<div align="center">

CMT3D_FLEXWIN **Readme**

</div>

CMT3D_FLEXWIN is re-written from my old CMT3D program to adapt to the output of Alessia's FLEXWIN package. The idea is that one runs the data and synthetics through FLEXWIN, pick time windows that have decent match between data and synthetics, and then use these windows in the moment-tensor and source mechanism inversions to achieve the best source solution.

1 Pre-processing

Once you have gathered the raw data from IRIS or SCEDC and computed synthetics from a preliminary cmt solution, you need to go through the pre-processing stage to present a proper version of data and synthetics to the cmt3d_flexwin package.

1. Processing data and synthetics

   ```
   ../scripts/process_data_and_syn.pl  \
    -d data,sac -s syn,  \
    -0 -m CMTSOLUTION -a station_file \
    -1 -b -26 \
    -2 -t 6/30 \
    -3 -x
   ```

   -d and -s give directory names and extension names of the raw data and synthetics. -0,-1-2,-3 represents different pre-processing stages which you can choose to run separately. During -0, information is gathered from CMTSOLUTION and station_file and written into sac headers. New directories data_PP and syn_PP are created to host these pre-processed data and syn files. During -1 stage, data and synthetics are padded before the beginning time to a value specified by -b, and proper cutting between data and synthetics are performed to ensure futher operations. -2 stage mainly deals with band-passing data and synthetics between the frequence bands given by -t, and moving the new processed files to directories data_T006_T030 and syn_T006_T030. -3 stage writes out the input file flexwin_T006_T030.input for FLEXWIN package after all the above steps have been executed. The script structure and many implementation details are inspired by a similar script written by Carl Tape.

2. Executing FLEXWIN

   ```
   ./flexwin < flexwin_T006_T030.input > flexwin_T006_T030.stdout
   ```

   and then cat the output files for individual pairs of data and synthetics into an output for cmt3d_flexwin package.

   ```
   ../scripts/prepare_meas_all.pl \
    flexwin_T006_T030 flexwin_T006_T030.output
   ```

2 Parameter file

For the structure of the program and subroutine calls, refer to `call-graph`. The name of the parameter file is hard-wired in the main program: `INVERSION.PAR`.

```
CMTSOLUTION
CMTSOLUTION_NEW
6                                -- npar : number of parameters inverted
0.0025 1.0  1.0e22               -- ddelta,ddepth,dmoment
flexwin.out
.true.                           -- weigh_data_files
2 2 1  0  1.15 0.55 0.78         -- weights of data comp,az,dist
.true.                           -- station_correction
.true.  .true. 0.0               -- zero_trace,double_couple,damping
.false.                          -- write_new_syn
```

The first line gives the old cmt solution file name. This is also most likely the cmt solution you have used to compute the synthetics for `FLEXWIN` picking. The second line gives name of the new cmt solution file you want to write after the inversion is done.

File `flexwin.out` holds the start and end time of the windows outputed from FLEXWIN package, in the following format:

```
nfiles
data-file-1
syn-file-1
 nwins-for-this-data-and-syn
 tstart(1) tend(1)
 ...
 tstart(nwin) tend(nwin)
data-file-2
syn-file-2
...
```

Only windows from this output file will be used to perform time integration in the process of assembling the inversion matrices.

All other parameters are pretty self-explanatory:

- Number of parameters to be inverted can only be 6(moment-only), 7(moment+depth), 9(moment+locations).

- The derivative synthetics should have been calculated based on the `ddelta`, `ddepth`, `dmoment` values. However, notice that strictly speaking, only the ones for moment are derivative synthetics, the ones for depth, longitude, and latitude are actual synthetics, from which you then need to subtract the old synthetics and divide by ddepth or ddelta.

- Weighting can be complicated if you choose to `weigh_data_files`. But the idea is that you utilize the `dist_km`, `dist_deg`, `azimuth`, and other file information from the sac headers to come up with a sensible weighting scheme for your particular scale/region, which you can accomplish by modifying subroutine `compute_data_weights()`.

- Station correction is used to align the data with the synthetics to accommondate 3D path effects that may skew your source inversion results.

