Seismic Moment Tensor Inversion

Maximilian Williams

November 2021

What is this?

This document is a brief explanation detailing my thinking for a small program I wrote to invert seismogram data for the seismic moment tensor of the source.

The goal

The goal of the program is to invert for the seismic moment tensor given synthetic seismogram data, precomputed Green's functions and the locations of the stations.

Assumptions

- The seismograms are cleaned, so they have average of zero, band pass filters were already applied etc...

Conventions

The seismic moment tensor will always be formmated in cartesian coordaintes.
The greens functions and seismograms will have vertical (Z), radial (R) and transverse (T) directions.

Explanation

We have siesmograms each with three components, $d_Z(p,t)$ vertical (Z), $d_R(p,t)$ radial (R) and $d_T(p,t)$ transverse (T). Here p is the location of the station which is at azimuthal angle θ_p from the event. We model the source as a set of force couples whose configuration is encoded in the seismic moment tensor M. The synthetic seismograms are then given by:

$$\begin{aligned} d'_Z(p,t) = M_{xx}(ZSS/2*\cos 2\theta_p - ZDD/6 + ZEP/3) \\ +M_{yy}(-ZSS/2*\cos 2\theta_p - ZDD/6 + ZEP/3) \\ +M_{zz}(ZDD/3 + ZEP/3) \\ +M_{xy}(ZSS\sin 2\theta_p) \\ +M_{xz}(ZDS\cos\theta_p) \\ +M_{yz}(ZDS\sin\theta_p) \end{aligned} \tag{1}$$

$$\begin{aligned} d'_R(p,t) = M_{xx}(RSS/2*\cos 2\theta_p - RDD/6 + REP/3) \\ +M_{yy}(-RSS/2*\cos 2\theta_p - RDD/6 + REP/3) \\ +M_{zz}(RDD/3 + REP/3) \\ +M_{xy}(RSS\sin 2\theta_p) \\ +M_{xz}(RDS\cos\theta_p) \\ +M_{yz}(RDS\sin\theta_p) \end{aligned} \tag{2}$$

$$\begin{aligned} d'_T(p,t) = M_{xx}(TSS/2*\sin 2\theta_p) \\ +M_{yy}(-TSS/2*\sin 2\theta_p) \\ +M_{xy}(-TTS\cos 2\theta_p \\ +M_{xz}(TDS\sin\theta_p) \\ +M_{yz}(-TDS\cos\theta_p) \end{aligned} \tag{3}$$

Here SS is the vertical strike-slip, DD the $\frac{\pi}{4}$ dip slip, DS the vertical dip-slip and EP the explosive component of the Green's functions with Z, T and R being the vertical, transverse and radial components. This method of computing the synthetic seismograms is from

lecture 2 week 5 of PHYS3070.
The computation of the synthetic data $d'_n(p,t)$ $n \in Z, R, T$ is the forwards problem. We will call obtaining the M such that d and d' are the most similar the inverse problem.

Program Inputs

The main program is main.py. It can be run using "python3 main.py".
The program takes two types of files to run. The first set of files are to be located in a folder called "Greensfunctions". Within the folder "Greensfunctions" are located greens functions in .sac format. The files are named XX.XXXX.00.DD.0000.YYY. Here X entries indicate the station, Ds indicate the depth and Ys indicate the component of the greens functions. For example, BK.FARB.00.5.0000.REX is the greens function between the event at a depth of 5 km and the station BK.FARB with component REX (radial explosive). The other file input is naming and location information about the stations that we will use to do the inversion. This is called "stationInformation.csv". This csv file contains two columns. The first column contains the names of the station used for the inversion. The second column contains the azimuthal angle the station is at relative to the event location. Finally, a depth for the source is specified manually in the program. By default it is 5km.

Method

The program does not take real seismograms, but we still want to show that it can invert for the moment tensor. To do this we arbitrarily define a target moment tensor M_t. We then perform the forwards problem to obtain the target data d_t. We now treat d_t as real data and try to find the moment tensor associated with it which is ofcourse ideally M_t.
I have chosen to view the inverse problem as an optimization problem. Specifically, I see the inverse problem as optimizing M such that it maximises $g(d', d_t)$ where $g(\cdot,\cdot)$ is the score metric defined in the appendix. Ideally, this returns M close to M_t. We note that we tend not to have large numbers of stations and the seismograms tend not to be very long. Therefore, we can expect computation of the forwards problem to be very fast on modern computers. For this reason, I chose to use a genetic algorithm to optimize M.

Genetic Algorithms

Genetic algorithms are a broad class of algorithms which use methods inspired by natural evolution to optimize. Here I used a basic variant. The algorithm generates a random population of seismic moment tensors. (1) For each seismic moment tensor in the population, the forwards problem is computed and the score computed (defined in appendix). The highest scoring seismic moment tensors are then bred together to produce a new population [1] (2). (1) to (2) is then repeated *epochs* number of times. Once complete, the average seismic moment tensor from the last population is returned. This algorithm is detailed below.

Algorithm 1

```
$Ms \leftarrow \{\}$
$S \leftarrow \{\}$
epoch $\leftarrow 0$
$Ms \leftarrow generateRandomPopulation(populationSize)$
**while** epoch $<$ epochs **do**
    $S \leftarrow scorePopulation(Ms)$
    $Ms \leftarrow breedPopulation(Ms, S)$
    epoch $\leftarrow$ epoch $+1$
**end while**
**return** average(Ms)
```

Results

I tested three inversions in my code. A DC mechanism, explosive and CLVD mechanism.

DC results

The target moment tensor is:

$$\begin{pmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4}$$

[1]When we breed the population, we pick two moment tensors and average them. The selection of individuals is done probabilistically, with the probability of being picked to breed being proportional to the score. This breeding process is repeated over and over until we generate a new population.

The program inverts the moment tensor and finds it to be:

$$\begin{pmatrix} -0.07 & 0.48 & -0.00 \\ 0.48 & 0.04 & 0.00 \\ -0.00 & 0.00 & -0.04 \end{pmatrix} \tag{5}$$

This closely matches the target moment tensor.

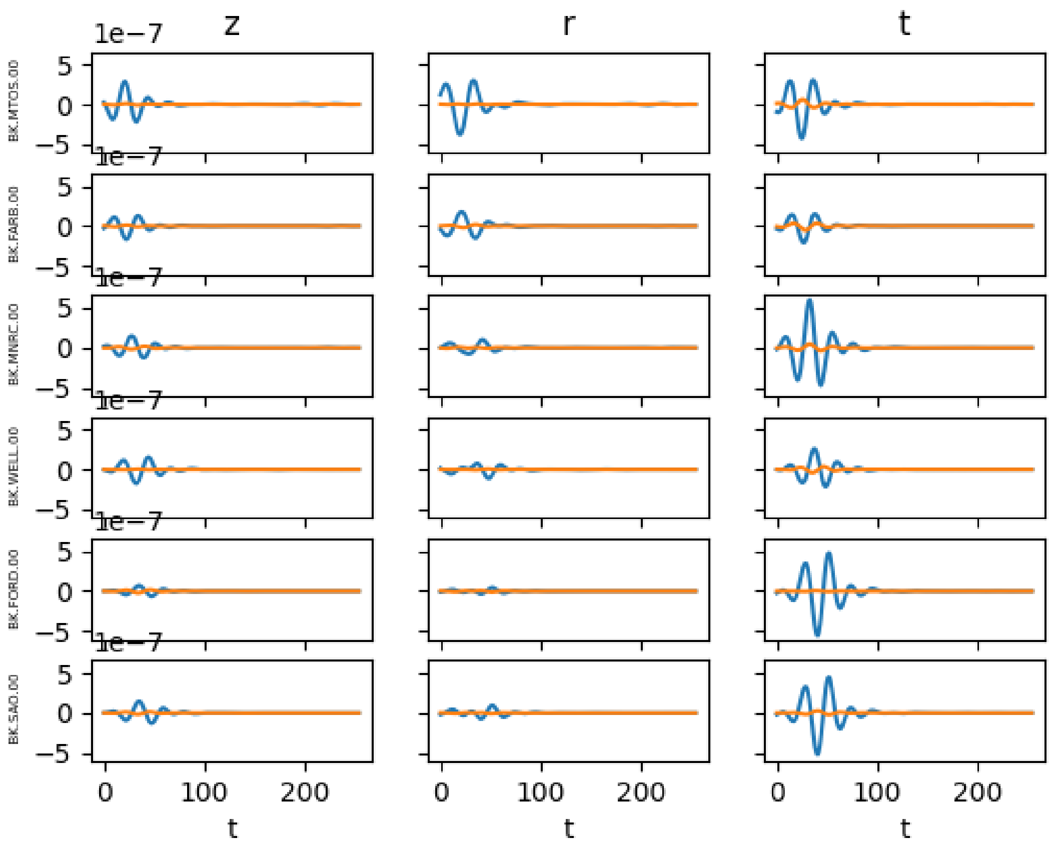

Figure 1: target (blue) and average seismogram (orange) before optimization with DC mechanism

