3Drelocate (Alpha)

-

Towards a Contributed Software Package for Earthquake Location Inversions Using 3D Velocity Models

Amir Allam – UAF

Malcolm White - UCSD

Overview

- Basic components
- Methodology
- Basic operation
- Test results
- Future development

Purpose

- The purpose of this software package is to leverage 3D velocity models in earthquake location inversions

Basic Components

- *generate_ttimes_fm3d*
- *3Drelocate*

- *loctools3d.core_tools*

- *fm3d – Nick Rawlinson (University of Aberdeen)*

Basic Components

- *generate_ttimes_fm3d*
  - Command line tool
  - An I/O wrapper around *fm3d*
  - Builds source-to-station travel-time lookup files, accounting for 3D seismic velocity structure
- *3Drelocate*
  - Command line tool
  - Interfaces Antelope database with inversion algorithm in *loctools3d.core_tools*
  - Can be replaced with interface to any data format

Basic Components

- *loctools3d.core_tools*
  - Python module
  - Implements location inversion algorithm
  - Implements internal data structures to allow for interface layer to be written for any data format
- *fm3d (N. Rawlinson)*
  - 3D wave-front tracking software
  - Accounts for 3D seismic velocity structure
  - Essential third-party component

Methodology

Input P-wave arrival time observations

↓

Brute force grid search

↓

Sub-grid inversion

↓

Output location

Methodology

- Want to find source location and time

$$\vec{u} = \langle x, y, z, t_0 \rangle$$

- Brute force grid search
  - Initial trial solution (location, time)
- Sub-grid inversion
  - Iterative small updates to trial solution

Methodology

- Input P-wave arrival time observations
  - *3Drelocate* iterates over events in an Antelope database and extracts event data to be passed to *loctools3d.core_tools* inversion algorithm
  - Extensible development model allows for inversion algorithm to be interfaced with arbitrary data formats

Methodology

- Brute force grid search

$$t_0^j \equiv \frac{1}{n}\sum_{i=1}^{n} t_{obs}^i - \tau_{pred}^{i,j}$$

where

t_0^j is the estimated origin time for a trial origin at the jth node

t_{obs}^i is the observed arrival time at the ith station

$\tau_{pred}^{i,j}$ is the predicted travel-time to the ith station from the jth node

Methodology

- Brute force grid search
  - Define residual at the jth node for the ith station

$$r^{i,j} \equiv t^{i}_{obs} - t^{i,j}_{pred}$$

$$r^{i,j} \equiv t^{i}_{obs} - (t^{j}_{0} + \tau^{i,j}_{pred})$$

  - Define the “misfit”

$$\sigma^{j} = \sum_{i=1}^{n} \left| r^{i,j} \right|$$

Methodology

- Brute force grid search
  - Grid node with smallest misfit is best fitting node

$$\min_j \left(\sigma^j \right) = j*$$

Best fitting grid node yields trial solution

$j*, t_0*$ or $\vec{u}* = \langle x*, y*, z*, t_0* \rangle$

Methodology

- Sub-grid inversion
  - estimate origin time based on best-fitting node

$$t_0 * = \frac{1}{n} \sum_{i=1}^{n} \left(t_{obs}^{i} - \tau_{pred}^{i,j*} \right)$$

  - define residual of the ith observation

$$r^i \equiv t_{obs}^{i} - \left(t_0 * + \tau_{pred}^{i,j*} \right)$$

Methodology

- Sub-grid inversion
  - Approximate the gradient of the travel-time field, for the ith station, at the position of the best fitting grid node, $\langle x^*, y^*, z^* \rangle$

$$\nabla \tau_i = \left\langle \frac{\partial \tau_i}{\partial x}, \frac{\partial \tau_i}{\partial y}, \frac{\partial \tau_i}{\partial z} \right\rangle$$

$$\frac{\partial \tau_i}{\partial x} \approx \frac{1}{2}\left[\left(\tau_{x^*,y^*,z^*} - \tau_{x^*-1,y^*,z^*}\right) + \left(\tau_{x+1^*,y^*,z^*} - \tau_{x^*,y^*,z^*}\right)\right]$$

