simpeg

Using simpegMT

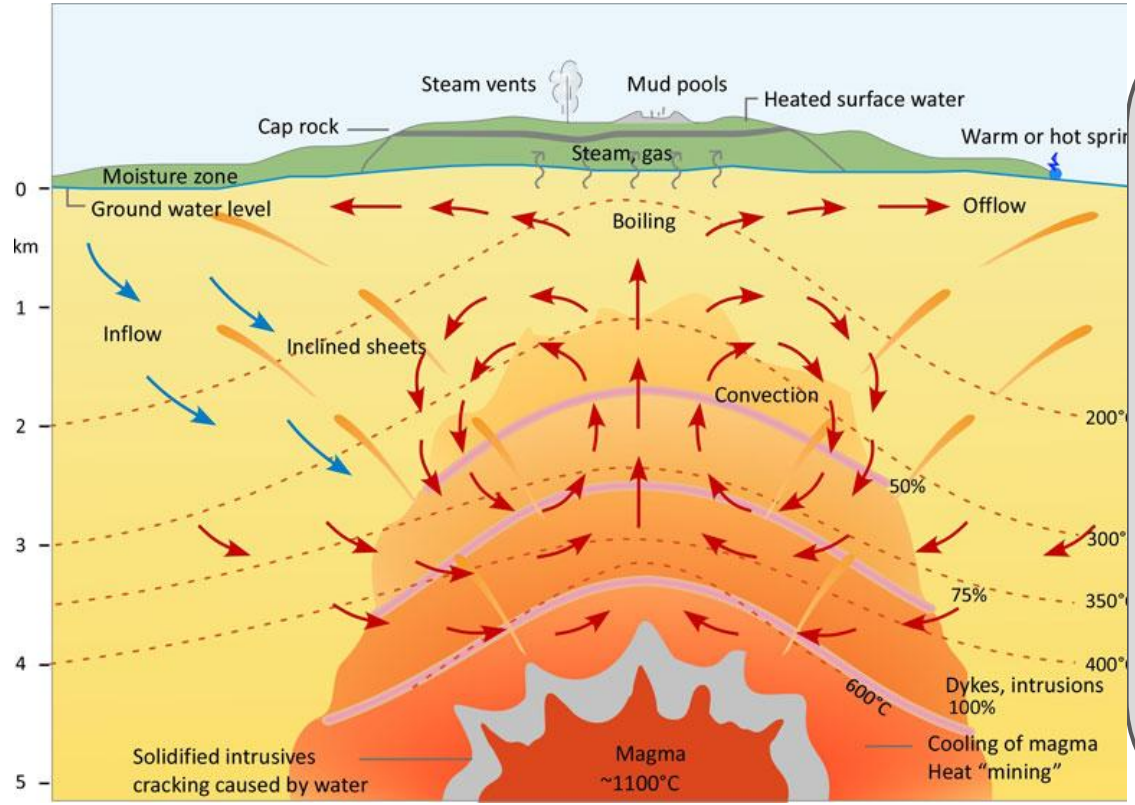
To demonstrate a modular modelling and inversion framework applied in a workflow for a hydrothermal system

Guðni Rosenkjær, Lindsey Heagy, Rowan Cockett & Seogi Kang

GIF

Geophysical Inversion Facility
University of British Columbia

Motivation:



http://www.unugtp.is/en/organization/what-is-

Geothermal systems are
- Large
- Open

They have complex
- geological structures
- tectonics
- physical properties

It is important to assess the resource with geophysics

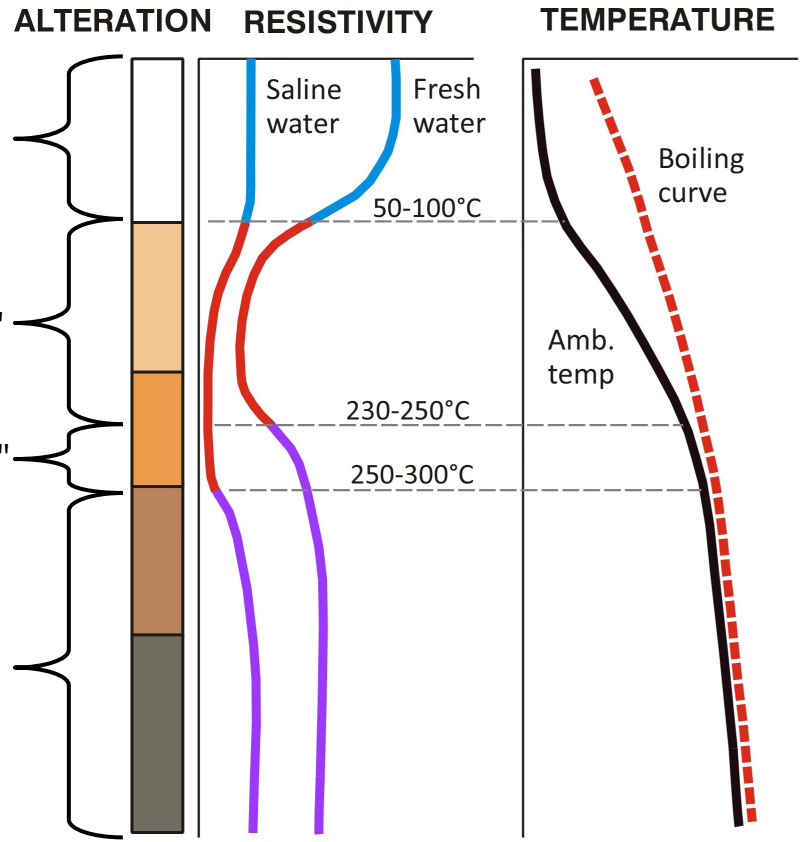
Resistivity in Geothermal systems

Unaltered
< 50 °C - "High resistivity"

Smectitie-zeolite
50 − 230 °C - "Decreasing resistivity"

Mixed clay
230 − 250 °C - "Increasing resistivity"

Clorite-epidote
> 250 °C " - Higher resistivity"



Pore fluid conduction

Surface conduction in clay minerals

Surface and pore fluid conduction

Rel. unaltered

Smectite- zeolite zone

Mixed layer clay zone

Chlorite zone

Chlorite-epidote zone

Magneto Tellurics

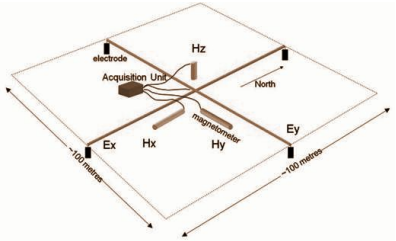
Natural Fields

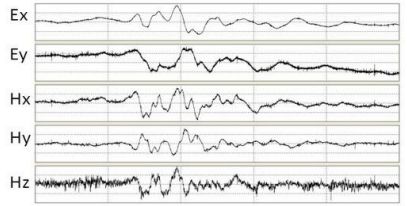
input energy

measure response

Subsurface:
Electrical Conductivity

Impedance Data

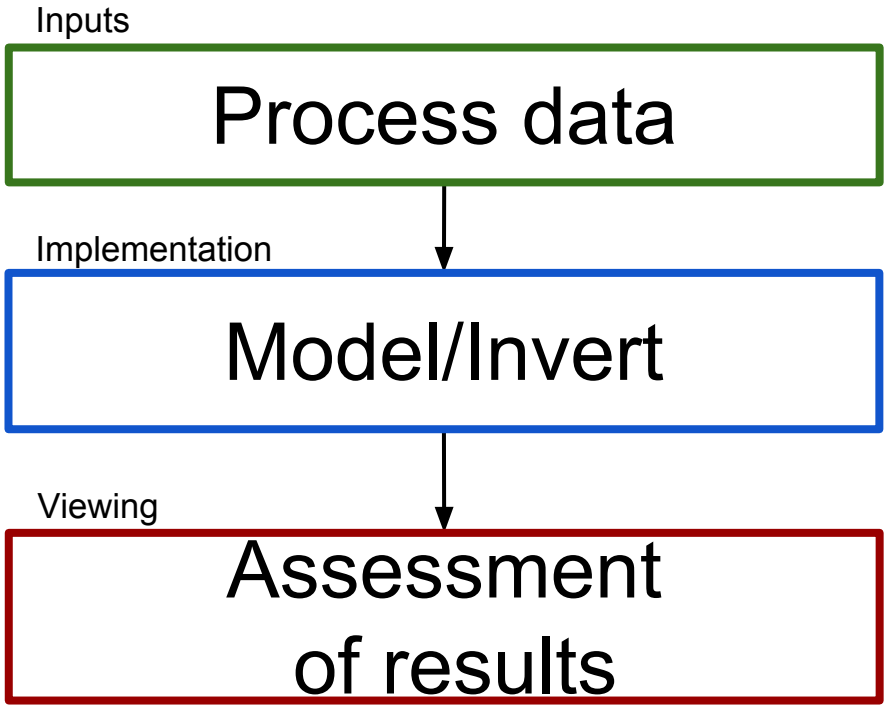




Process

$$\mathbf{Z}(\omega) = \begin{bmatrix} Z_{ii}(\omega) & Z_{ij}(\omega) \\ Z_{ji}(\omega) & Z_{jj}(\omega) \end{bmatrix} = \begin{bmatrix} \frac{E_i(\omega)}{H_i(\omega)} & \frac{E_i(\omega)}{H_j(\omega)} \\ \frac{E_j(\omega)}{H_i(\omega)} & \frac{E_j(\omega)}{H_j(\omega)} \end{bmatrix}$$