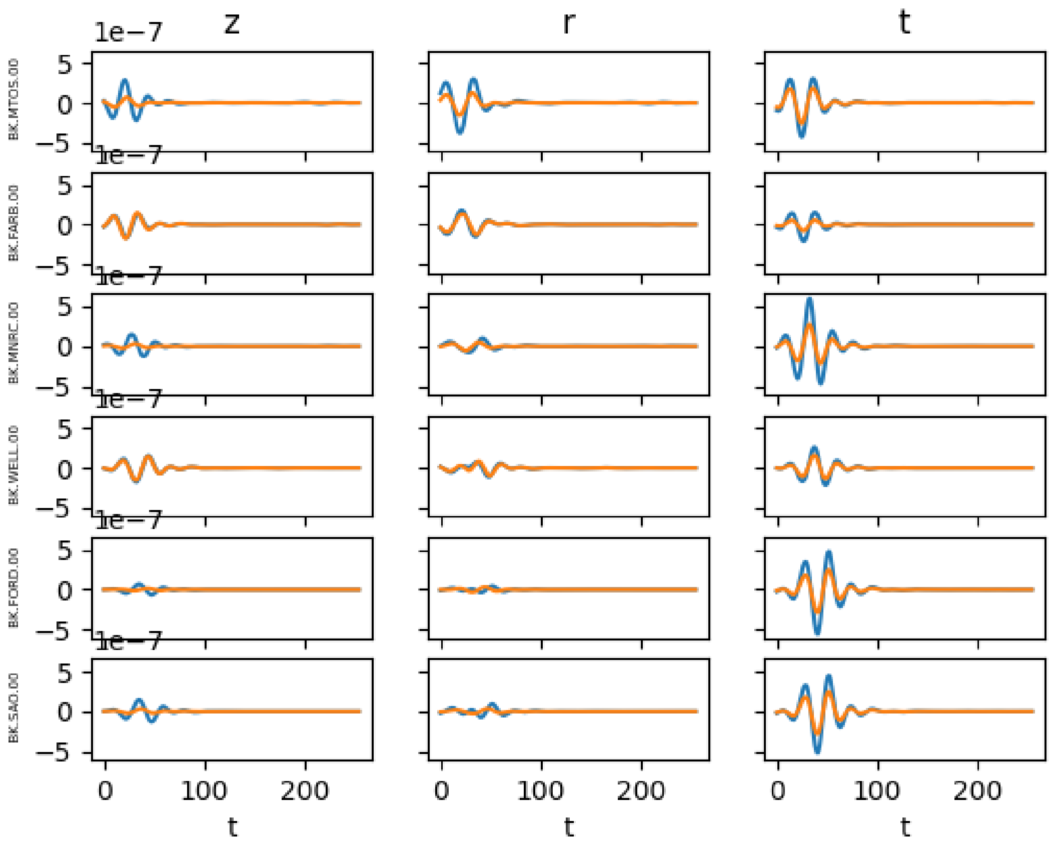

Figure 2: target (blue) and optimized seismograms (orange) for iteration 4 of genetic algorithm with DC mechanism

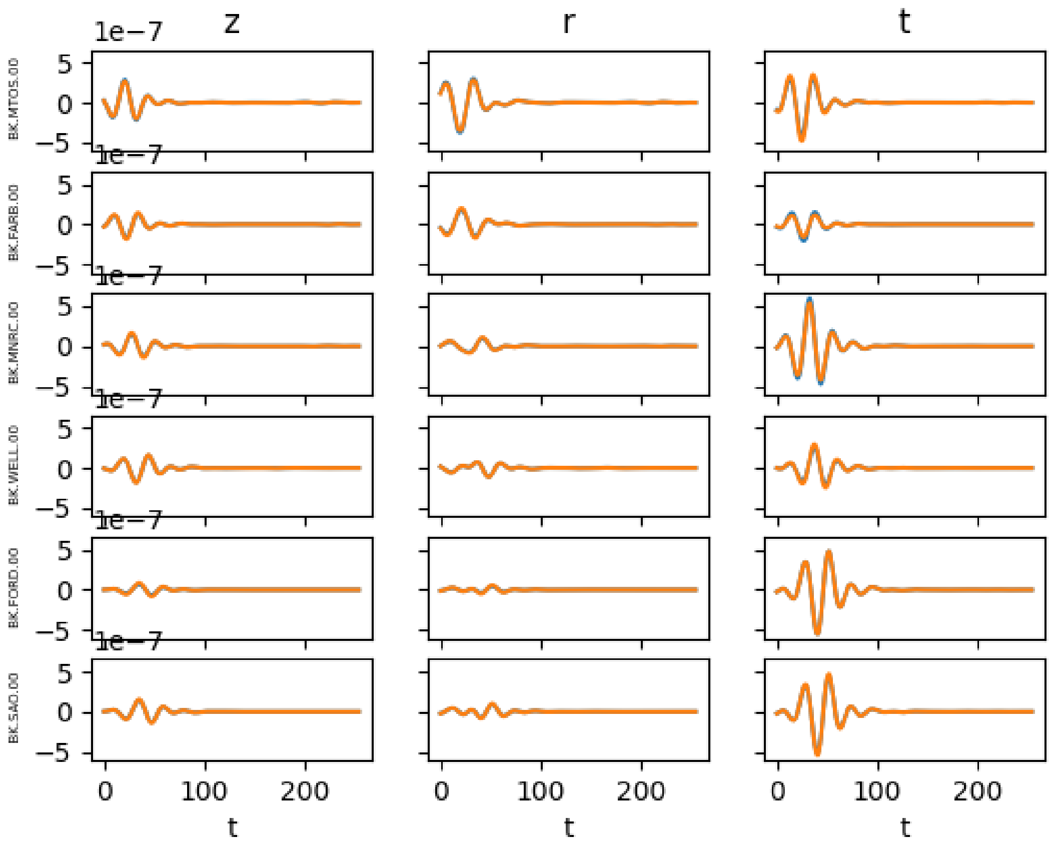

Figure 3: Target (blue) and optimized seismograms (orange) for iteration 50 of genetic algorithm with DC mechanism

Figures 1, 2, 3 show that as the genetic algorithm iterates the optimized seismograms approach their target. This convergence can also be seen in figure 4 which shows the score (similarity between target and optimized seismograms) increases as we iterate the program. This increase in score coencides with a decrease in deviation across the population of moment tensors as shown in figure 5 and an increasing similarity between the average Moment tensor and the target moment tensor shown in figure 6. These features show the genetic algorithm converged to the correct solution.

