

Validation source perturbation (Source uncertainties) V2

To maintain compatibility with current workflows while adding additional functionality to the workflow, we intend to change the way perturbed source parameters are passed to the components of the slurm ground motion workflow.

- [Proposed workflow change](#)
- [Script usage](#)
 - [gcmt_to_realisation:](#)
 - [generate_realisations_from_gcmt:](#)
 - [realisation_to_srf:](#)
 - [generate_srf_from_realisations:](#)
- [Creating perturbations](#)
- [Currently available distributions](#)
- [Internal changes](#)
 - [Parameters for srf generation](#)
- [Parameters for each component](#)

Proposed workflow change

It has been proposed that srf generation will be split into two steps: perturbation and srf generation.

This will split the current two workflow steps (gcmt2srf, nhm2srf) to generate srf's into three (gcmt_to_realisation, nhm_to_realisation, realisation_to_srf).

There will be an intermediary file containing all the parameters to generate the srf, allowing users to manually modify these files if they choose to do so.

This will be able to be integrated into the automated workflow by making each step work on a single realisation, with a multi process wrapper to allow it to also be used as a stand alone step. Unless configured to not progress past file creation the automated workflow step would not allow time for the user to modify the intermediary file.

For each fault with perturbations a mean file is created in that fault's Data directory, this can be used to create the Velocity model (Eventually).

Script usage

Currently 4 scripts are available for the new source perturbation workflow. They are:

- `gcmt_to_realisation`: Takes in an event name, realisation count, gcmt file and srf type, creates realisation files
- `generate_realisations_from_gcmt`: Takes in a fault selection file, gcmt file and srf type, creates realisation files
- `realisation_to_srf`: Takes in a realisation file and creates the related srf, info, stoch and yaml files
- `generate_srf_from_realisations`: Takes in a cybershake root directory and creates all srf, info, stoch and yaml files from available realisation files

Brief usage information is available below, more help is available by calling "`<script> --help`".

If Pre-processing is installed using pip these scripts will be available from the terminal.

`gcmt_to_realisation`:

Usage: `gcmt_to_realisation.py event_name realisation_count gcmt_file type [--version <version>] [--output_dir <directory path>] [--aggregate_file <file path>] [--source_parameter <name> <file path>] [--common_source_parameter <name> <value>] [--vel_mod_1d <file path>] [--vel_mod_1d_out <directory path>]`

Arguments:

- `event_name`: Name of the event to generate realisations for.
- `realisation_count`: Number of realisations to generate. Does not use the `_RELXX` naming scheme if this is 1.
- `gcmt_file`: The path to the geonet cmt solution file.
- `type`: The type of srf to be generated. Currently types 1 and 2 are supported.
- `version`: Name of the perturbation version to use. Should be the name of the perturbation file without the `.py` extension.
- `output_dir`: Path to the directory to write realisation files to. Defaults to the current directory.
- `aggregate_file`: A filepath to the location an aggregate file should be stored. There should not be a file already present at this location.
- `source_parameter`: Values to be passed to be added to each realisation, specified in a file. The first argument should be the name of the value, the second the filepath to the space separated file containing the values. The file should have two columns, the name of a source followed by the value for that source, separated by some number of spaces. Pandas will automatically determine the type of the value. If multiple source parameters are required this argument should be repeated.
- `common_source_parameter`: Values to be passed to be added to each realisation, specified on the command line. The first argument should be the name of the value, the second argument should be the value of the argument. Values are cast to int or float where possible, otherwise left as a string. If multiple source parameters are required this argument should be repeated.
- `vel_mod_1d`: Path to a 1d velocity model profile to pass to the perturbation function.
- `vel_mod_1d_out`: The directory perturbed 1d velocity models will be saved to.

`generate_realisations_from_gcmt`:

Usage: `generate_realisations_from_gcmt.py fault_selection_file gcmt_file type [--version <version>] [--cybershake_root <directory path>] [--n_processes <int>] [--aggregate_file <file path>] [--source_parameter <name> <filepath>] [--common_source_parameter <name> <value>] [--vel_mod_1d <file path>]`

Arguments:

- `fault_selection_file`: Path to the fault selection file for this simulation, uses the standard format. (<fault_name> <realisation_count>\r\n)
- `gmt_file`: Path to the gmt file to use for this simulation.
- `type`: The type of srf to be generated. Currently types 1 and 2 are supported.
- `version`: Name of the perturbation version to use. Defaults to unperturbed, which will not perform any perturbations.
- `cybershake_root`: Path to the simulation root directory. Defaults to the current directory
- `n_processes`: Number of realisations to create at once. Values larger than the total number of realisations are not useful. Defaults to 1.
- `aggregate_file`: Path to a file to store a copy of all saved values in, useful for manual verification of distributions.
- `source_parameter`: Values to be passed to be added to each realisation, specified in a file. The first argument should be the name of the value, the second the filepath to the space separated file containing the values. The file should have two columns, the name of a station followed by the value for that station, separated by some number of spaces. Pandas will automatically determine the type of the value. If multiple source parameters are required this argument should be repeated.
- `common_source_parameter`: Values to be passed to be added to each realisation, specified on the command line. The first argument should be the name of the value, the second argument should be the value of the argument. Values are cast to int or float where possible, otherwise left as a string. If multiple source parameters are required this argument should be repeated.
- `vel_mod_1d`: Path to a 1d velocity model profile to pass to the perturbation function.

