strong Ground Motions Seismological Research Letters, January/February 2015, v. 86, p. 68-74, First published on December 17, 2014, doi:10.1785/0220140098
2. Atkinson, G. M., and Assatourians, K. (2015) Implementation and Validation of EXSIM (A Stochastic Finite−Fault Ground−Motion Simulation Algorithm) on the SCEC Broadband Platform Seismological Research Letters, January/February 2015, v. 86, p. 48-60, First published on December 17, 2014, doi:10.1785/0220140097
3. Crempien, J. G. F., and Archuleta, R. J. (2015) UCSB Method for Simulation of Broadband Ground Motion from Kinematic Earthquake Sources Seismological Research Letters, January/February 2015, v. 86, p. 61-67, First published on December 17, 2014, doi:10.1785/0220140103
4. Dreger, D. S., Beroza, G.C., Day, S. M., Goulet, C. A., Jordan, T. H., Spudich, P. A., and Stewart, J. P. (2015). Validation of the SCEC Broadband Platform V14.3 Simulation Methods Using Pseudospectral Acceleration Data, Seismol. Res. Lett., 86, no. 1, doi:10.1785/0220140118.
5. Dreger, D. S., and Jordan, T. H. (2015) Introduction to the Focus Section on Validation of the SCEC Broadband Platform V14.3 Simulation Methods Seismological Research Letters, January/February 2015, v. 86, p. 15-16, doi:10.1785/0220140233
6. Goulet, C.A., Abrahamson, N.A., Somerville, P.G. and K, E. Wooddell (2015) The SCEC Broadband Platform Validation Exercise: Methodology for Code Validation in the Context of Seismic-Hazard Analyses, Seismol. Res. Lett., 86, no. 1, doi: 10.1785/0220140104
7. Graves, R., and Pitarka, A. (2015) Refinements to the Graves and Pitarka (2010) Broadband Ground−Motion Simulation Method Seismological Research Letters, January/February 2015, v. 86, p. 75-80, First published on December 17, 2014, doi:10.1785/0220140101
8. Olsen, K. B., and Takedatsu, R. (2015) The SDSU Broadband Ground−Motion Generation Module BBtoolbox Version 1.5 Seismological Research Letters, January/February 2015, v. 86, p. 81-88, First published on December 17, 2014, doi:10.1785/0220140102

The primary reference for the Broadband Platform software system (v15.3.0 and later, including v16.50) is:

1. Maechling, P. J., F. Silva, S. Callaghan, and T. H. Jordan (2015). SCEC Broadband Platform: System Architecture and Software Implementation, Seismol. Res. Lett., 86, no. 1, doi: 10.1785/0220140125.

Retrieved from "https://scec.usc.edu/scecwiki/index.php?title=Broadband_User_Guide_v16.5.0&oldid=16540"

- This page was last modified on 25 May 2016, at 16:40.