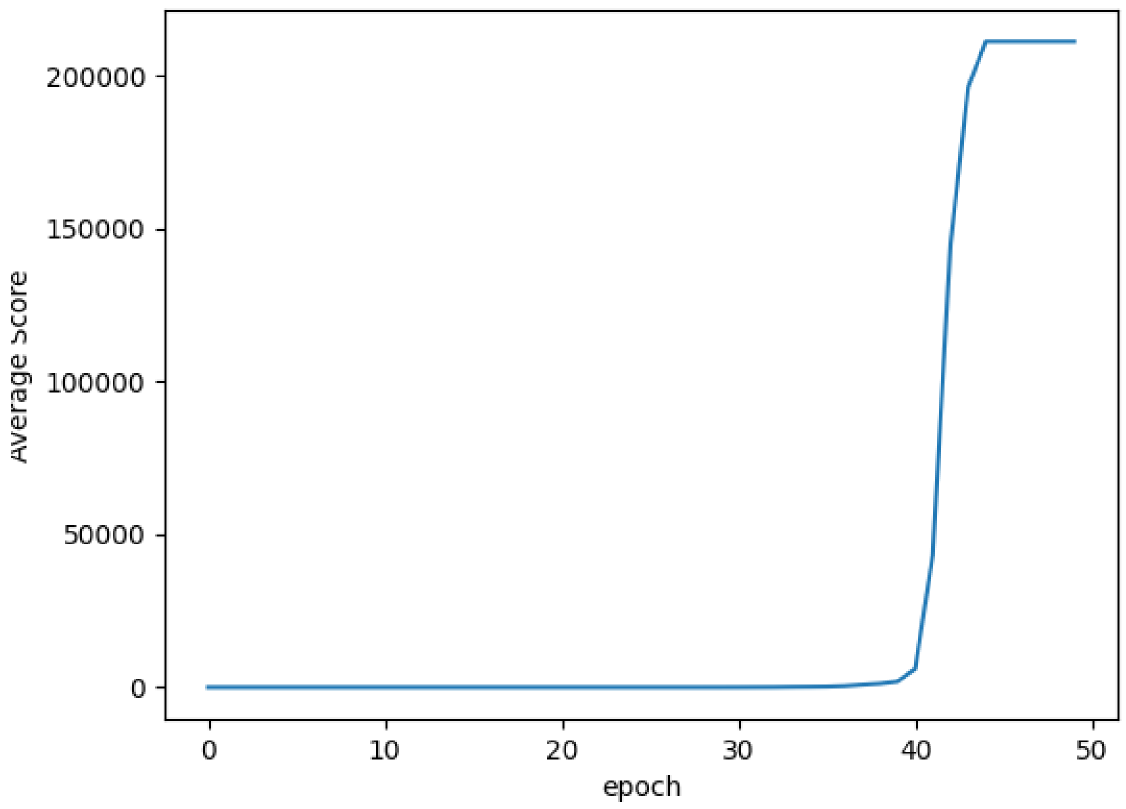

Figure 4: Average score of genetic population against number of generations (epochs) for DC target

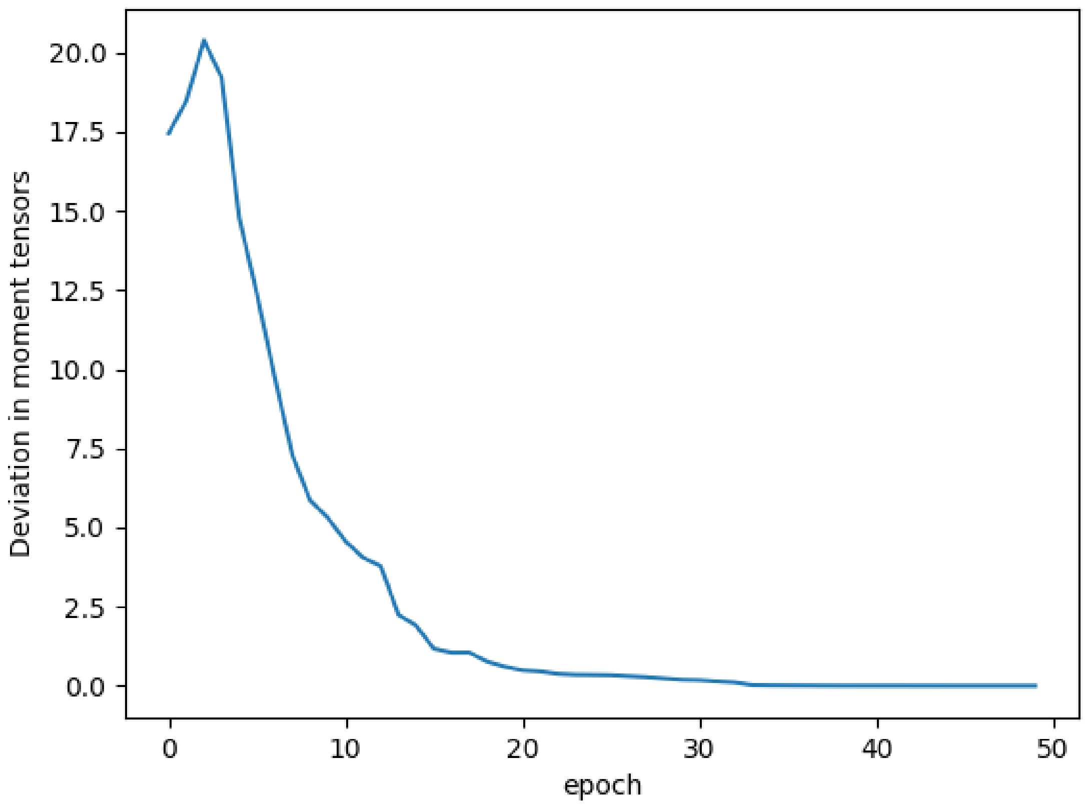

Figure 5: Deviation in moment tensor population against number of generations (epochs) for DC target

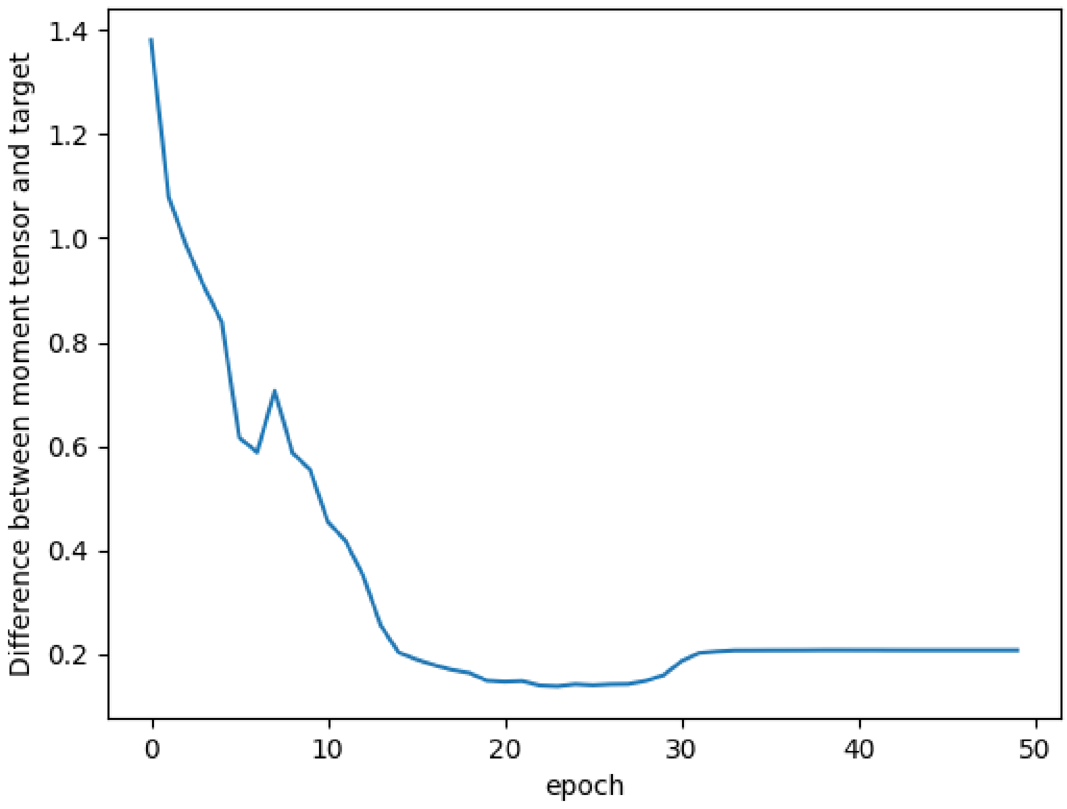

Figure 6: Similarity between average moment tensor and number of generations (epochs) for DC target