realisation_to_srf:

Usage: `realisation_to_srf.py realisation_file`

Arguments

- `realisation_file`: Path to a realisation file. All generated files are placed adjacent to this file.

generate_srf_from_realisations:

Usage: `generate_srf_from_realisations.py [--n_processes <n>] [--cybershake_root <path>]`

Arguments:

- `n_processes`: number of srf realisations to create at once. Values larger than the total number of realisations are not useful. Defaults to 1.
- `cybershake_root`: path to the simulation root directory. Defaults to the current directory.

Creating perturbations

For a given set of perturbations the user will create a perturbation file, the template for which can be found in `Pre-processing/srf_generation/source_parameter_generation/uncertainties`, and must be placed in the `versions` folder in that directory.

The version of the perturbation file is the name of the file (without the .py extension).

There are a number of distributions and transformations available in the files `common`, `distribution` and `mag_scaling`.

Currently available distributions

Distribution name	Distribution variable	Notes
uniform	half_range	mean is the median (middle) value, and half_range half the difference between the highest and lowest value.
uniform_relative	scale_factor	A normal distribution where the highest and lowest possible value are $\text{mean} * (1 + \text{scale_factor})$ and $\text{mean} * (1 - \text{scale_factor})$ respectively. Scale factor to be given as a decimal.
weibull	k, scale_factor	Does not take a mean value. Equivalent to $\text{scale_factor} * (-\ln((U)))^{1/a}$ where U is the uniform distribution over [0, 1).
truncated_weibull	truncation_threshold, k, scale_factor	Same as above, except if the value is greater than the truncation threshold then it is resampled. With default values for dhypo calculation this will occur in 0.005% of cases
truncated_normal	std_dev, std_dev_limit	Similar to normal distribution, except all values fall within [std_dev_limit] standard deviations of the mean. std_dev_limit has a default value of 2.
truncated_log_normal	std_dev, std_dev_limit	Similar to log normal distribution, except all values of the underlying normal distribution fall within [std_dev_limit] standard deviations of the mean. std_dev_limit has a default value of 2. Mean must be given as a string of the form "log(x)" where x is the unmodified value of the mean.

Internal changes

In order to simplify the implementation of this, we have chosen to have every parameter added to the slurm script and passed as an argument to the simulation script, which will then discard any parameters it does not recognize.

Parameters for srf generation

These parameters are available for srf generation. The parameter and its value must be placed in a dictionary mapped to the "params" key of the results dictionary to be used.

This table only shows values that are currently available, as other values become available they will be added

Key:

- R: Required to run
- O: Optional, the default value is given in brackets

Parameter	Type 1	Type 2	Type 3	Type 4
type	R			
name	R (Automatically set)			
latitude	R	R		Obtained from the Type3 subfaults
longitude	R	R		Obtained from the Type3 subfaults
depth	R	R		
magnitude	R			
mom	O (calculated from mag using mag2mom)			
strike	R	R	R	Obtained from the Type3 subfaults
rake	R	R	R	R
dip	R	R	R	R
vs	O (3.2)			
rho	O (2.44)			
risetime	O (0.5)			
target_area_km	O ¹			
target_slip_cm	O ¹			
stype	O ("cos") ²			
inittime	O (0.0)			
rvfac		O		O
rough		O		
slip_cov		O		
flen		R	R	Obtained from the Type3 subfaults
dlen		R		
fwid		R	R	Obtained from the Type3 subfaults
dwid		R		
dtop		R	R	R
shypo		R	R	R
dhypo		R	R	R
v_mod_1d_name	Automatically set (From input value or default)			
dbottom			R	R
tect_type				R
plane_count				R

1) target_area_km and target_slip_cm are used to calculate the area of the fault plane and slip of the fault. target_area_km takes precedence over target_slip_cm, but if neither are given the area and slip are calculated from the moment.

2) Available options are: brune, urs, ucsb, ucsb-T, ucsb2, ucsb-varT1, esg2006, cos, seki, delta.

Parameters for each component

The parameters for each component are as follows:

	hf	bb
flo		
fmin		
fmidbot		
lfvsref		
sim_bin		
t-sec		
sdrop		
rayset		
seed		
duration		
dt		
fmax		
kappa		
qfexp		
rvfac		
rvfac_shallow		
rvfac_deep		
czero		
calpha		
mom		
rupv		
velocity-model		
site-vm-dir		
vs-moho		
fa_sig1		
fa_sig2		
rv_sig1		
hf_path_dur		
dpath_pert		