Work approach with simpegMT



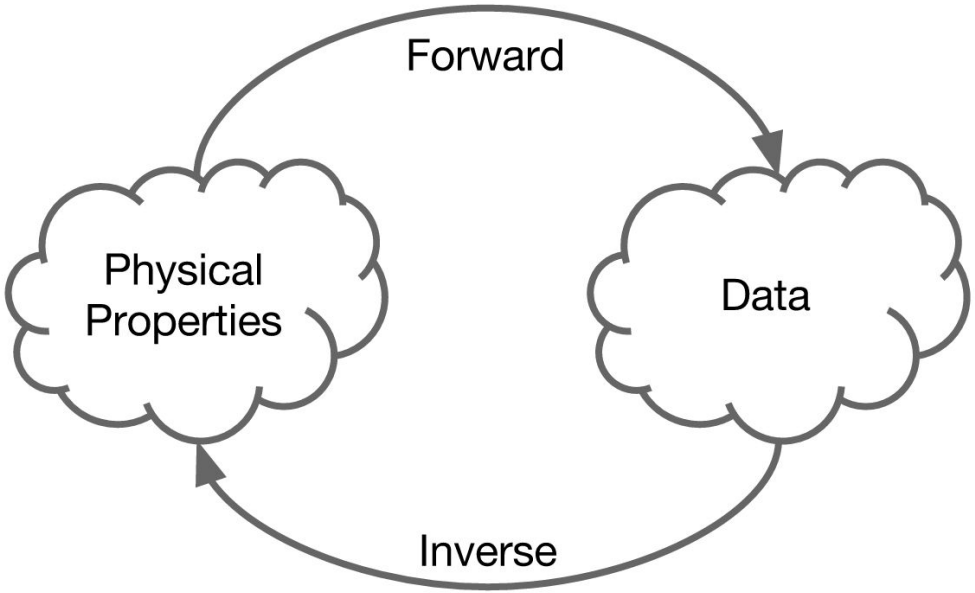
In order to be able invert the data we need to

- Process the data
  - Assign uncertainties
- Setup the inversion
  - Define the physics
- Assess the results
  - View the models

Prefer having a single platform to do all these tasks.

Inversion

Want to recover conductivity model based on our MT observation



Implementation

Implemented in Python!

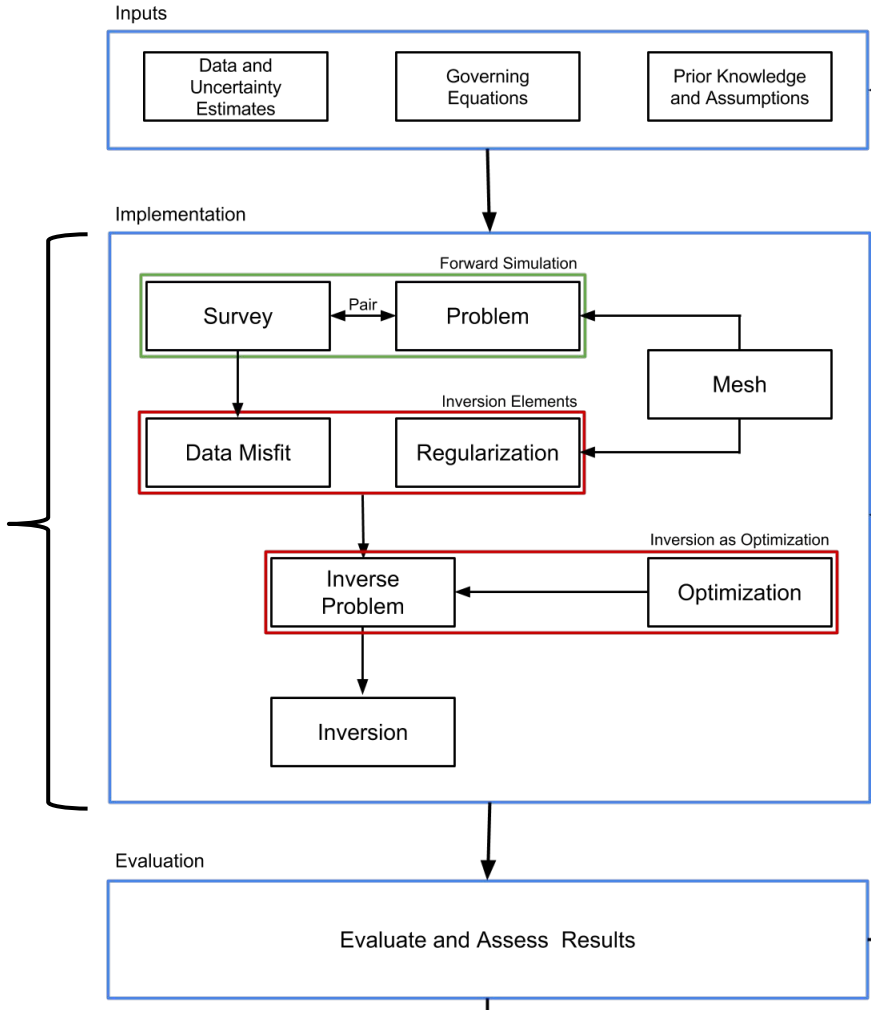
```
In [1]: import SimPEG

In [ ]: SimPEG.
```

SimPEG.DataMisfit
SimPEG.Directives
SimPEG.Fields
SimPEG.InvProblem
SimPEG.Inversion
SimPEG.Maps
SimPEG.Mesh
SimPEG.Models
SimPEG.Optimization
SimPEG.Problem

jupyter

simpeg

http://simpeg.xyz

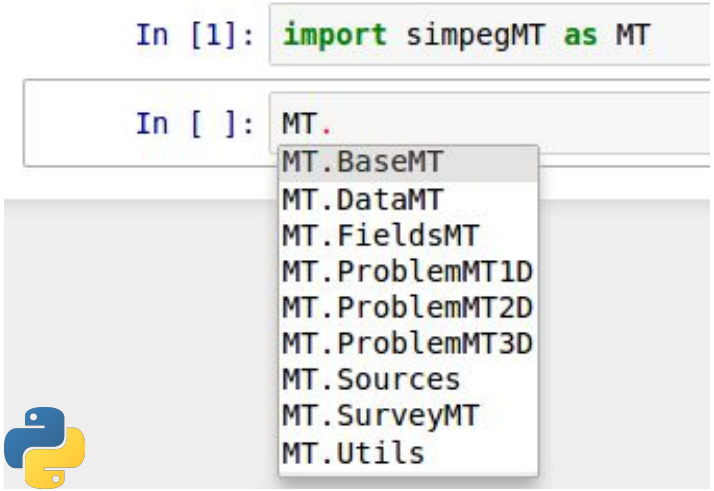


simpegMT

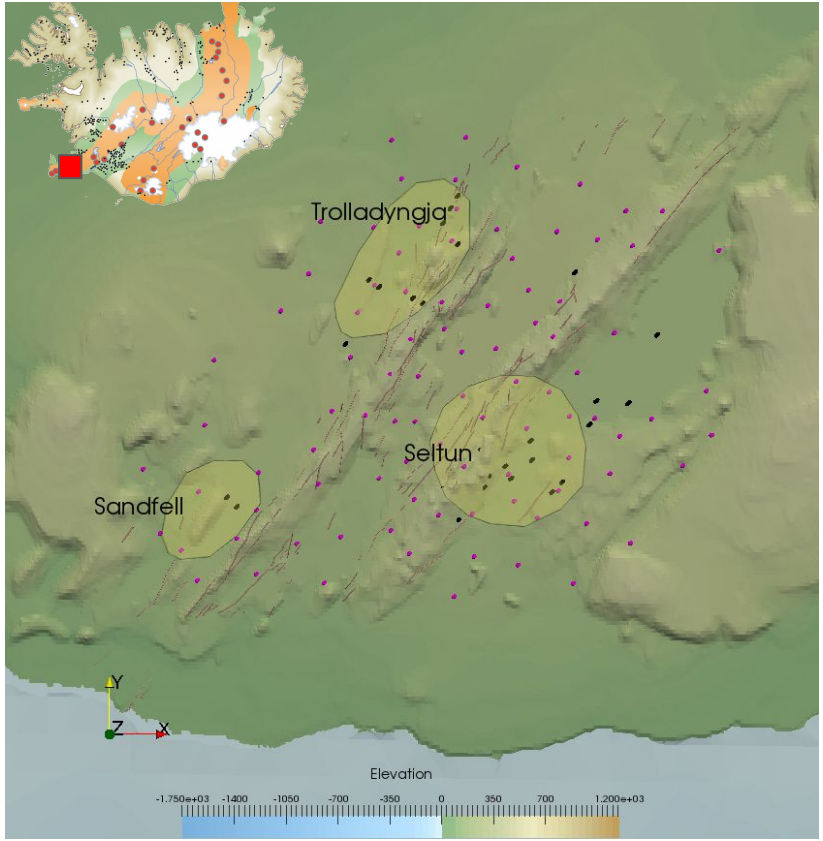
Built on the SimPEG framework
- Fully compatible elements
  - Meshing
  - Inversion
  - etc...

Interoperable with other problems built on the SimPEG framework

simpegMT implements the MT forward problem, data projections and method specifics



Krysuvik geothermal area



The Krysuvík geothermal area is 30km South of Reykjavik.

