File Formats Used On GM

This page has descriptions of the file formats that we use in various places.

SRF Format

- https://strike.scec.org/scecpedia/Standard_Rupture_Format
- SRF-Description-Graves_2.0.pdf
- SRF File Format Version 1 (output of genslip): srf_description_version_1.pdf
- Details on Source Modelling : Source Modelling for GMSim

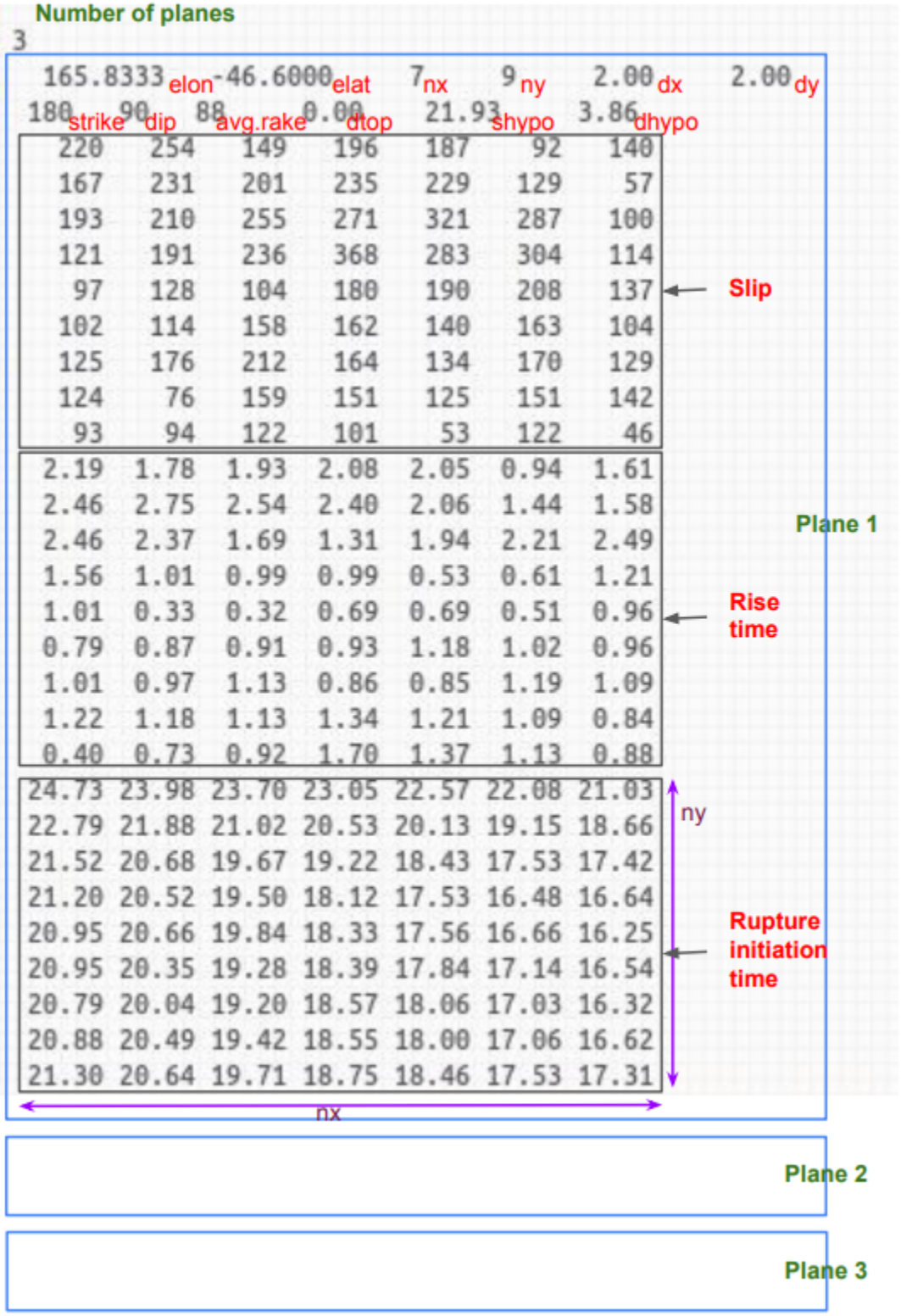
SRF info format

The **.info** files that accompany **.srf** files are in **HDF5** format

Example (CBalleny.info)

