NAME

mteig − Creates a Network Sensitivity Solution (NSS) based on Ford et al. (2010,2012). NSS is the maximum variance reduction for points in source type space (e.g., eigenvalues on sphere or lune by Tape&Tape 2012).

SYNOPSIS

mteig par=(string) nthreads=(integer) nsim_eig=(integer) nsim_evec=(integer,deprecated) eigvec_fac=(integer) Mo=(float) fixz=(float) [no]color [no]doplt seed=(integer) [no]parallel title=(string) [[no]Add_user_Eig e0=(float) e1=(float) e2=(float)]

DESCRIPTION

Details on this implementation of the Ford et al. (2010) Network Sensitivity Solution on the Fundamental Lune can be found in Ichinose et al. (2020) https://doi.org/10.1029/2020JB020827

This NSS approach (e.g., Ford et al., 2010; Tape & Tape, 2017), with a spherical uniform random sampling in both eigenvalue and vectors on the fundamental lune, is different compared to other approaches using a uniform grid. We used a random sampling of **nsim_eig** number of eigenvalues from a uniform spherical distribution thereby avoiding oversampling near the poles. For every random point on the fundamental lune, we search **nsim_evec** number of eigenvectors by mapping a uniformly distributed random vector into a 3 Ã 3 rotation matrix. Given a point on the fundamental lune represented by a set of three-eigenvalues, each randomly chosen three-eigenvectors are then paired to form a moment tensor (MT). Then the MT is used to forward calculate synthetics and a Percent Varience Reduction (%VR) is estimated with the data. We finally sort the resulting **nsim_evec** number of calculated %VR for the maximum value and this is contoured for the given point in fundamental lune. The main reason for using random sampling instead of uniform sampling NSS approach (e.g., Ford et al., 2010) is that uniform sampling requires calculating %VR for 1011 to 108 synthetic seismograms, using strike, dip and rake formulation for each point on the source grid assuming fixed depth and M0, rather than a more reasonable 106 calculations using random sampling in eigenvalues and eigenvector formulation approach.

We begin to identify some rough broad %VR contours on the fundamental lune with only **nsim_eig** 200 points and after **nsim_eig** 2,000 points there are only small differences relative to using **nsim_eig** 4,000 or more points. The use of a minimum **nsim_evec** 4,000 eigenvectors is required and particularly important for reaching a smooth %VR contours near the double-couple (DC) region on the fundamental lune. Tests using **nsim_eig** = 2000 and **nsim_evec** = 4,000 results in 8-million forward calculations for each event NSS but with multithreading can be processed on a desktop computer within several minutes for four-station data set. A faster and more efficient alternative developed by Nayak and Dreger (2015) performs an iterative non-linear inversion of the eigenvectors for each point in source type space avoiding the issues of under- or over-sampling of the eigenvectors on the fundamental lune. Overall the forward calculated grid-search results are similar to their method in test comparisons, despite small differences of the %VR contours near the DC point on the eigen-sphere lune.

Update 2020: the parameter nsim_evec=(integer) is deprecated because it assigned a constant number of eigenvectors anywhere on the lune. The parameter eigvec_fac instead applies a Gaussian distribution whith more near the equator (lat=0, lon=0) and tapers off at the edges of the lune near the poles. This is more efficient since near the poles there is no need to search the eigenvector space since very little deviatoric moment left near the poles.

REQUIRED PARAMETERS

par={string}

This is the mtinv.par PARAMETER file see man page for glib2inv or mtinv for format.

nsim_eig=(integer)
#iter of createlamb() - creates random uniform distrib eigenvalues (best range 500-4000 use 2000)

nsim_evec=(integer)
[DEPRECATED! use eigvec_fac] #iter of randomEigVec() - creates random uniform distrib of eigenvectors

eigvec_fac=(integer)
factor that scales the amount of eigenvectors for each random point on lune (best range 15000 to 20000 use 17000)

seed=(integer)
seed > 0 random non-repeatable
seed < 0 random repeatable (useful for testing requiring same numbers)

Mo=(float)
Seismic Scaler Moment (dyne cm) to scale unit eigenvalues

fixz=(float)
Source Depth to select from Green's function library, this code only handles one fixed depth at a time.

