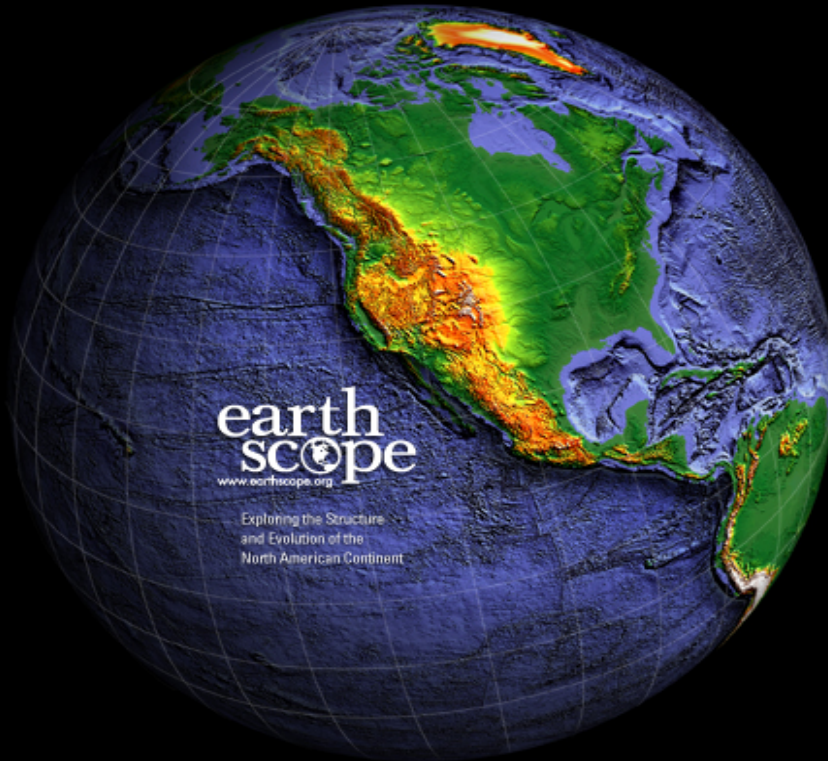
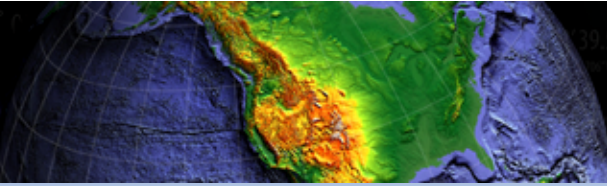


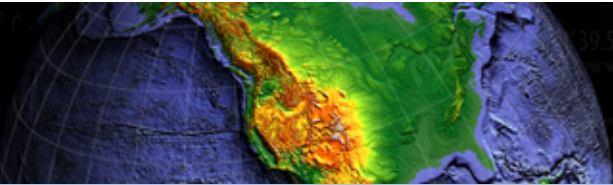
# Preliminary Results of the Moment Tensor Code in Antelope

**Antelope Users' Group Meeting  
University of Nevada Reno Campus  
October 23-25, 2012**



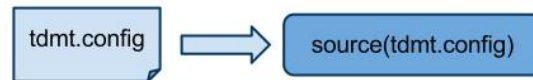


- Get origins from Datascope tables.
- Subset stations in quadrants.
- Get Green's Functions from Datascope based on distance and depth of event.
- Extract, rotate and filter data from stations.
- Reject stations with bad cross-correlation.
- Invert the MT and extract the eigen values/vectors.
- Update Datascope with results.