Group	
Path	/
File Path	CBalleny.info
Raw	▶ Inspect
Attributes	
Centre	[[165.833328,-46.599995],[165.816666,-46.708332],[165.800003,-46.941662]]
Corners	[[[165.833328,-46.533519689819904],[165.833328,-46.6664703101801],[165.833328,-46.6664703101801],[165.833328,-46.533519689819904]], [[165.83292856835644,-46.66668097572494],[165.8003783151683,-46.749980716873175],[165.8003783151683,-46.749980716873175], [165.83292856835644,-46.66668097572494]],[[165.800003,-46.749871746980375],[165.800003,-47.13345225301961],[165.800003,-47.13345225301961], [165.800003,-46.749871746980375]]]
Dbottom	[18,18,18]
Dhyp	[3.8575,-999.9,-999.9]
Dhyp0	3.85753669909383
Dip	[90,90,90]
Dt	0.005
Dtop	[0,0,0]
Hdepth	3.85
Hlat	-47.028778
Hlon	165.800003
Length	[14.8,9.6,42.7]
Mag	7.082216870560312
Ndip	[180,180,180]
Nstrike	[148,96,427]
Rake	90
Shyp	[21.9291,-999.9,-999.9]
Shyp0	[21.92912052127512]
Strike	[180,195,180]
Tect_type	"ACTIVE_SHALLOW"
Type	4
Vm	"lp_generic1d-gp01_v1.vmod"
Width	[18,18,18]

Stoch format



LF/HF/BB binary format

These files store timeseries data. All formats follow a style derived from the LF seis format produced by EMOD3D:

1. station size, common metadata
2. station list with station metadata
3. timeseries

Numbers are 4 bytes in length and may be little or big endian. You can see or use the **existing interfaces** that also take care of endianness at github: ucgmsim/qcore/qcore/timeseries.py::LFSeis, HFSeis, BBSeis.

File size can be derived knowing the format and the number of stations, and shape of time-series (all necessary values are at the beginning of the file).

The **second section is repeated** for each station.

The LF format contains unnecessarily repeated common metadata in the station list section (in italics below).

In BB, **stations are ordered** to match the station input file used in the LF. This means if there was an 'index of station in input file' in BB, it would run incrementally from 0. This is also the case for HF however it is based on the station file given which should therefore be the same (same order of stations) and this is currently a requirement for BB.

HF and BB have gaps between the first 2 sections to allow future additions to section 1 without breaking backwards compatibility.