Isotropic result

The target moment tensor is:

$$\begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} \tag{6}$$

The program finds a moment tensor:

$$\begin{pmatrix} 0.20 & -0.00 & 0.00 \\ -0.00 & 0.20 & -0.00 \\ 0.00 & -0.00 & -0.19 \end{pmatrix} \tag{7}$$

This inversion is not very accurate, it has optimized all components with value 0 correctly, but gotten the wrong sign on m_{zz} and the magnitudes for m_{xx} and m_{yy} are substantially different. It has found a pure CLVD moment tensor rather than an isotropic one! However, as seen in figure 7, 8, 9 the optimized seismograms still converges to the target siesmograms. Figures 10, 11, 12 show similar behviour to their DC counterparts confirming that the genetic algorithm did converge. However, the high gradient near the end of the simulation (40-50th iteration) in figure 10 suggests the genetic algorithm could optimize further. This optimization would most likely be a refinement of the final moment tensor:

$$\begin{pmatrix} 0.20 & -0.00 & 0.00 \\ -0.00 & 0.20 & -0.00 \\ 0.00 & -0.00 & -0.19 \end{pmatrix} \tag{8}$$

and would still be substantially different from the target moment tensor due to the small deviation in the population of moment tensors on the final iteration (iteration 49) seen in figure 11.

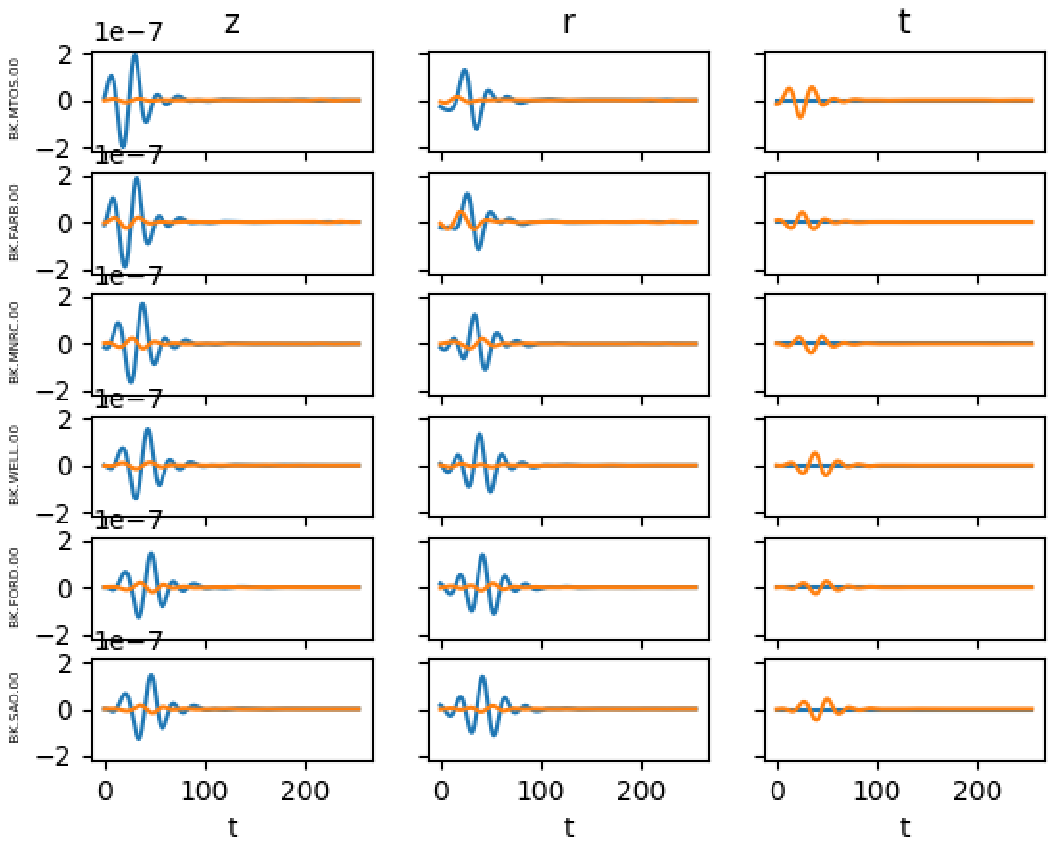

Figure 7: target (blue) and average seismogram (orange) before optimization with ISO mechanism

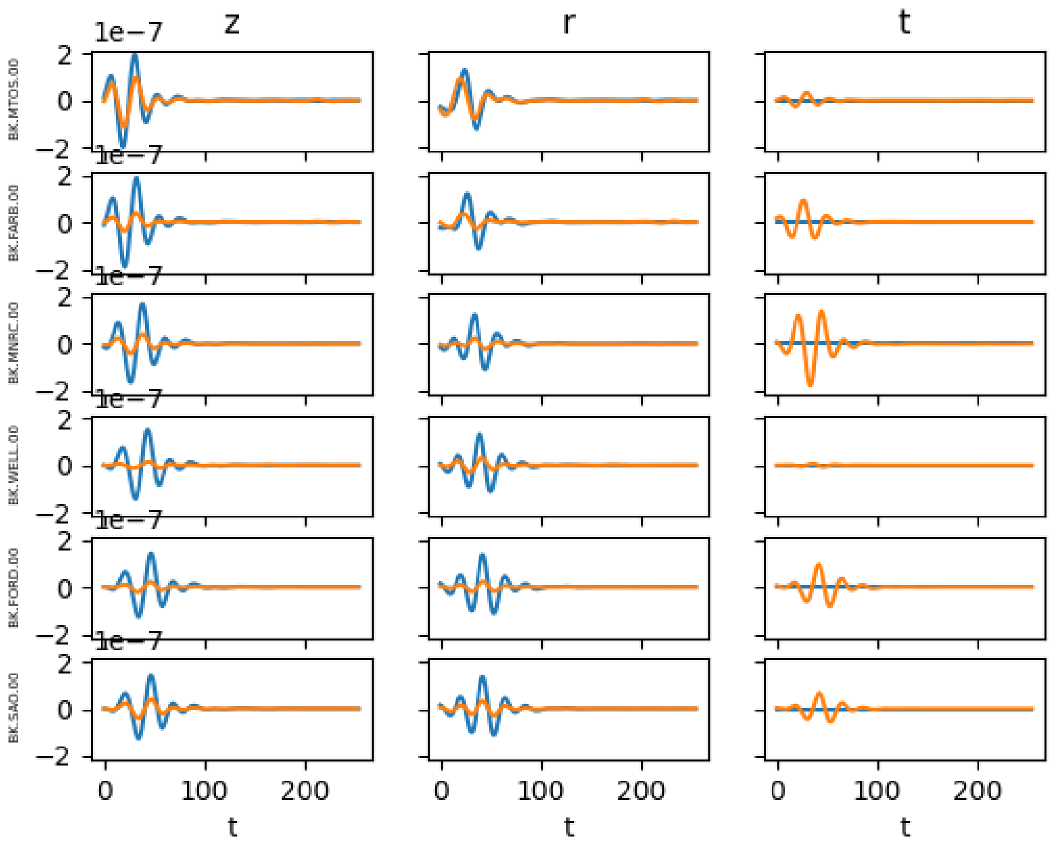

Figure 8: target (blue) and optimized seismograms (orange) for iteration 4 of genetic algorithm with ISO mechanism