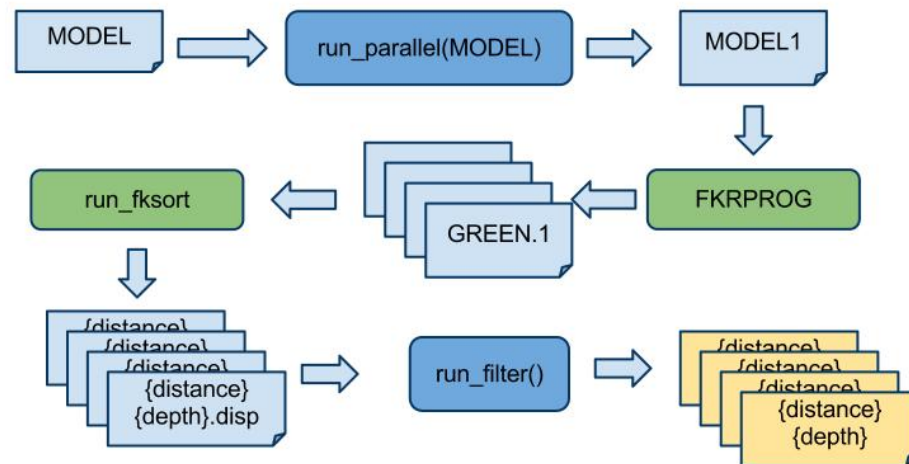


# Previous Model

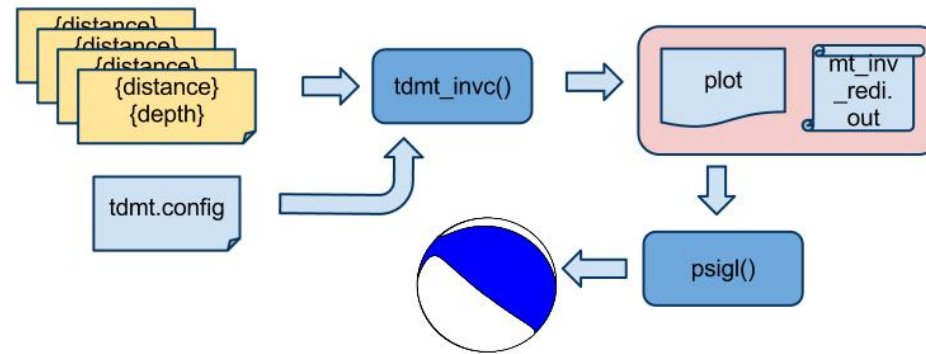
## Configure Environment

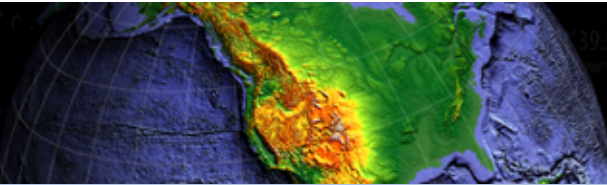


## Get the Green functions

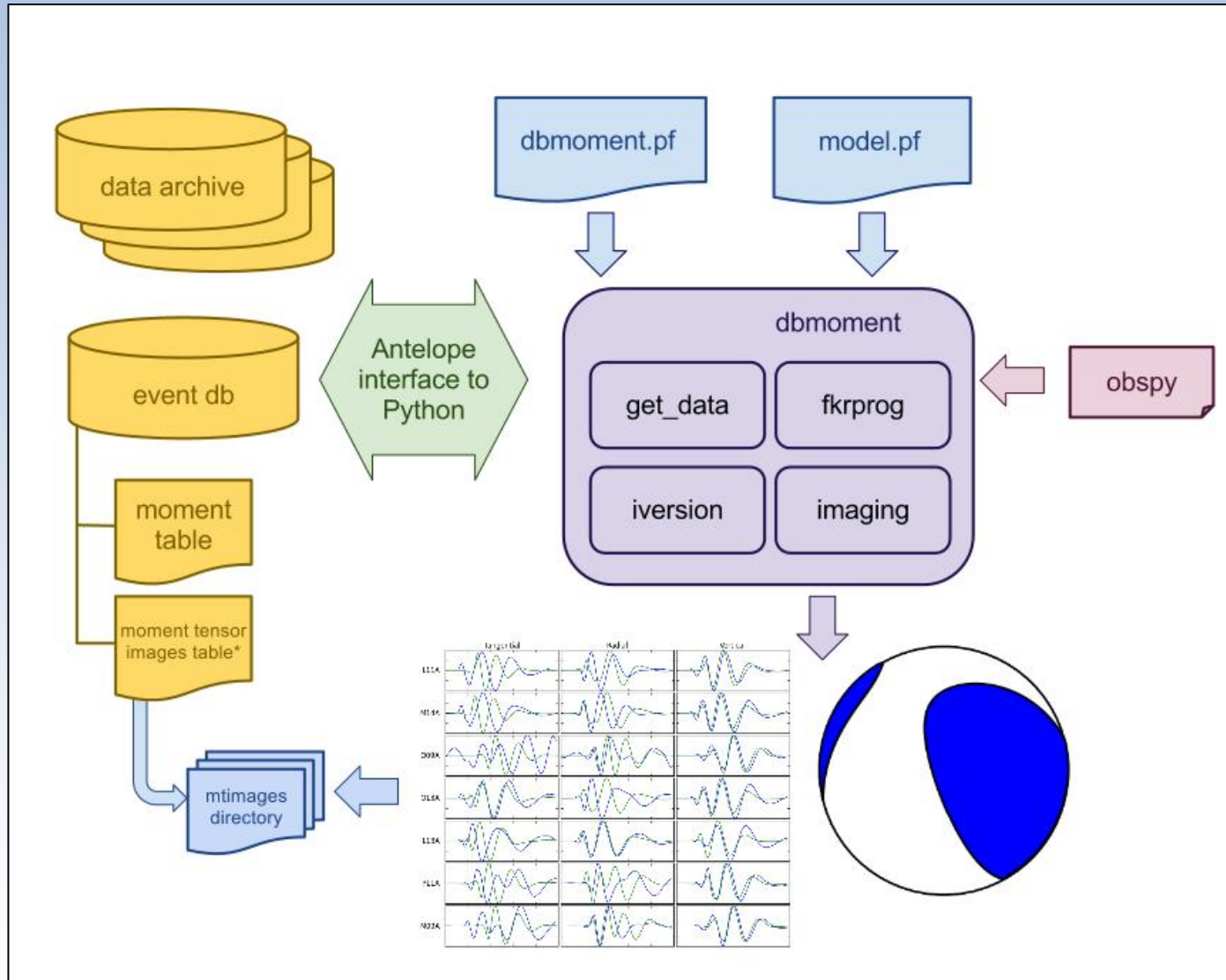


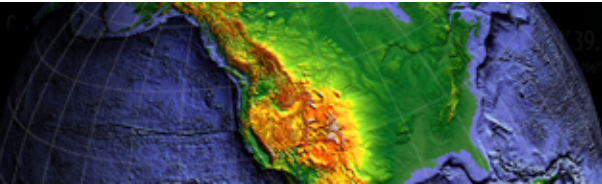
## Run Moment Tensor Inversion





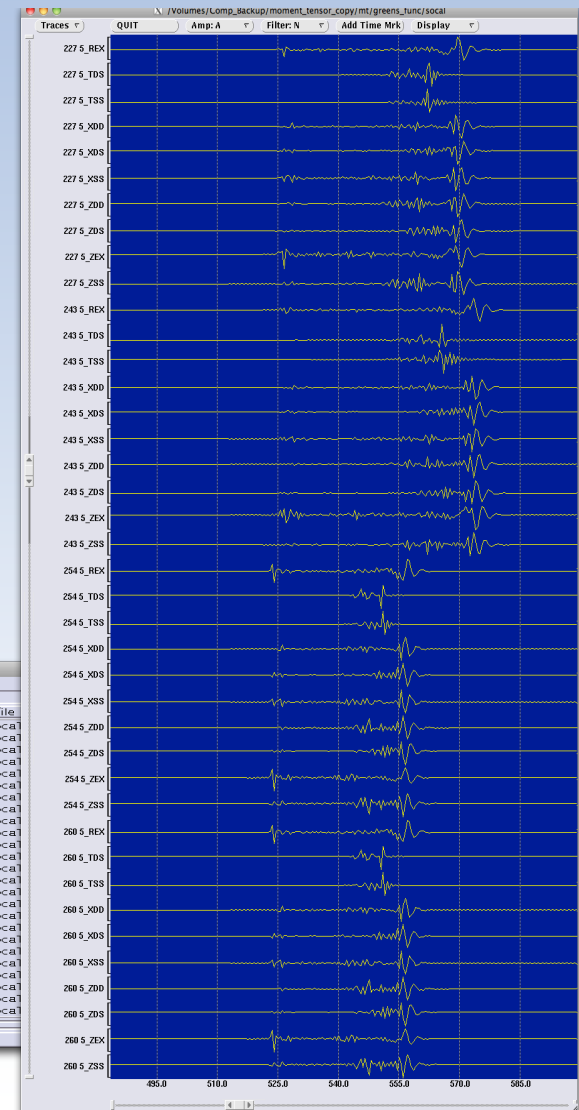
# Current Model

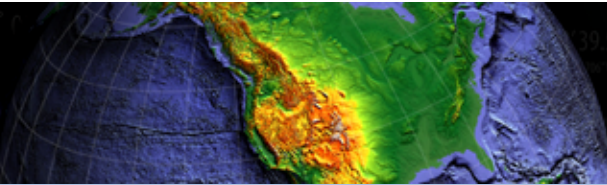




Green's Functions are build dynamically upon request if not already present in archive. Newly constructed functions are stored on a database referenced by a wfdisc table.

sta	chan	time	endtime	nsamp	samprate	calib	instype	sectype	datatype	dir	dfile	
138	S_XDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	138_5_social
138	S_XSS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	138_5_social
138	S_ZDD	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	138_5_social
138	S_ZDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	138_5_social
138	S_ZSS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	138_5_social
172	S_REX	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	172_5_social
172	S_TDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	172_5_social
172	S_TSS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	172_5_social
172	S_XDD	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	172_5_social
172	S_ZDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	172_5_social
172	S_ZSS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	172_5_social
200	S_REX	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_TDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_TSS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_XDD	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_XDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_XSS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_ZDD	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_ZDS	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social
200	S_ZEX	1/01/1970 (001)	0:00:01.00000	1/01/1970 (001)	1:08:17.00000	2048	2.0000000	1	V	as	social/5	200_5_social





## dbmoment.pf

```
# parameter file for dbmoment

# Name of the library to use
# for Green'sFunction class
gf_lib fkrprog

# Name of the library to use
# for Inversion class
inv_lib inversion

# Name of the library to use
# for Event class
data_lib get_data

# channel to use in MT-inversion, default is LH.
chan_to_use LH.*

# Name of the model parameter file
model_name SOCIAL_MODEL
# Use displacement (d) or velocity (v)
model_type v

...
```



## inversion.py

```
from __main__ import * # Get all the libraries from parent

class MomentTensor():
    """
    Class for building moment tensors and doing the inversion
    """
    def __init__(self, distance_weighting, isoflag, trim_value, verbose=False, debug=False):
+--- 8 lines: -----

    def _log(self,message):
+--- 7 lines: -----

    def construct_data_matrix(self, stachan_traces):
+--- 37 lines: -----

    def plot_cross_cor(self, a, b, shift, maxval,xcor=None,a_name="b_name="):
+--- 51 lines: -----

    def _cross_cor(self, a, b):
+--- 22 lines: -----

    def get_time_shift(self, data, greens ,delta = False):
+--- 64 lines: -----

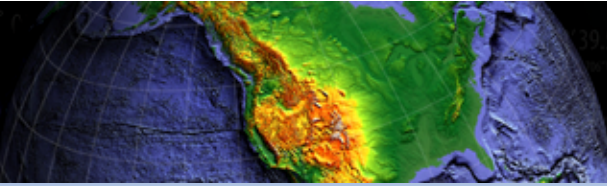
    ...

    def determine_solution_vector(self, dict_AIV, dict_B):
+--- 71 lines: -----

    def decompose_moment_tensor(self, matrix_M):
+---166 lines: -----

    def fitcheck(self, dict_g, dict_s, matrix_M, m0, ev2sta, size):
+--- 74 lines: -----

    def quality_check(self, vr):
+--- 22 lines: -----
```

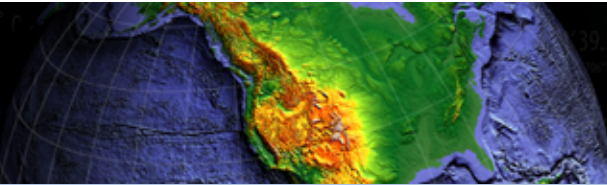


1. Download from Github
2. Install any additional libraries (ObsPy)
3. Build the contributed code source
4. Configure the command for your system by editing the Antelope configuration file (dbmoment.pf ) - Run the executable dbmoment with the following options:

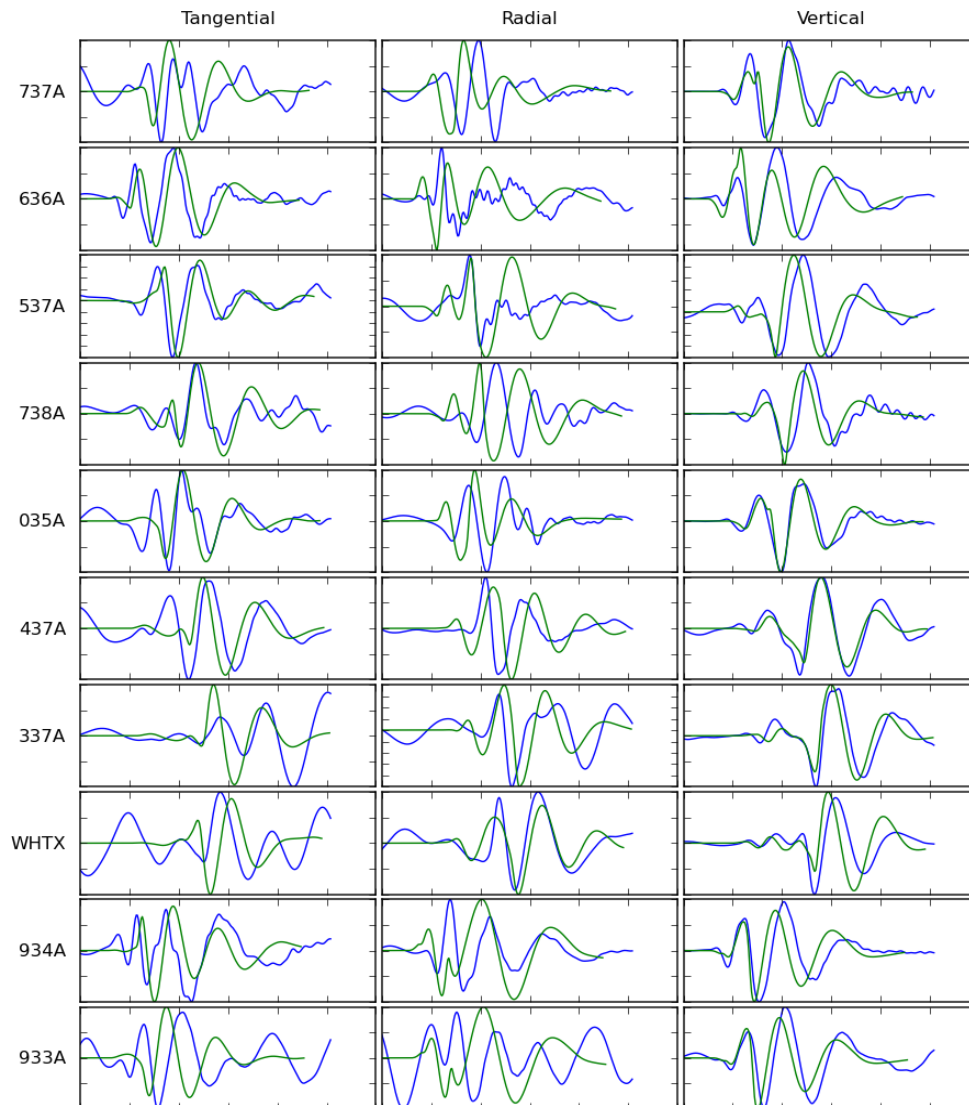
Usage: dbmoment [-vVde] [-p pfname] orid

Options:

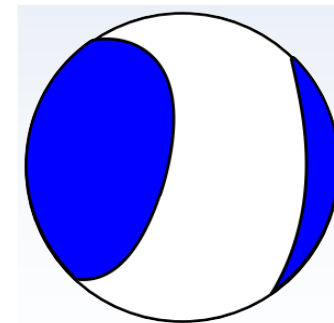
- h, --help
- v, --verbose
- V, --veryverbose
- d, --debug
- p PF, --pf=PF
- e EVENT, --event=EVENT
- s SELECT, --select=SELECT
- r REJECT, --reject=REJECT



# Results

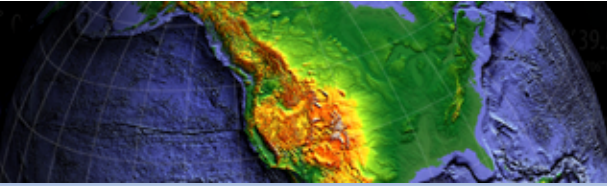


orid = 3830971  
 time = 10/20/2011 12:24:40  
 Strike 1 = 78  
 Rake 1 = -72  
 Dip 1 = 72  
 Strike 2 = 211  
 Rake 2 = -137  
 Dip 2 = 24  
 Mo = 1.340E+23  
 Mw = 4.718  
 % DC = 28.644  
 % CLVD = 71.356  
 % ISO = 0.000  
 VR = 3.624E-09  
 VAR = 3.624E-09









39.53°N 110.12 20.02°W

0.153 25.41 052000°N 1.90 20.30 052000°W

46.644°N 114.6° 0' 43.08" W

Code

Github:

[https://github.com/jreyes1108/antelope\\_contrib/tree/moment\\_tensor](https://github.com/jreyes1108/antelope_contrib/tree/moment_tensor)