LF	HF	BB
i4 number of stations **TOTAL 4 BYTES**	**i**4 number of stations **i**4 number of timesteps **i**4 seed used in simulation **i**4 site amplification used (bool) **i**4 path duration method • 0: GP2010 formulation • 1: WUS modification trial/error • 2: ENA modification trial/error • 11: WUS formulation of BT2014 over predicts for multiple rays • 12: ENA modification trial/error over predicts for multiple rays **i**4 number of ray methods known options for first-fourth type below: • 1: direct • 2: moho **i**4 first ray method used **i**4 second ray method used **i**4 third ray method used **i**4 fourth ray method used **i**4 nbu parameter **i**4 ift parameter **i**4 nlskip parameter **i**4 icflag parameter (bool) **i**4 individually run stations (bool) **i**4 site specific VMs used (bool) **f**4 duration of timeseries (s) **f**4 timestep of timeseries (s) **f**4 start time of timeseries (s) **f**4 stress drop average (bars) **f**4 kappa parameter **f**4 q frequency exponent **f**4 max sim frequency (Hz) **f**4 flo parameter (Hz) **f**4 fhi parameter (Hz) **f**4 rupture velocity factor (rupture : Vs) **f**4 rvfac shallow fault multiplier **f**4 rvfac deep fault multiplier **f**4 czero coefficient, -1: used binary default **f**4 calpha coefficient, -1: used binary default **f**4 seismic moment, -1: used rupture model **f**4 rupture velocity, -1: used rupture model **f**4 depth to moho, -1: used binary default (999.9) **f**4 vp_sig parameter sig parameters are likely uncertainties **f**4 vsh_sig parameter **f**4 rho_sig parameter **f**4 qs_sig parameter **f**4 fourier amplitude uncertainty (1) **f**4 fourier amplitude uncertainty (2) **f**4 rupture velocity uncertainty **s**64 stoch file used (basename) **s**64 velocity model used (basename) **TOTAL 288 BYTES**	**i**4 number of stations **i**4 number of timesteps **f**4 duration of timeseries (s) **f**4 timestep of timeseries (s) **f**4 start time of timeseries (s) **s**256 LF directory path used **s**256 LF VM directory path used **s**256 HF file path used possibly add vsite file path used here? **TOTAL 788 BYTES**

START OFFSET 4 BYTES	START OFFSET 512 BYTES	START OFFSET 1280 BYTES
i4 index of station in input file **i**4 x gridpoint of station **i**4 y gridpoint of station **i**4 z gridpoint of station ***i**4 simulation number of timesteps* ***f**4 simulation timestep (s)* ***f**4 simulation grid spacing (km)* ***f**4 simulation grid rotation (degrees)* **f**4 latitude of station (degrees) **f**4 longitude of station (degrees) **s**8 name of station **TOTAL 48 BYTES * NUM_STATIONS**	**f**4 longitude of station (degrees) **f**4 latitude of station (degrees) **s**8 name of station **f**4 epicentre distance to station (km) **f**4 vs30 at station (m/s) **TOTAL 24 BYTES * NUM_STATIONS**	**f**4 longitude of station (degrees) **f**4 latitude of station (degrees) **s**8 name of station **i**4 x gridpoint of station **i**4 y gridpoint of station **i**4 z gridpoint of station **f**4 epicentre distance to station (km) **f**4 HF vs30ref (m/s) **f**4 LF vs30ref (m/s) **f**4 BB vs30 (m/s) **TOTAL 44 BYETS * NUM_STATIONS**
START OFFSET 0 FROM ABOVE **f**4 velocity (cm/s) timeseries in array dimensions: **timestep**, **station**, **component** (x, y, z, ..., 9) **TOTAL 4 BYTES * PRODUCT_OF_DIMENTIONS**	**START OFFSET 0 FROM ABOVE** **f**4 acceleration (cm/s^2) timeseries in array dimensions: **station**, **timestep**, **component** (x, y, z) **TOTAL 4 BYTES * PRODUCT_OF_DIMENTIONS**	**START OFFSET 0 FROM ABOVE** **f**4 acceleration (g) timeseries in array dimensions: **station**, **timestep**, **component** (x, y, z) **TOTAL 4 BYTES * PRODUCT_OF_DIMENTIONS**

XYTS.e3d binary format

This file is produced by EMOD3D and contains a timeseries of ground motions on the XY plane. Unlike the LF seis files, this contains data at all grid points and may have a decimated resolution specified when running EMOD3D through the *e3d.par* file with the parameters *dxts* and *dyts*.

Numbers are 4 bytes in length and may be little or big endian. You can see or use the **existing interfaces** that also take care of endianness at github: ucgmsim/qcore/qcore/xyts.py::XYTSFile.

File size can be derived knowing the format and the shape of time-series (all necessary values are at the beginning of the file).

The **gridpoints are based on a model** which is an area with equidistant gridpoints in the X, Y, and Z directions. It is centred on a position (longitude, latitude) and may be rotated.