$$\frac{\partial \tau_i}{\partial y} \approx \ldots$$

$$\frac{\partial \tau_i}{\partial z} \approx \ldots$$

Methodology

- Sub-grid inversion
  - A small change in the origin location results in a change in the travel-time given by

$$\Delta\tau^i = \nabla\tau^i \cdot \Delta\vec{x}$$

where $\Delta\vec{x}$ is the displacement vector

  - The change in arrival time at the ith station based on an updated solution can be written

$$\Delta t^i_{pred} = \nabla\tau^i \cdot \Delta\vec{x} + \Delta t_0$$

$$\Delta t^i_{pred} = \left\langle \nabla\tau^i, 1 \right\rangle \cdot \Delta\vec{u}$$

Methodology

- Sub-grid inversion
  - Requiring

$$\Delta t^{i}_{pred} = -r^{i}$$

  - Yields residuals for updated solution given by

$$r'^{i} = t^{i}_{obs} - (t^{i}_{pred} + \Delta t^{i}_{pred})$$

$$r'^{i} = t^{i}_{obs} - (t^{i}_{pred} - r^{i})$$

$$r'^{i} = r^{i} - r^{i}$$

$$r'^{i} = 0$$

Methodology

- Sub-grid inversion
  - set up the linear system of i equations in 4 unknowns

$$\left\langle \nabla \tau^i, 1 \right\rangle \cdot \Delta \vec{u} = -r^i$$

  - take $\Delta \vec{u}$ to be the least-squares solution to this set of equations.

Methodology

- Sub-grid inversion
  - The updated solution is

$$\vec{u}^* \rightarrow \vec{u}^* + \Delta\vec{u}$$

  - Iterate until change in location falls below threshold

Basic Operation

I) Build travel-time lookup files

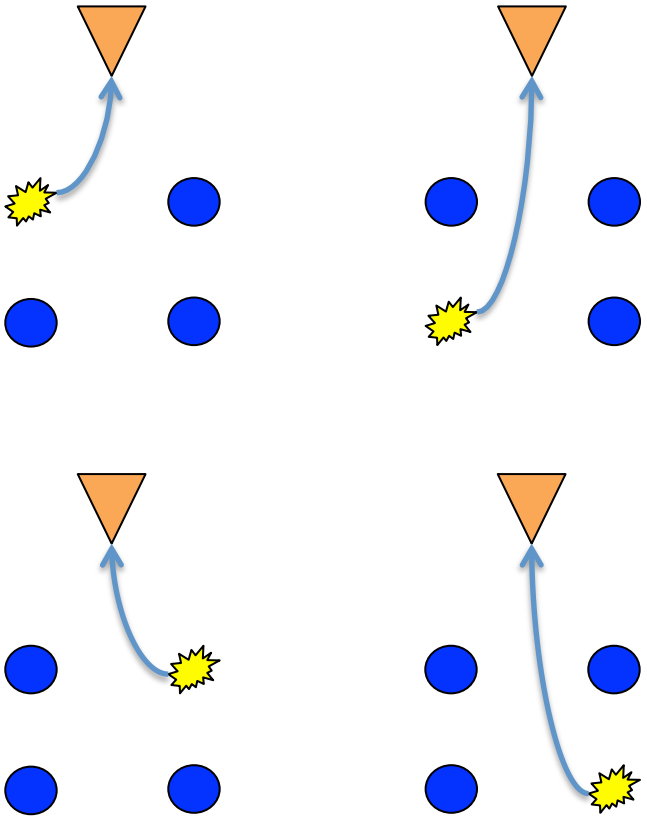
generate_ttimes_fm3d

II) Invert for origin using static travel-time files

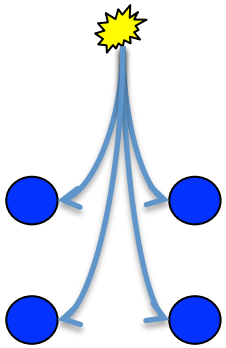
3Drelocate

Reciprocity Principle

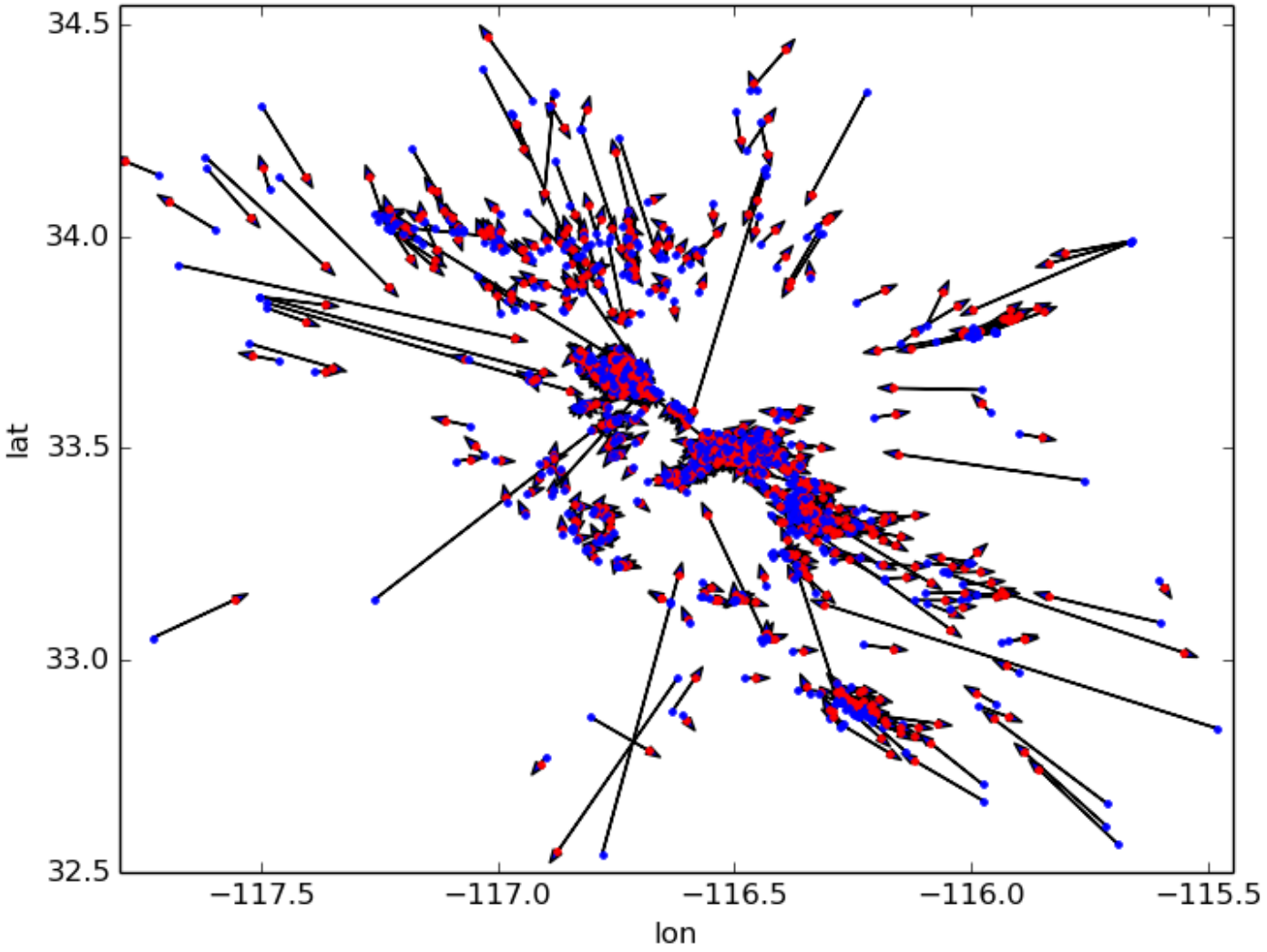
O(n) executions



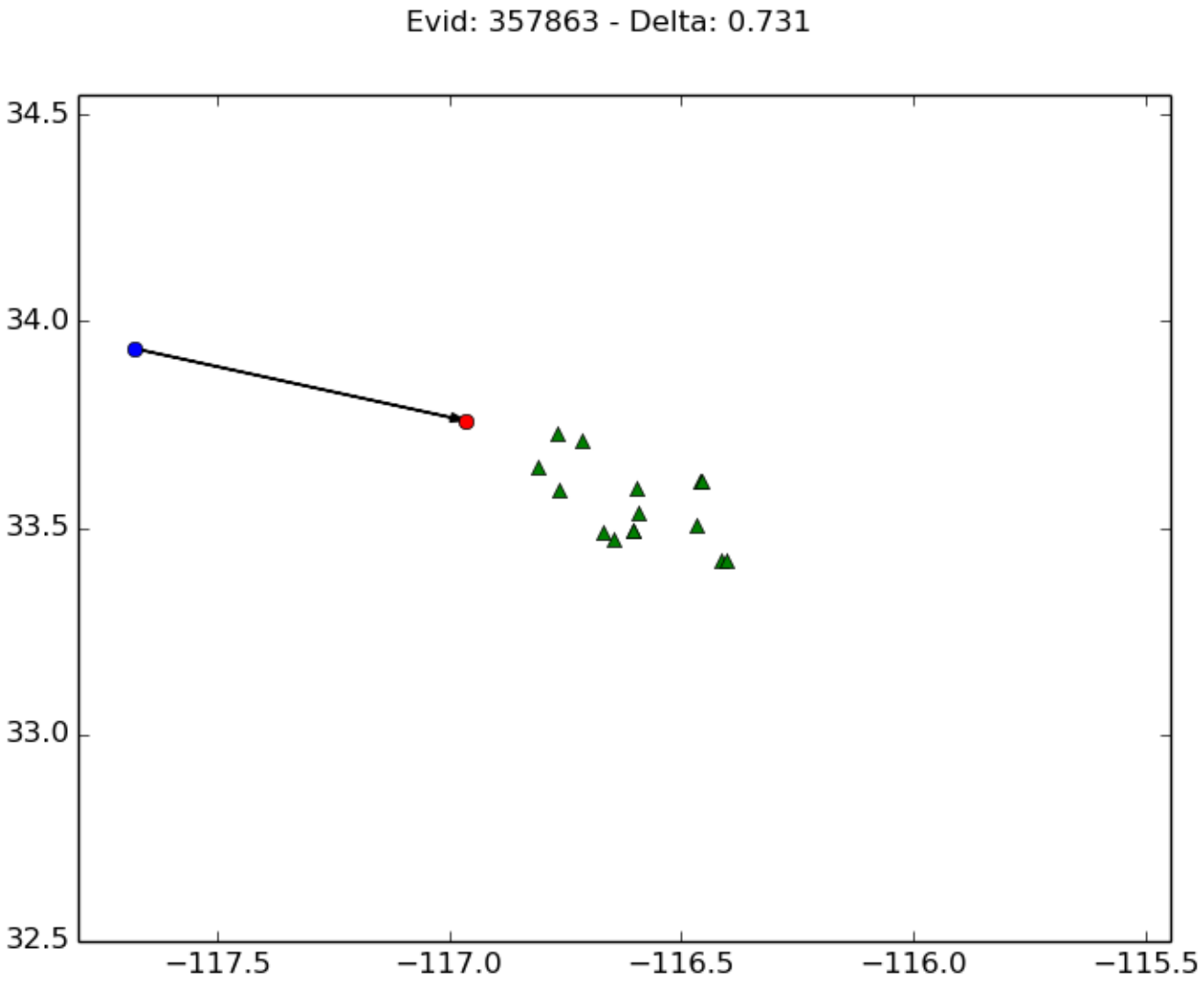