OPTIONAL PARAMETERS

[no]parallel
[boolean] default off. Multithreading using posix pthreads

nthreads=(integer)
[default 2] if parallel on then this is the number of pthreads used up to NUM_MAX_THREADS=16

[no]color
[boolean] GMT lune plot %VR uses rainbow color pallette or gray scale [default off: grayscale]

[no]doplt
[boolean default yes] system call to execute GMT lune plot, otherwise just create the C-shell GMT script

title=(string)
text title in GMT Lune plot

[no]domech
[boolean default yes] add psmeca in GMT C-shell script to plot best fitting %VR MT focal mech

[no]norm
[boolean] distance normalization default is off

R0=(float)
normalize Green functions to distance of R/R0 [default R=1km]

[no]dump
[boolean] Write out data and syn vecs as SAC files when %VR > max_vred boolean [default off].

max_vred=(float)
Maximum percent variance reduction to write data and syn as SAC files range... -100 to 100% see [no]dump option

[no]Add_DC_iso
boolean default on - adds %VR special points to Lune (DC,ISO,CLVD,crack,LVD)

[no]Add_user_Eig
boolean default on - Adds a user supplied eigenvalues adds to Lune %VR NSS plot this option then requires Eigenvalues: e0={float} e1={float} e2={float}, mteig will compute a %VR for this spot.

e0=(float) e1=(float) e2=(float)
Add_user_Eig on requires a user specified eigenvalues, mteig will compute a %VR for this spot. mtbestfit copies the best fitting MT eigenvalues here.

EXAMPLE: mteig.csh

```
#!/bin/csh
### mteig assumes degree-of-freedom is 6 for full-MT to compute %VR on lune
cat >! mtinv.par << EOF
#### REGION COMMENT ############################
CM New Madrid, MO
#### Date and Origin Time ######################
OT 2021-11-18T02:53:04.000
#### Forward Calculations ######################
##    stk    dip    rak   Mw  evlo  evla    Z ##########
EV   26   66   -169 4.09    -90.543    36.9077  14
################################################################################################
# sta net loc model  np pas lf  hf  nt  dt   tr  tt v/d  mulfac used(Y/N)  ts0  weight ###
CGM3 NM 00 cus 3 2 0.075 0.15  512 0.10 0 0 d 1 y 0.0 1 Surf/Pnl # R=89.5 Az=60
PENM NM 00 cus 3 2 0.075 0.15  512 0.10 0 0 d 1 y 0.0 1 Surf/Pnl # R=96.0 Az=121
HENM NM 00 cus 3 2 0.075 0.15  512 0.10 0 0 d 1 y 0.0 1 Surf/Pnl # R=97.6 Az=102
GNAR NM 00 cus 3 2 0.075 0.15  512 0.10 0 0 d 1 y 0.0 1 Surf/Pnl # R=114.0 Az=155
CCM  IU 00 cus 3 2 0.075 0.15  512 0.14 0 0 d 1 y 0.0 1 Surf/Pnl # R=141.7 Az=334
CCM  IU 10 cus 3 2 0.075 0.15  512 0.14 0 0 d 1 y 0.0 1 Surf/Pnl # R=141.7 Az=334
SIUC NM 00 cus 3 2 0.075 0.15  512 0.14 0 0 d 1 y 0.0 1 Surf/Pnl # R=147.5 Az=52
SLM  NM 00 cus 3 2 0.075 0.15  512 0.16 0 0 d 1 y 0.0 1 Surf/Pnl # R=193.9 Az=7
### NOTE! stas not used, are commented out and not loaded by mteig ###
### because there is no prediction only millions of forward calcs ###
EOF

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel
### PROCESS DATA ###
sacdata2inv par=mtinv.par path=../Data respdir=../Resp noverbose nodumpsac parallel

time mteig par=mtinv.par nthreads=8
        nsim_eig=2000 nsim_evec=4000 eigvec_fac=17000
        Mo=1.698473e+22 fixz=14
```

color doplt seed=1 parallel
title="2021-11-18 New Madrid, MO"
Add_user_Eig e0=+1.69847 e1=+0.171992 e2=-1.29014

SEE ALSO

mkgrnlib(1), *glib2sac*(1), *mtinv*(1)