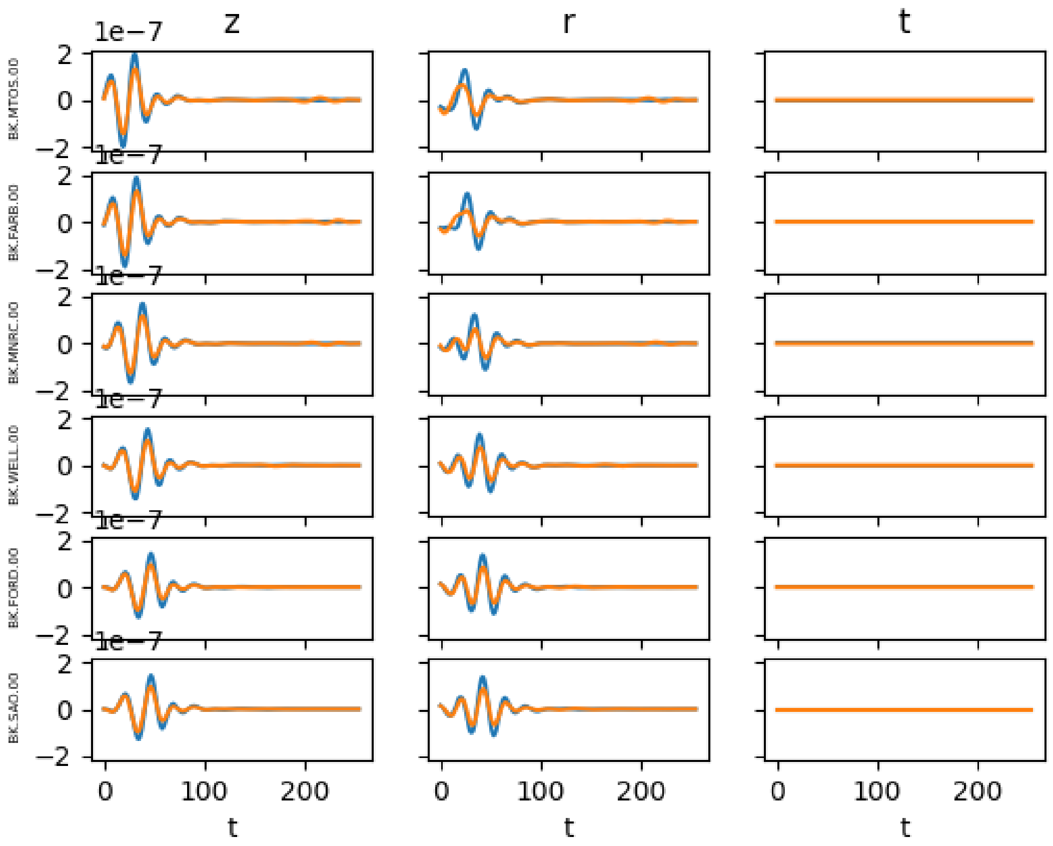

Figure 9: Target (blue) and optimized seismograms (orange) for iteration 50 of genetic algorithm with ISO mechanism

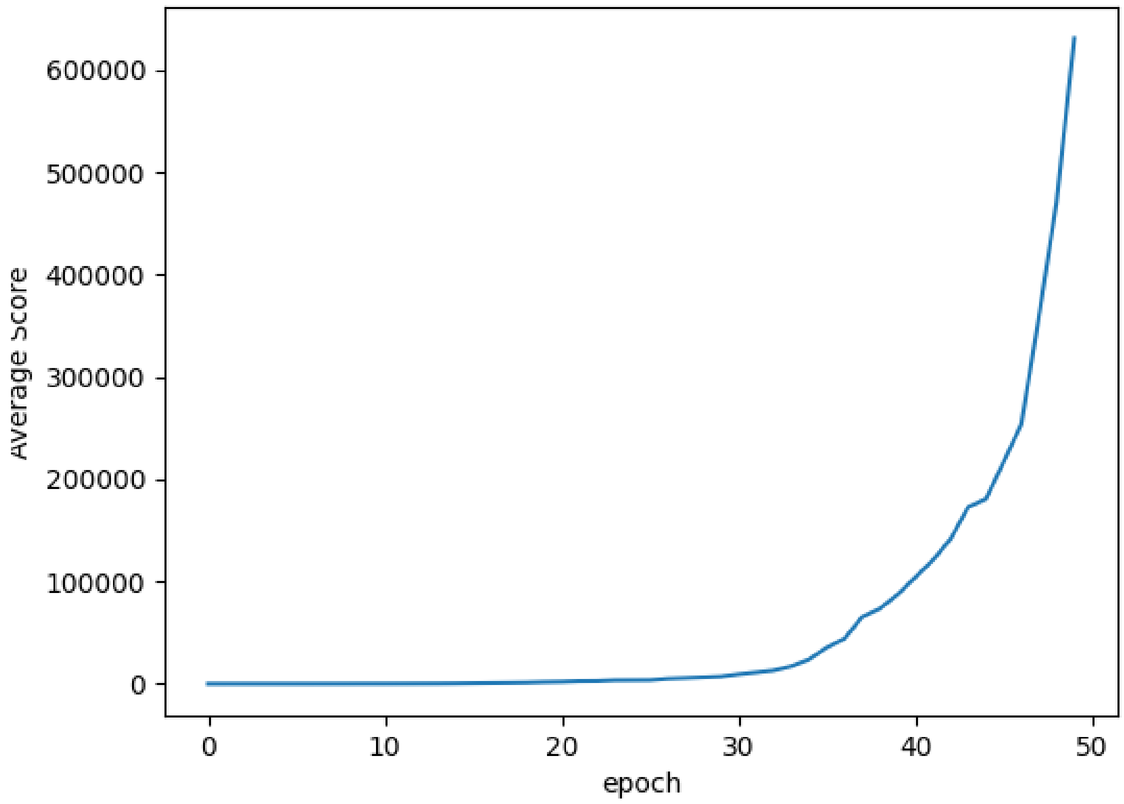

Figure 10: Average score of genetic population against number of generations (epochs) for ISO target

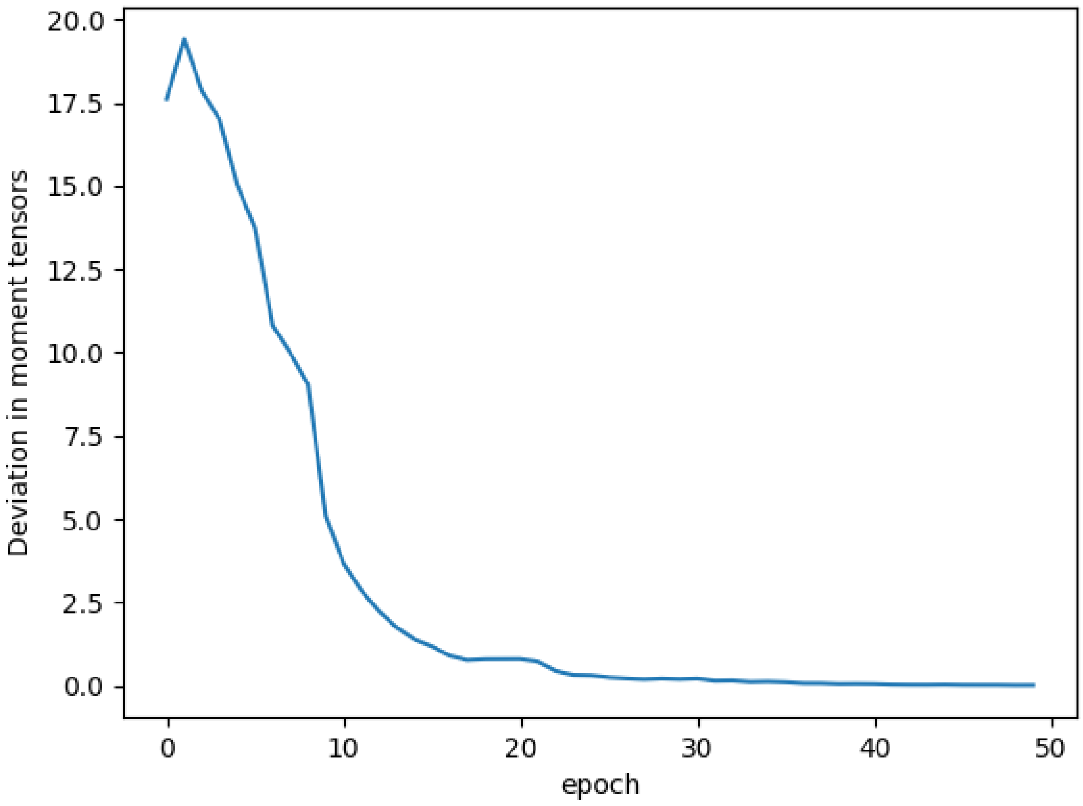

Figure 11: Deviation in moment tensor population against number of generations (epochs) for ISO target

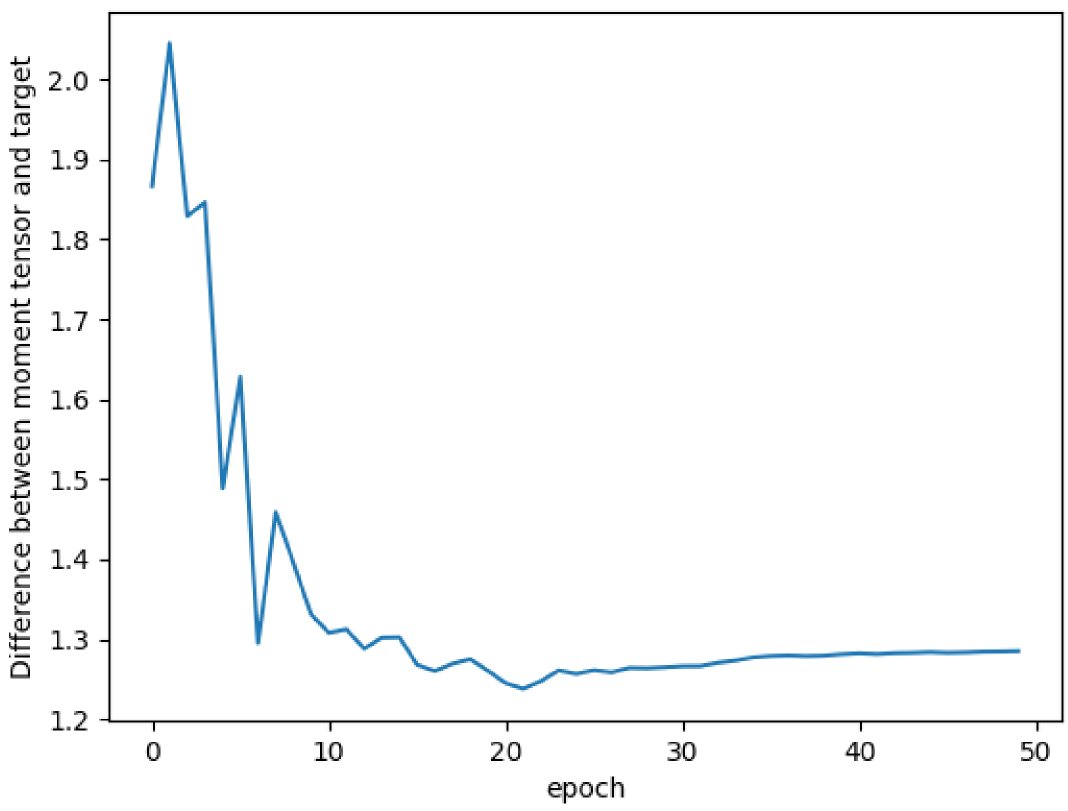

Figure 12: Similarity between average moment tensor and number of generations (epochs) for ISO target

CLVD results

The target moment tensor is:

$$\begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & -0.5 \end{pmatrix} \tag{9}$$

The program finds a moment tensor:

$$\begin{pmatrix} -0.03 & -0.00 & 0.13 \\ -0.01 & 0.06 & -0.04 \\ 0.12 & -0.04 & -0.14 \end{pmatrix} \tag{10}$$

The optimized moment tensor is again substantually different from the target. The seismograms also did not converge correctly as seen in figure 13, 14, 15. The genetic algorithm did converge however, as shown by the decreasing deviation in the population of moment tensors (figure 18) and the flattening of the score in figure 16. Its just that this convergence was not to a good solution.

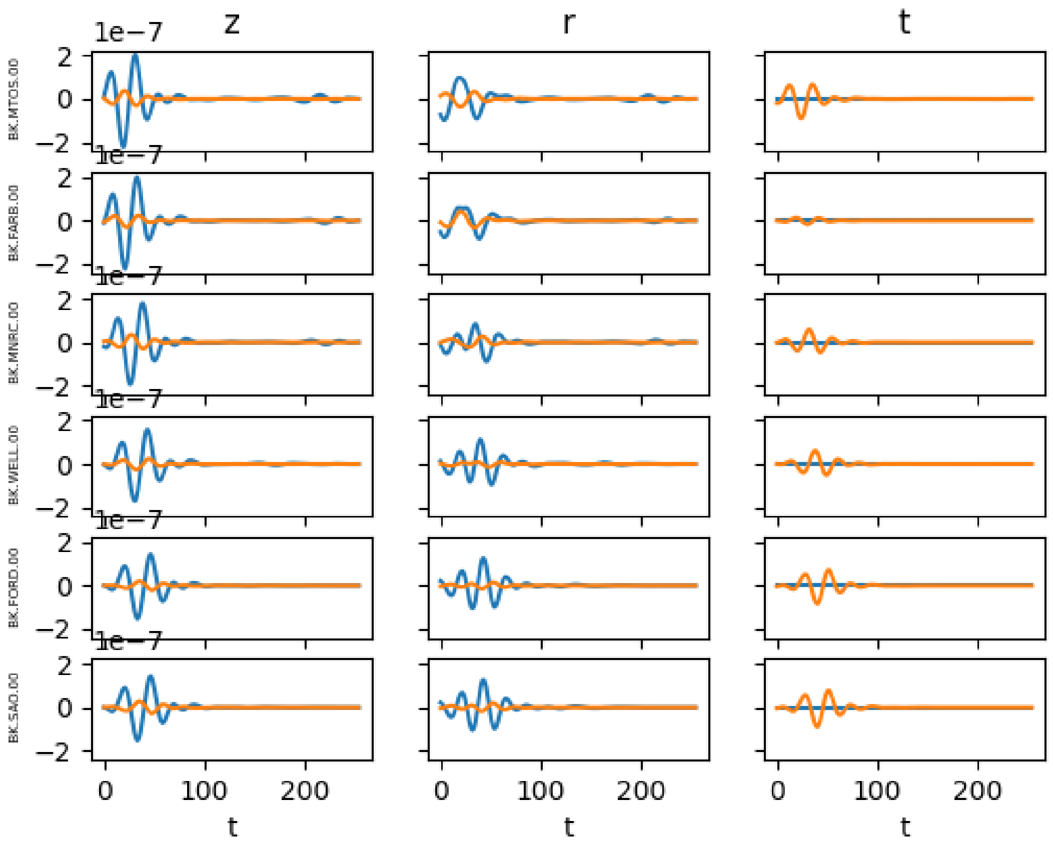

Figure 13: target (blue) and average seismogram (orange) before optimization with CLVD mechanism