On Reykjanes peninsula SW-NE directed fissures and faults (brown lines)

100 MT data stations (purple dots)

3 main hydrothermal area's (yellow polygons)

1D inversion

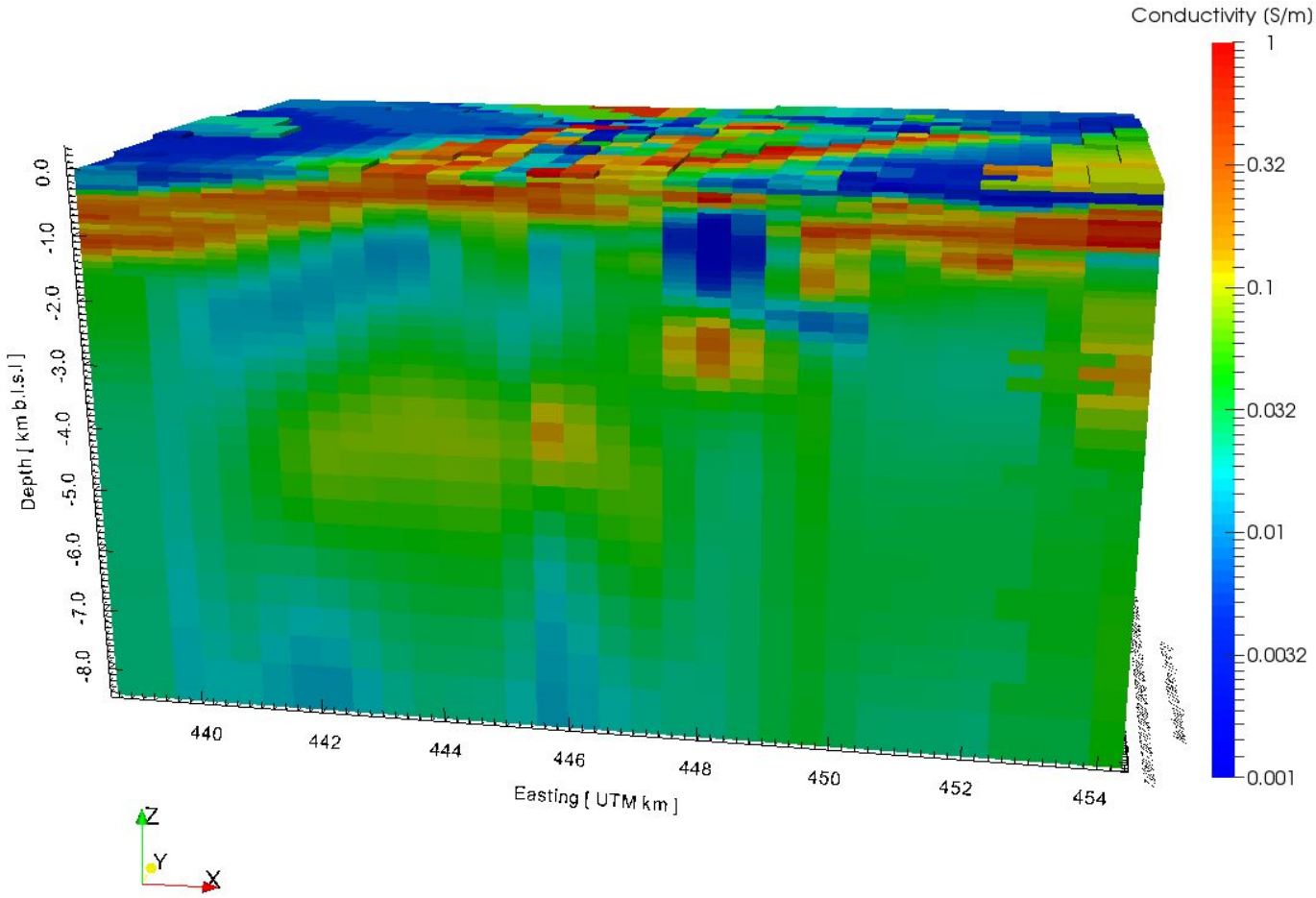
Motivation:
Find a layer model that fits data at each measurement location

- Fast preliminary estimate
- One component at a time Z_{xy} , Z_{yx} or $Z_{det} = \sqrt{Z_{xx}Z_{yy} - Z_{xy}Z_{yx}}$

In [1]:

```
import SimPEG as simpeg
import simpegMT as simpegmt

# Load EDI data
MTdata = simpegmt.Utils.EDIimporter(ediFileList)

# Make a mesh and problem
mesh1D = simpeg.Mesh.TensorMesh([LayerThickness])
problem = simpegmt.ProblemMT1D.eForm_psField(m1d,sigmaPrimary = sigma1d)

# Run for all the locations
for dat in simpegmt.Utils.dataUtils.convert3Dto1Dobject(KryDobs,'zdet'):
    runInversion(dat,problem,m1d,'zdet')
```

1D inversion results



3D inversion

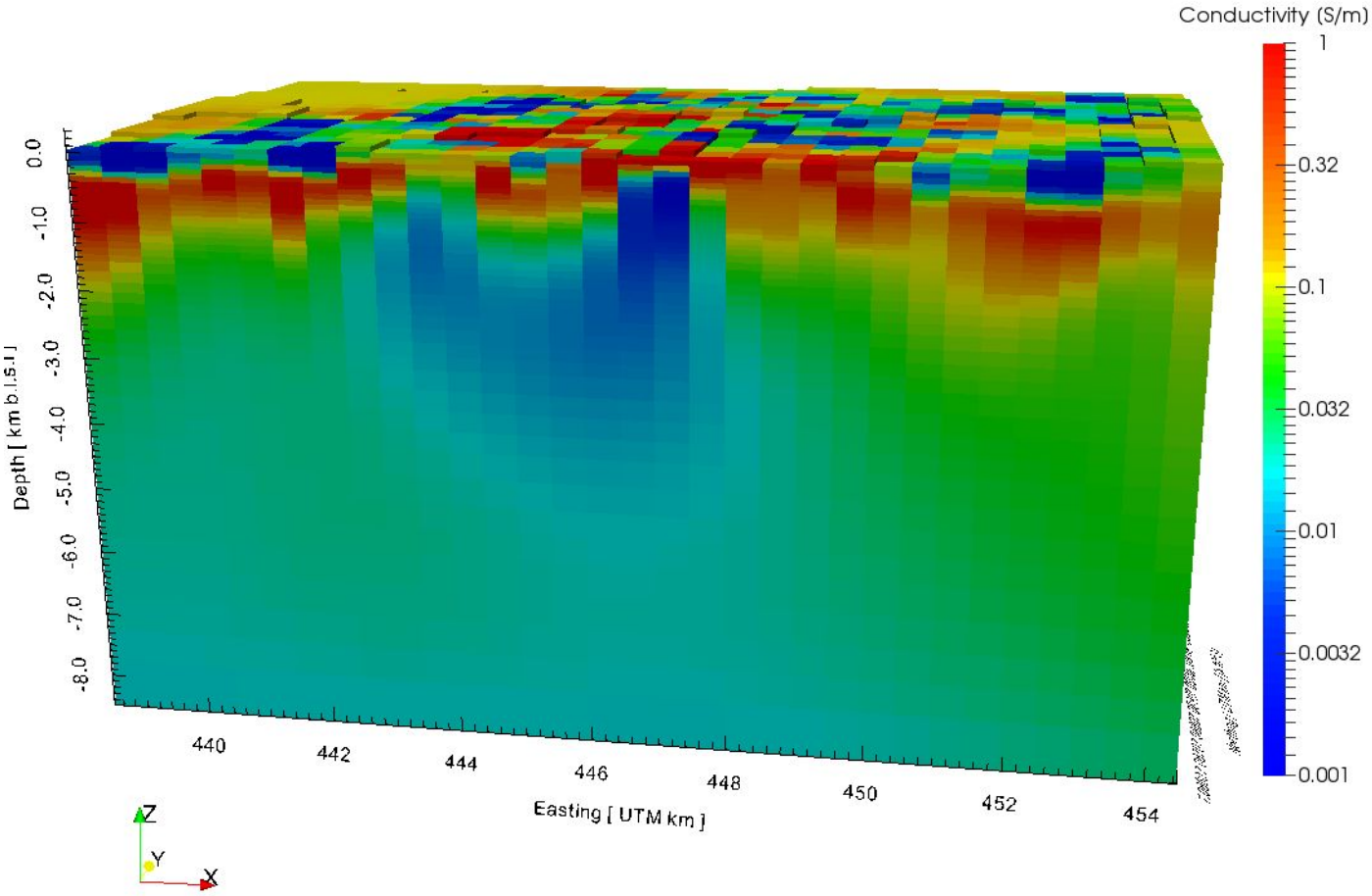
Motivation:
Find a 3D model that explain the measurements

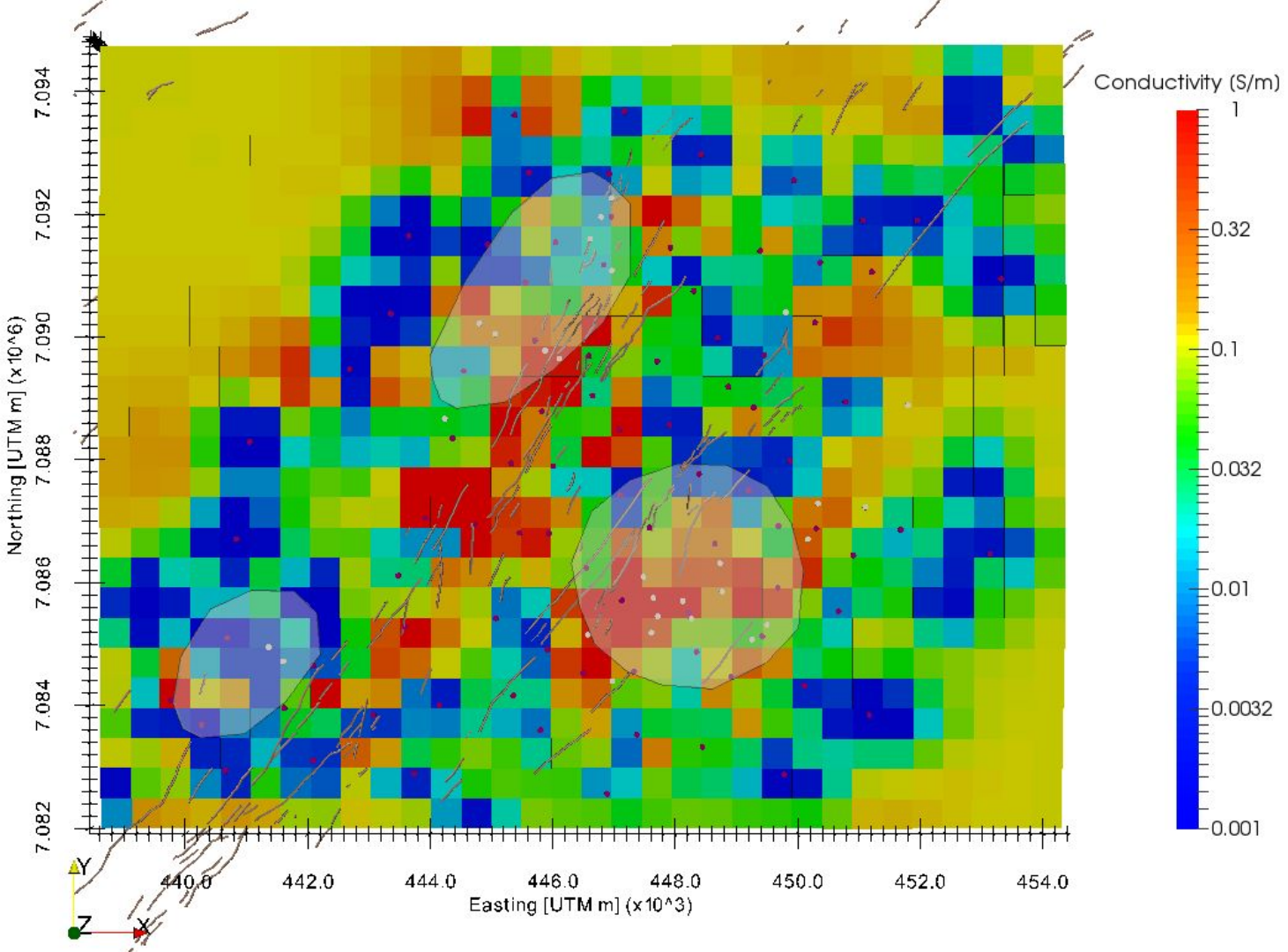
- Explain geological structures in the area
- Use off-diagonal components only Z_{xy} and Z_{yx}