O(1) executions



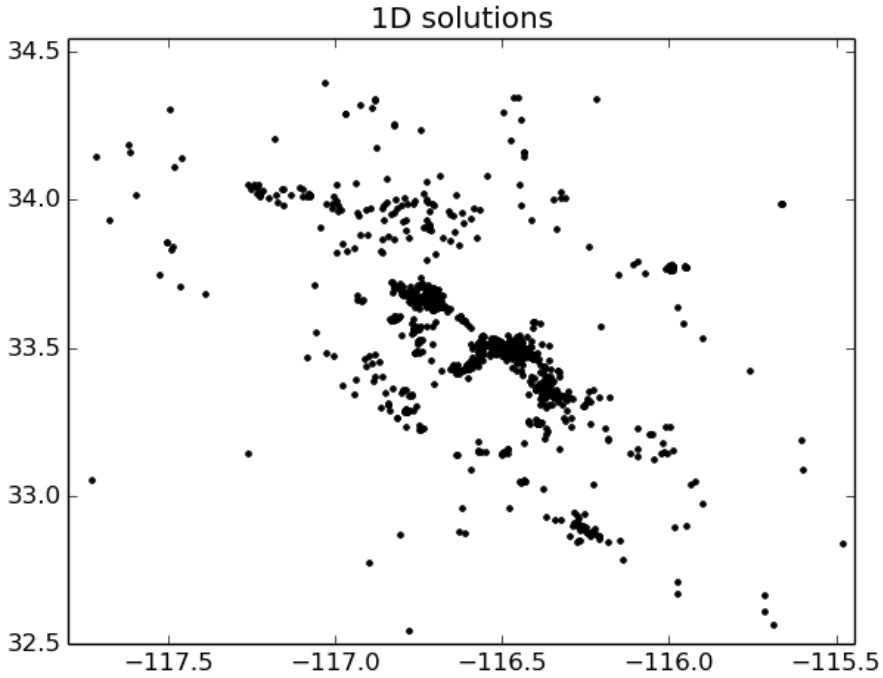
Test Results

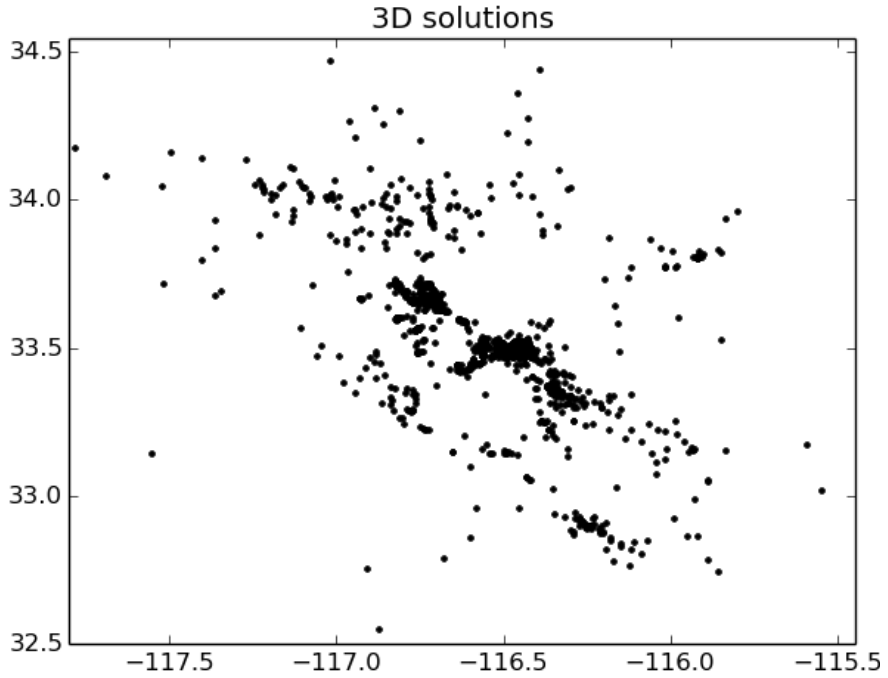


Test Results

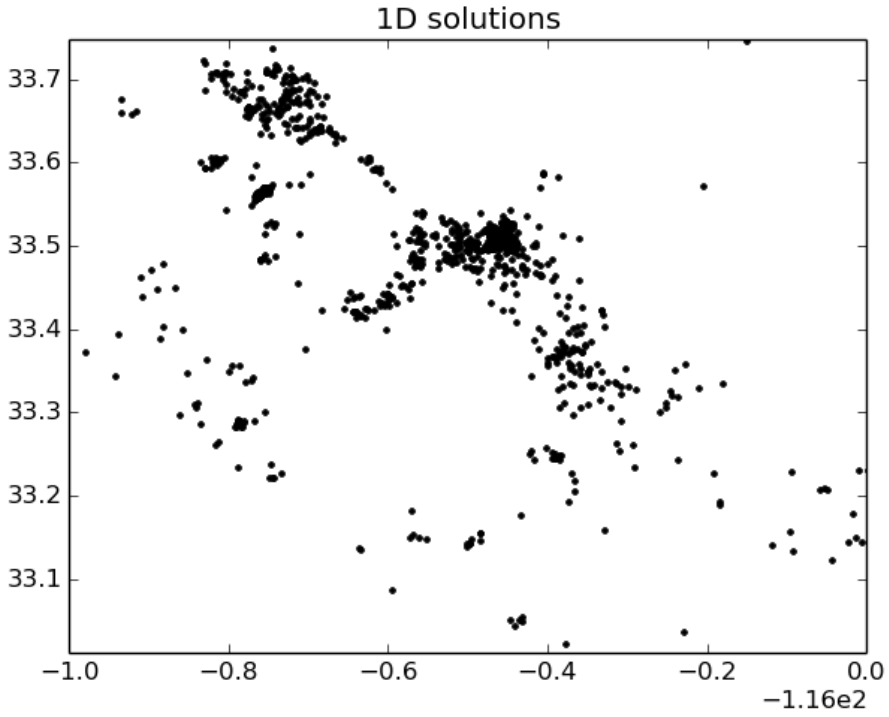


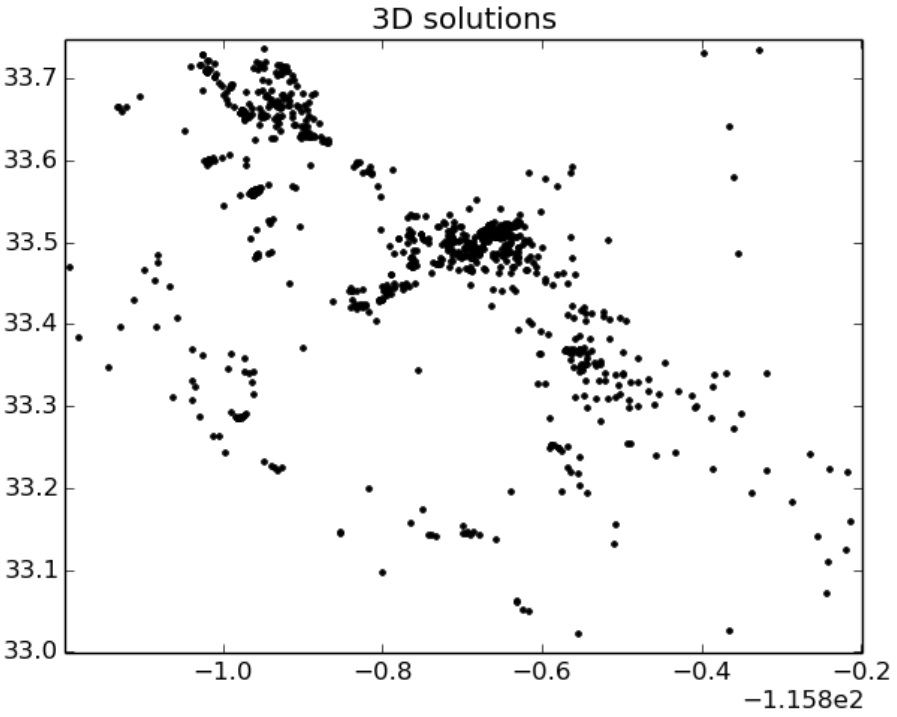
Test Results





Test Results





Future Work / Improvements

- Account for S-wave observations in inversion
- Optimization
  - Memory usage
    - Memory mapping of travel-time lookup files?
    - Travel-times are currently being loaded into memory for speed
  - CPU cycles