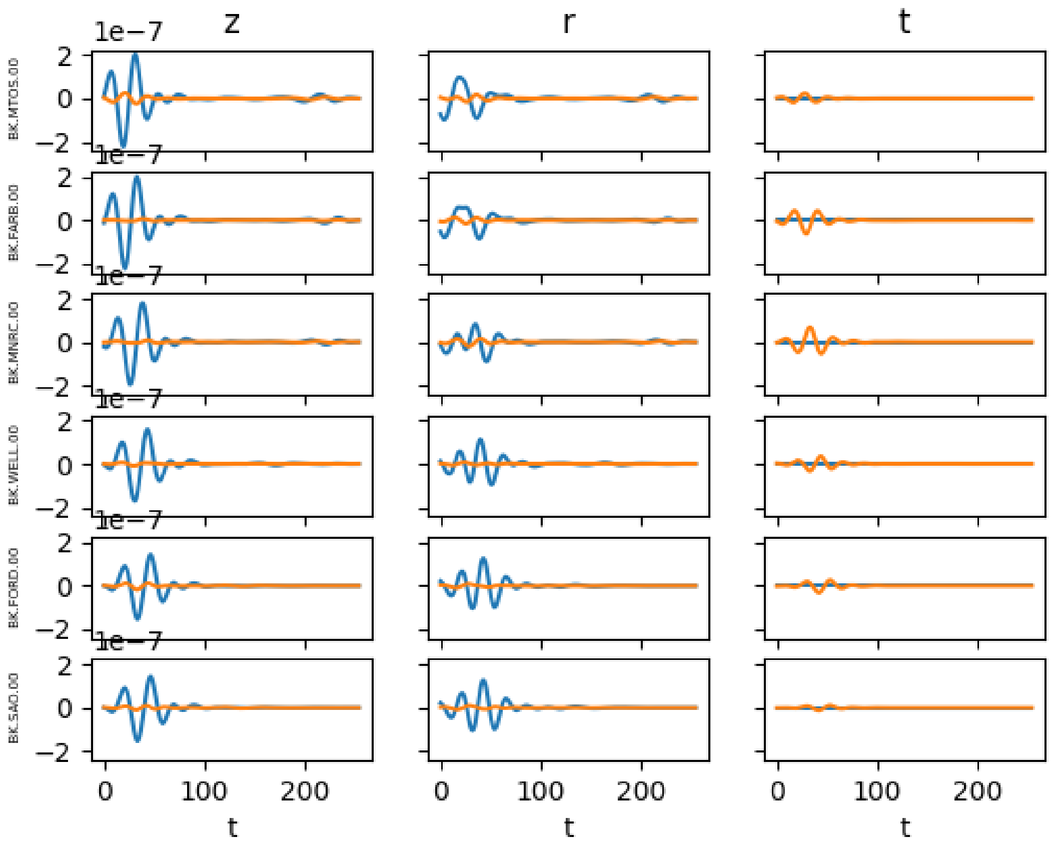

Figure 14: target (blue) and optimized seismograms (orange) for iteration 4 of genetic algorithm with CLVD mechanism

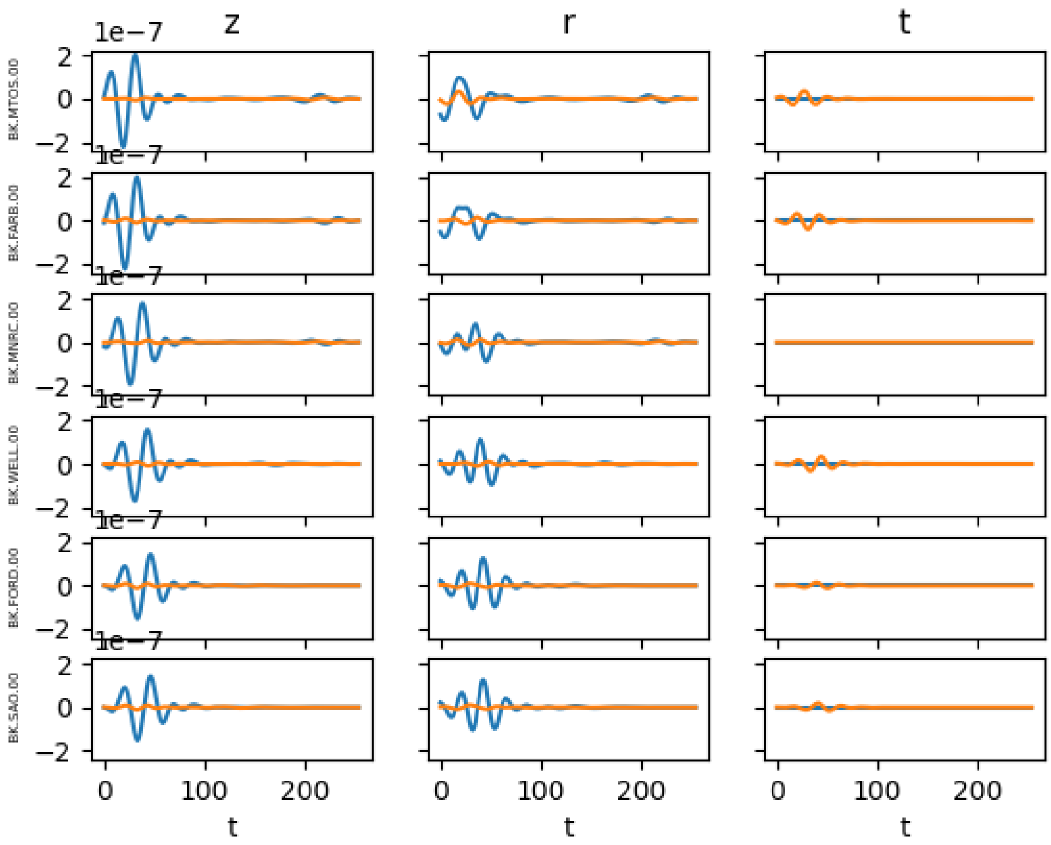

Figure 15: Target (blue) and optimized seismograms (orange) for iteration 50 of genetic algorithm with CLVD mechanism

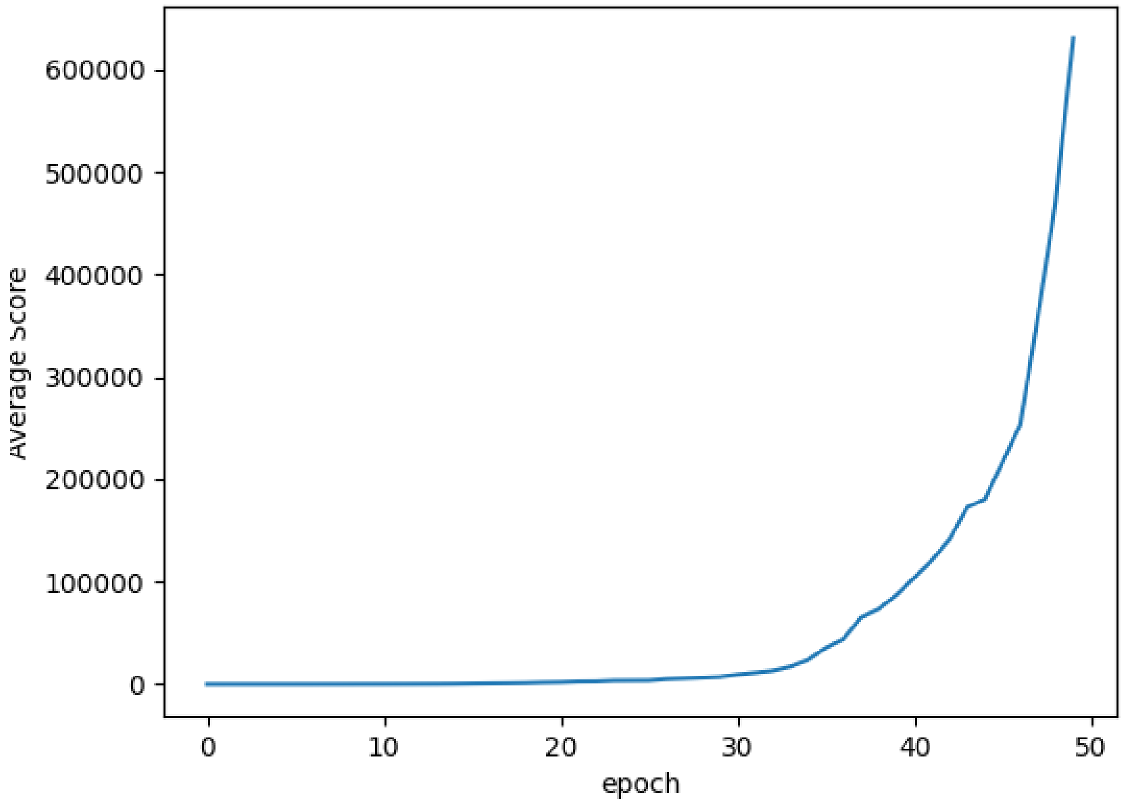

Figure 16: Average score of genetic population against number of generations (epochs) for CLVD target