- Different constraints on the moment-tensor are allowed, including zero-trace and double-couple. To improve the stability of the matrix inverse, damping is also allowed before Gaussian ellimination.

- Finally synthetics for the new cmt solution can be written, however, notice that these are obtained from 'linearized' approximation, not from an actual SEM simulation.

Several things one should always keep in mind, especially in comparision to the old version of this package:

- Filtering data and synthetics is no longer an option, because you are supposed to run your data and synthetics through pre-processing and filter them with the right frequency bands, and then go through the windowing code `flexwin` with filtering turned off.

- All auxilliary functionalities, such as padding the sac headers with information, outputing the time shifts and amplitude ratios, and writing the filtered data and synthetics have been eliminated from this program, since they are either done in `flexwin` or can be very easily accomplished by simple scripts (refer to section 2 for details). The gist of the program is Gaussian elimination to solve the inverse problem.

- If you want to invert either only surface waves or body waves, make sure you set the optimal parameters in flexwin.

- How do the weights work:

```
data_weights(nwint) =  cmp_weight(nwint) &
      * (dist/REF_DIST) ** dist_exp_weight(nwint) &
      / ( naz(k) ** az_exp_weight)
```

where component weights are applied linearly, distance and azimuth weights are applied exponentially. `naz(k)` is the number of traces available in a particular azimuth (+1). One can set `az_exp_weight` to be 0 to mute out the azimuthal effect, same thing with distance weights, and component weights.

- Envelope inversion is no longer included since they are rarely used in real inversions.

3   Testing

Two types of datasets are provided to accurately test this inversion package, one with station correction, and one without. Synthetics and derivative synthetics for corresponding CMTSOLUTION are always in `syn_T006_T030`. No pre-processing are needed for these dataset as they have gone through all the procedures, although to achieve inversion result that is accurate to several digits after the decimal point, several tweaks with the usual pre-processing procedures have to be done.

- we should not use -h in the synthetics processing to ensure that the synthetics data and the derivative synthetics go through the exact same set of processing procedures. we also need to take out the `rtrend` and `rmean` line in the filtering steps of `process_syn_new.pl` and `process_cal_data.pl`. Since the data and synthetics presented in `test_dataset` have gone through the script:

  ```
  process_data_and_syn.pl  -d data,sac -s syn,  \
          -0 -m CMTSOLUTION -a station_file   \
          -1 -b -26 -2 -t 6/30
  ```

  which generated `process_ds.csh` script for all the commands, we can just tweak individual lines in `process_ds.csh` to achieve our test dataset. Also note that the `-i none` option for data processing and `-h` option for synthetics processing have to be taken out from the script to make sure that synthetics and data are processed in the same way.

- The synthetic data for the 7 paramter case are generated by the following command:

  ```
  xadd_frechet_full s syn_file_list old_cmt true_cmt \
                    doment depth dx[dy]
  ```

  which is a fancier version of the original

  ```
  xadd_frechet_derivatives s syn_file_list \
                      cmt_file moment_scale
  ```

  Of course we can't really use the true synthetics for the true cmt solution in a 7-par case, since we know that the synthetics is a non-linear function of depth, and the inversion won't accurately recover depth. Linearized 'pseudo-synthetics' will ensure accurate results, which is the purpose of testing.

- It is impossible to design test dataset for 7 parameter inversion with time shift, since time shifts interfer with the effect of waveform change caused by depth variation, therefore, may not give exact solution, and sometimes may even produce the incorrect solution. This is part of the reason why I also wrote the grid search package `GRID3D_FLEXWIN`.

4 Auxilliary Scripts and Programs

A list of the scripts that are used before and after the package:

- `process_data_and_syn.pl`, which includes `process_data_new.pl`(global/regional) or `process_cal_data.pl` (socal), `process_syn_new.pl` (global/regional) or `process_trinet_syn_new.pl`(socal), `pad_zeros.pl` (include `pad_zero_to_syn`), `rotate.pl`, `saclst`. Of course `sac` has to be present in the system.

- `prepare_meas_all.pl`

- `xadd_frechet_derivatives` and `xadd_frechet_full`