INPUT PARAMETERS AND IMPLEMENTATION DETAILS

The `marchenko_primaries` program has the following parameters and options:

```
MARCHENKO_primaries - Iterative primary reflections retrieval

marchenko_primaries file_tinv= file_shot= [optional parameters]

Required parameters:

  file_shot= ............... Reflection response: R

Optional parameters:

INTEGRATION
  ishot=nshots/2 ........... shot number(s) to remove internal multiples
  file_tinv= ............... shot-record (from R) to remove internal multiples
  file_src= ................ optional source wavelet to convolve selected ishot(s)
COMPUTATION
  tap=0 .................... lateral taper R_ishot(1), file_shot(2), or both(3)
  ntap=0 ................... number of taper points at boundaries
  fmin=0 ................... minimum frequency in the Fourier transform
  fmax=70 .................. maximum frequency in the Fourier transform
  plane_wave=0 ............. model plane wave
  src_angle=0 .............. angle with horizontal of plane source array
  src_velo=1500 ............ velocity to use in src_angle definition
  t0=0.1 ................... time shift in plane-wave source wavelet for migration
MARCHENKO ITERATIONS
  niter=22 ................. number of iterations to initialize and restart
  niterec=2 ................ number of iterations in recursive part of the time-samples
  niterskip=50 ............. restart scheme each niterskip samples with niter iterations
  istart=20 ................ start sample of iterations for primaries
  iend=nt .................. end sample of iterations for primaries
MUTE-WINDOW
  shift=20 ................. number of points to account for wavelet (epsilon in papers)
  smooth=shift/2 ........... number of points to smooth mute with cosine window
REFLECTION RESPONSE CORRECTION
  tsq=0.0 .................. scale factor n for t^n for true amplitude recovery
  Q=0.0 ......,............. Q correction factor
  f0=0.0 ................... ... for Q correction factor
  scale=2 .................. scale factor of R for summation of Mi with M0
  pad=0 .................... amount of samples to pad the reflection series
OUTPUT DEFINITION
  file_rr= ................. output file with primary only shot record
  file_iter= ............... output file with -Mi(-t) for each iteration: writes
             ............... M0.su=M0 : initialisation of algorithm
             ............... RMi: iterative terms
             ............... u1min.su: u1min terms
  file_update= ............. output file with updates only => removed internal multiples
  T=0 ...................... :1 compute transmission-losses compensated primaries
  verbose=0 ................ silent option; >0 displays info


author  : Lele Zhang & Jan Thorbecke : 2020
```

Defining `file_iter` writes for each iteration the focusing update term $-M_i(-t) = RM_i(t)$ in Algorithm 1 before applying the mute window. This same option will also write

$k^-_{1,i}$ and $v^+_{1,i}$ after the update. By setting the `verbose=` option to 2 the energy of the focusing update term is printed out for each iteration and can be used to monitor the convergence of the scheme. When `file_update=` is given an output name the program writes the updates (= estimated internal multiples) to disk. The `scale` parameter can be useful when the (modeled) data does not have the correct amplitude.

The parameter `niterskip=` enables the fast algorithm when it is set larger than 1. The first instant time value `istart=` is run with `niter=` iterations. If `niterskip=` is set to a value > 1 the fast algorithm is in effect and the next `niterskip` iterations use only `niterec=2` iterations. After `niterskip` fast iterations the scheme uses the full `niter` iterations to avoid possible cumulative numerical instabilities caused by amplified artefacts. The scheme continues with `niterec` fast algorithm iterations and the cycle repeats itself. By setting `niterec=0` the scheme does not do any new iterations in the fast cycle and directly uses the result of the previous iteration. The `niterec=0` setting will work well if `niterskip=` is set to ≈`shift` samples and is possible due to limited bandwidth of the data.

The `T=` parameter is a switch to enable the T-MME algorithm. The options `plane_wave, src_angle, src_velo, xorig` use plane-waves as input shot record as explained in Meles et al. (2018, 2020).

The commands to reproduce all figures in this paper can be found in the directory `marchenko/demo/mme`. The README_PRIMARIES in that directory explains in detail how to run the scripts. A more complicated (lateral varying) model can be found in the directory `marchenko/demo/twoD`. This example will take several hours to compute the reflection data and is not discussed here.

Besides the new Marchenko primaries removal program the package also contains the earlier published finite difference modeling code, that is used to model all data in the examples, in directory `fdelmodc` (Thorbecke and Draganov, 2011), and the standard Marchenko programs (Thorbecke et al. (2017)). The directory `utils` contains programs to calculate a gridded model (`makemod`), source wavelets (`makewave`) and programs for basic processing steps.

REFERENCES

Meles, G.A., K. Wapenaar, and J. Thorbecke, 2018, Virtual plane-wave imaging via Marchenko redatuming: Geophysical Journal International, Vol. **214** (1), p. 508–519.

Meles, G.A., L. Zhang, J. Thorbecke, K. Wapenaar and E. Slob, 2020 Data-driven retrieval of primary plane-wave responses: Geophysical Prospecting, Vol. **68**, p. 1834–1846.

Thorbecke, J. and D. Draganov, 2011, Finite-difference modeling experiments for seismic interferometry: Geophysics, Vol. **76** (6), p. H1–H18.

Thorbecke, J., E. Slob, J. Brackenhoff, J. van der Neut, and K. Wapenaar, 2017, Implementation of the Marchenko method: Geophysics, Vol. **82** (6), p. WB29–WB45.