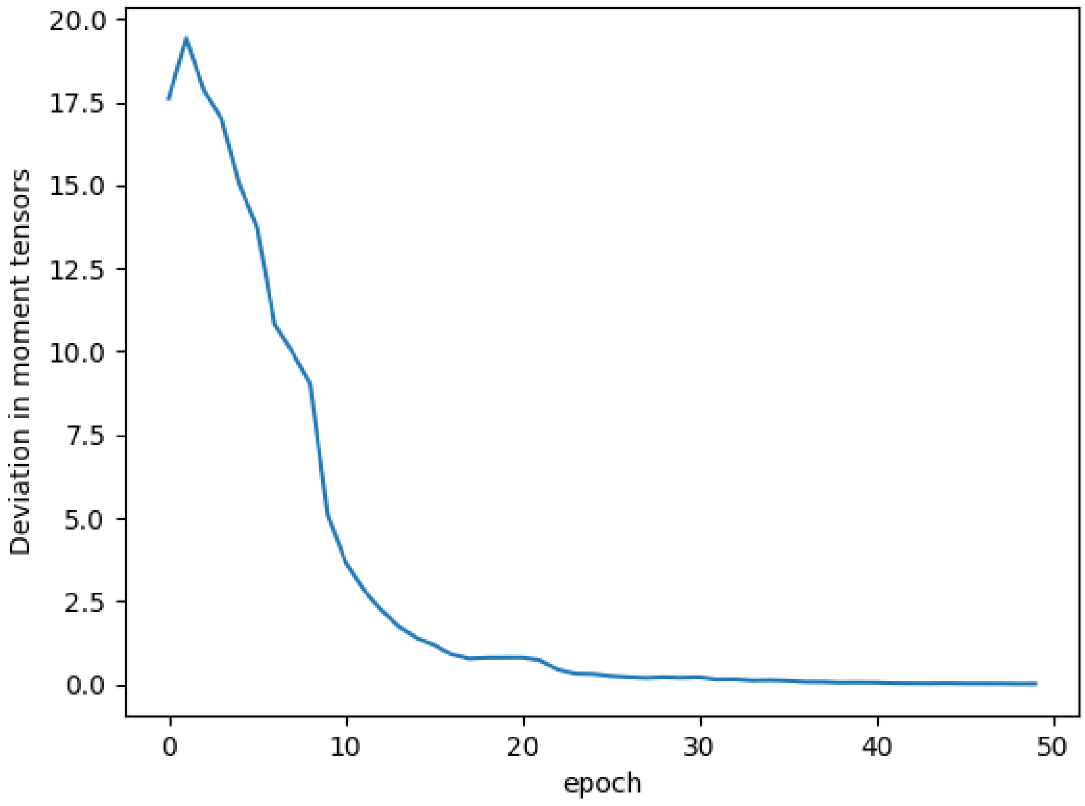

Figure 17: Deviation in moment tensor population against number of generations (epochs) for CLVD target

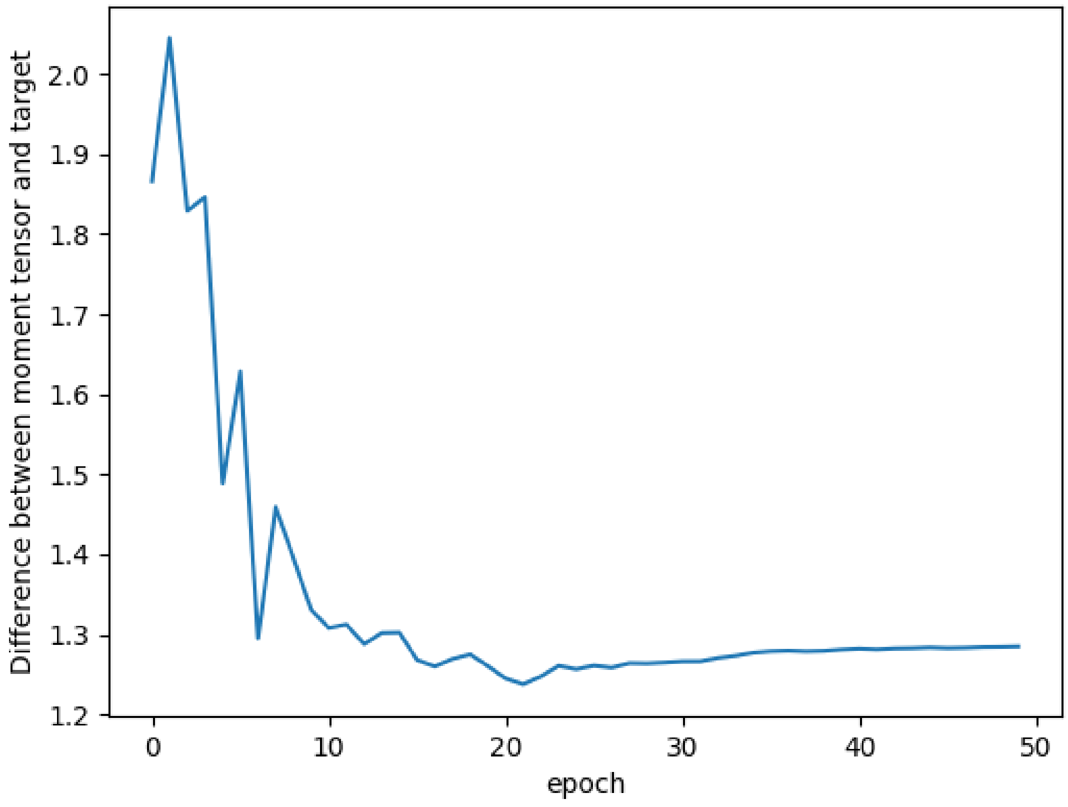

Figure 18: Similarity between average moment tensor and number of generations (epochs) for CLVD target

Summary of results

The three components conveiently show the three different outcomes of running the genetic algorithm. The first outcome shown in the DC case is the algorithm converges to a highly optimized moment tensor that is the target moment tensor. The second outcome shown in the isotropic case is that the genetic algorithm converges to a high optimized moment tensor that is not the target. Finally, as shown in the isotropic case, the algorithm can converge to a poorly optimized moment tensor that does not match the target.
To remedy the second and third outcomes the genetic population could be increased. This increases computation time. The second outcome may also be a feature of the greens functions not giving suffecient information for a unique inversion which would not benifit from an increased genetic population.
The program is quite slow, even at the current settings which use a small genetic population of 100 and 50 generations it takes about 10 minutes to run. The program could be sped up by using a faster language such as goLang or C.

Interesting Extensions

An additional term can be added to the metric $g(\cdot,\cdot)$ that makes solutions unique in the population get higher scores and therefore more likely to breed and spread their influence onto the next generation. By adjusting this term and clustering moment tensor solutions rather than returning their average, multiple solutions that fit the situation well could be found. This may allow for multiple interpretations of the event which can be woven together with other observations to provide a more coherent picture of the situation. As far as I understand standard techniques do not accommodate this.

Appendix

Definitions of metrics used

Score or score metric

The score metric is between two seismograms or sets of siemograms. For target seismogram d and another seismogram d', it is the sum $s(d, d') = \sum_i \sqrt{di**2}/\sqrt{((di - di')**2)}$. For a set of seismograms it is the sum over all $s(d, d')$ where d and d' are from the same station and are the same direction component.

Moment tensor deviation in a population

A population is composed of a set of moment tensors. For element with indexes i, j we find the maximum and minimum of $m_{i,j}$ where m is a moment tensor in the population. The difference between the maximum and minimum is the deviation for index i, j. The deviation is then the sum of deviations for each indexes i, j.