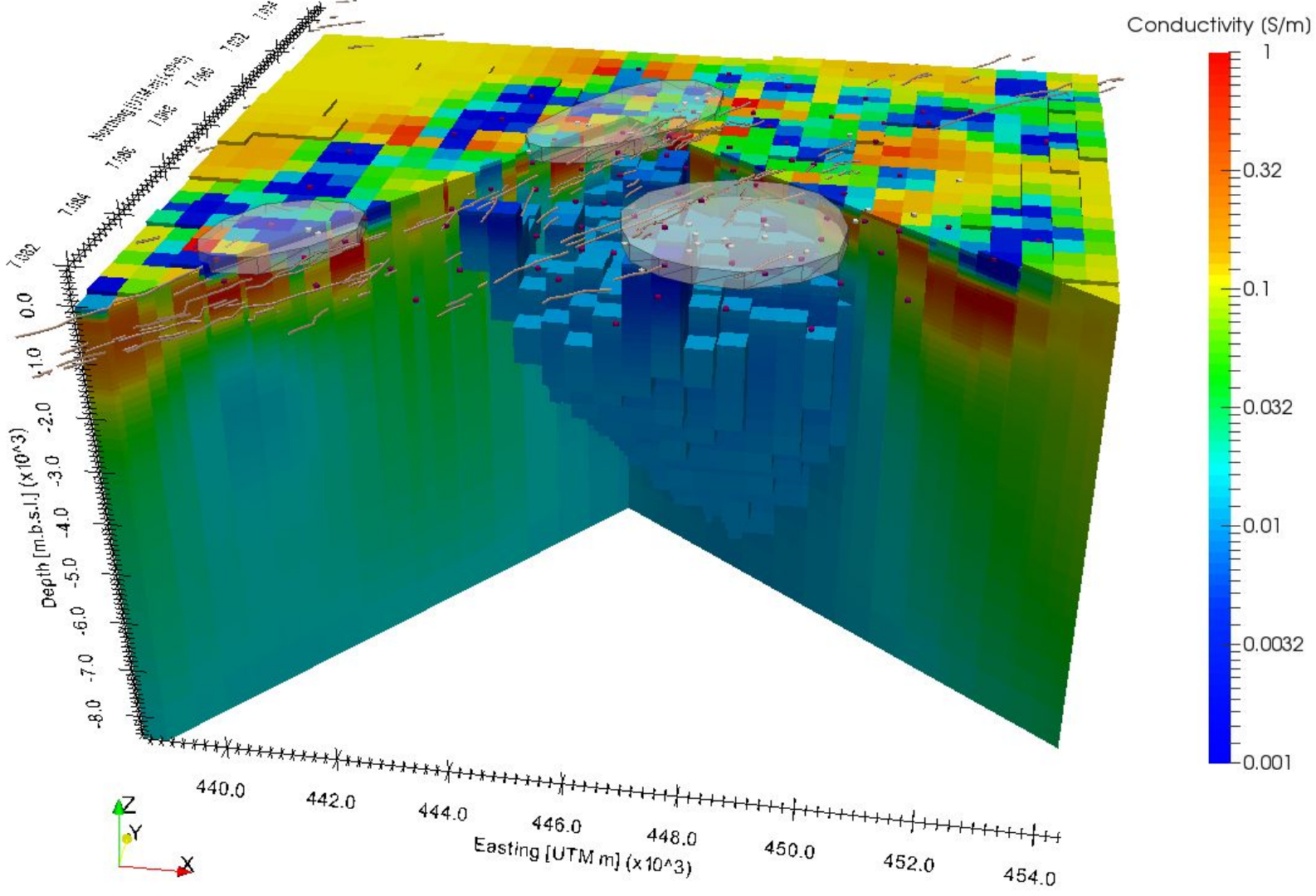
In [1]:

```
import SimPEG as simpeg
import simpegMT as simpegmt

# Load EDI data
MTdata = simpegmt.Utils.EDIimporter(ediFileList)

# Make the mesh
mesh = simpeg.Mesh.TensorMesh([cellSizeTensor],origin)

## Setup the problem object
problem = simpegmt.ProblemMT3D.eForm_ps(mesh,sigmaPrimary)

# Set the optimization
opt = simpeg.Optimization.InexactGaussNewton(maxIter = 50)

# Data misfit
dmis = simpeg.DataMisfit.l2_DataMisfit(survey)

# Regularization
reg = simpeg.Regularization.Tikhonov(mesh)


# Inversion problem
invProb = simpeg.InvProblem.BaseInvProblem(dmis, reg, opt)

# Create an inversion object
inv = simpeg.Inversion.BaseInversion(invProb, directiveList)

# Run the inversion
mopt = inv.run(m_0)
```

3D inversion results




Conductivity (S/m)
1
0.32
0.1
0.032
0.01
0.0032
0.001
Northing [UTM m] (x10^6)
7.094
7.092
7.090
7.088
7.086
7.084
7.082
Easting [UTM m] (x10^3)
440.0
442.0
444.0
446.0
448.0
450.0
452.0
454.0
Y
Z
X


Conductivity (S/m)
1
0.32
0.1
0.032
0.01
0.0032
0.001
Depth [m.b.s.l.] (x10^3)
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
-6.0
-7.0
-8.0
440.0
442.0
444.0
446.0
448.0
450.0
452.0
454.0
Easting [UTM m] (x10^3)
Z
Y
X

Summary of talk

- Motivated by geothermal problems
- Overview of the simpegMT code
- Example
  - Krysuvik geothermal area
    - 1D determinant inversion
    - 3D off-diagonal element inversion

Further work

Use of simpegMT:

- Continue work in Krysuvik
- Apply to other area's
- Etc..

Get others to use simpegMT

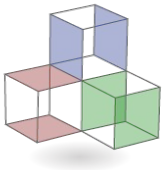
Development of simpegMT:

- Release stable version early 2016 within SimPEG
- Add functionality
  - Tipper inversions
  - ZTEM

Leverage new developments in SimPEG

Thank You!

Questions?



simpeg.xyz

github.com/simpeg

grosenkj@eos.ubc.ca

Thanks to Iceland GeoSurvey and HS Orka