1. simulation metadata
    a. **INTEGERS**
        i. number of first x gridpoint
        ii. number of first y gridpoint
        iii. number of first z gridpoint
        iv. number of first timestep
        v. number of x gridpoints
        vi. number of y gridpoints
        vii. number of z gridpoins (always 1 by definition of X-Y file)
        viii. number of timesteps
    b. **FLOATS**
        i. x spacing between given gridpoints (km)
        ii. y spacing between given gridpoints (km)
        iii. original (pre-decimated) grid spacing between gridpoints used in simulation (km)
        iv. timestep in timeseries (s)
        v. model rotation of gridpoints (degrees)
        vi. model centre latitude (degrees)
        vii. model centre longitude (degrees)
2. timeseries
    - float array of velocities in the dimentions of **timesteps**, **components** (x, y, z), **y** grid positions, **x** grid positions.

Intensity Measure calculation

The IM calculation code will produce a number of text files (decided as of 25/05/2018). We will summarize them in the following.

Intensity measure files

There are two types of IM files: per station and aggregate. The per station one has the following format:

```
component, IM_1, IM_2, ...., IM_N
```

Note: The per station file does not have the station name, as it is the file name.

The aggregate one has all the stations on a single place:

```
station, component, IM_1, IM_2, ...., IM_N
```

Empirical IMs

As above the empirical intensity measures have a similar format:

```
station, component, IM_1, IM_1_sigma, IM_2, IM_2_sigma, ...., IM_N, IM_N_sigma
```

Notes: 1) the component for empirical IMs is always 'geom' 2) only total sigma is saved to the csv file

Rrup file

The file format for this is:

```
station, lat, lon, rrup, rjbs, rx
```

Note: we don't have rx calculations yet, so we may dump an invalid value just to conform with the format.

Metadata file

So far the requirements indicate that we need:

```
identifier, rupture, type, date, version
```

GSF File

The GSF file is used to define the geometry of a source in a source modelling problem. It is the first step in the SRF generation process after a realisation is read.

A GSF file contains three sections in order:

1. Errata/Header section,
2. Declaration of the number of points (N),
3. The geometry description: N lines representing each point in the geometry.

There are utilities to read GSF files in the `gsf` module within qcore.

Errata/Header

The header consists of a number of commented lines, each beginning with # . Here is an example:

```
# nstk= 179 ndip= 215
# flen=    17.0720 fwid=   21.2853
# LON  LAT  DEP(km)  SUB_DX  SUB_DY  LOC_STK  LOC_DIP  LOC_RAKE  SLIP(cm)  INIT_TIME  SEG_NO
```

This is the typical output of `fault_seg2gsf` . The first line has the number of points in the strike and dip directions, respectively. Then the length and width of the fault, and the last line is a description of each column in the points section.

The header is skipped by programs parsing GSF files and may contain any number of lines. The Python GSF generator, for example, only prints out the column description.

```
# LON  LAT  DEP(km)  SUB_DX  SUB_DY  LOC_STK  LOC_DIP  LOC_RAKE  SLIP(cm)  INIT_TIME  SEG_NO
```

Declaration of the Number of Points

Immediately following the header, there is one line containing the number of points in the GSF geometry definition.

Geometry Description

The geometry description has N lines, where N is the number of points declared in the previous section. Each line has 11 space separated values representing one point in the geometry.

Column	Description
LON	The longitude of the point.
LAT	The latitude of the point.
DEP	The depth of the point (in kilometres, with -10 meaning 10km below ground level).
SUB_DX	The subdivision length in the strike direction.
SUB_DY	The subdivision length in the dip direction.
LOC_STK	The fault segment strike.
LOC_DIP	The fault segment dip.
LOC_RAKE	The fault segment rake.
SLIP	The total slip at this point (cm), usually -1.
INIT_TIME	The initial rupture time of this point, usually -1.
SEG_NO	The number corresponding to the segment this point belongs to.: