simpeg

SimPEG

Development practices and lessons learned

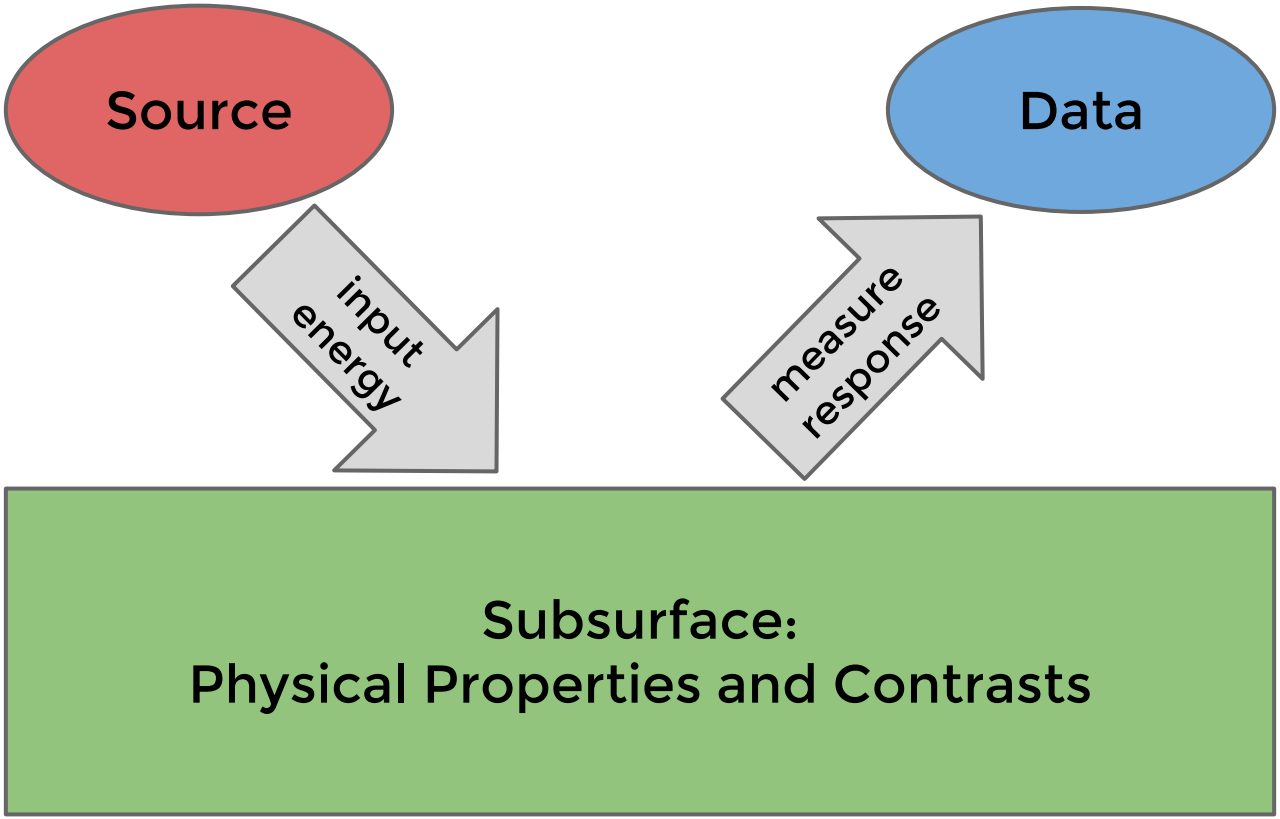
Rowan Cockett, Lindsey Heagy, Seogi Kang, & Guðni Rosenkjær

Geophysical Inversion Facility
University of British Columbia

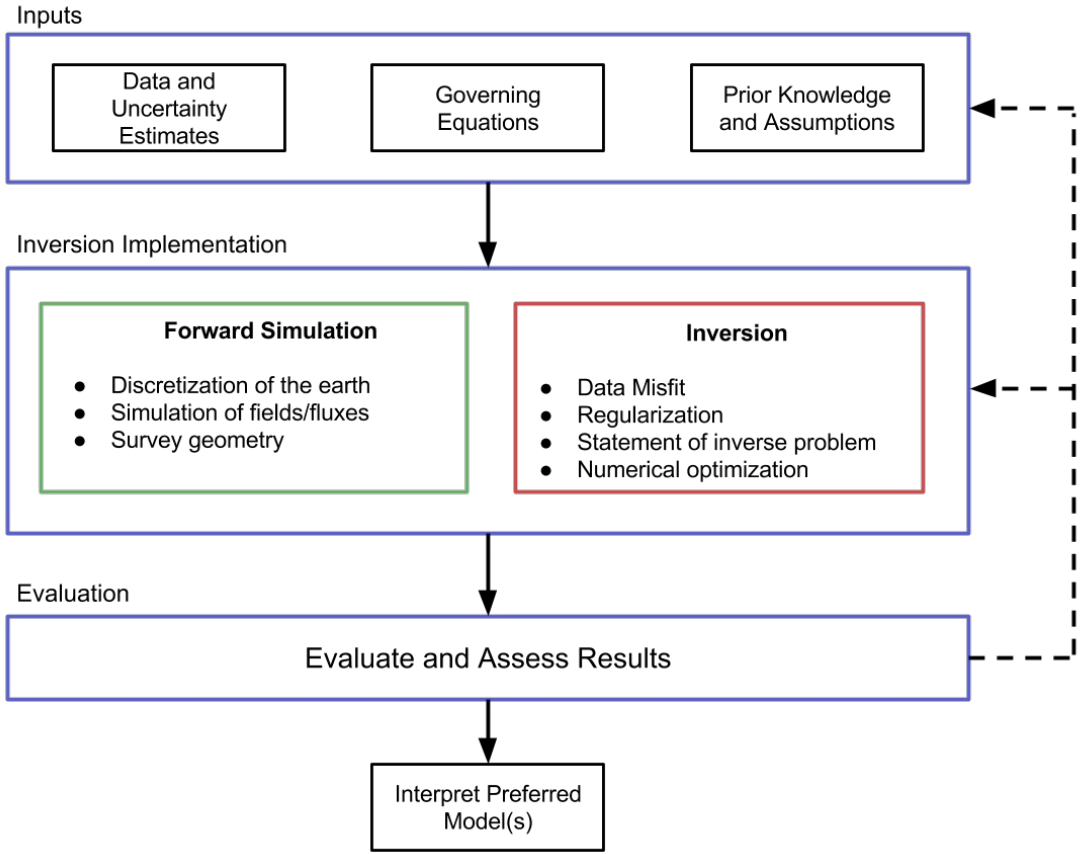
SimPEG

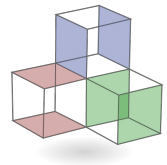
Simulation and Parameter Estimation in Geophysics

Geophysics!

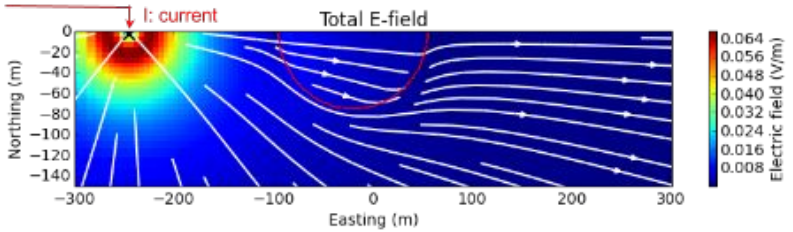

Source
Data
input energy
measure response
Subsurface:
Physical Properties and Contrasts

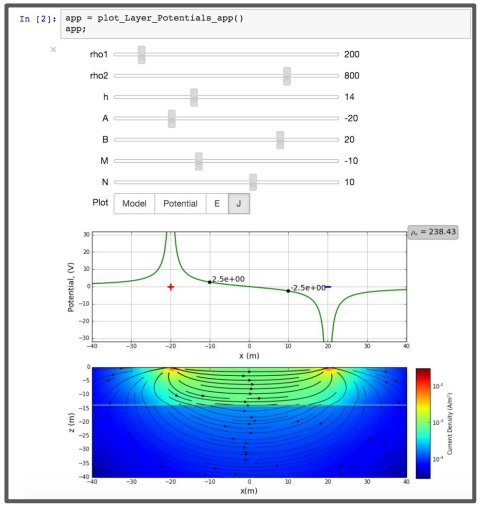
Inversion Pieces

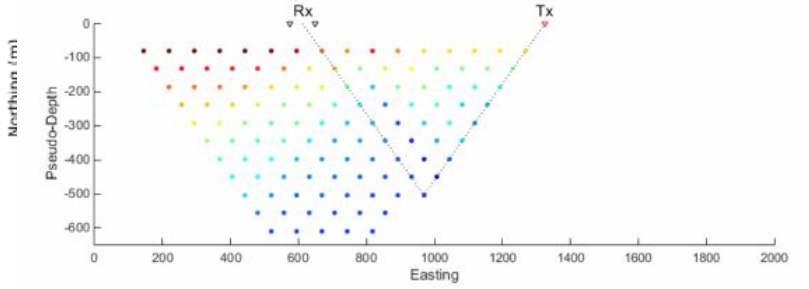


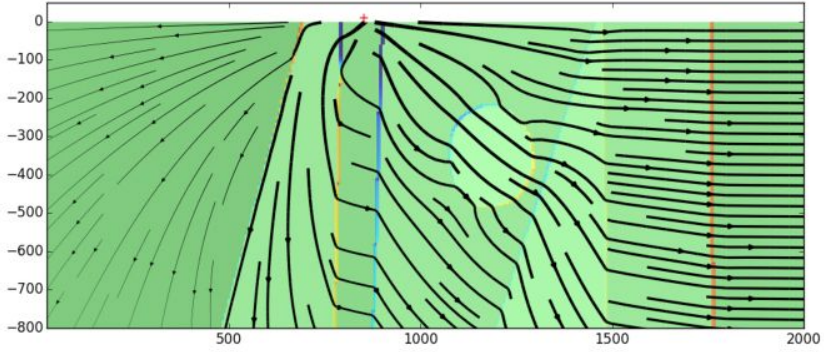


DC Resistivity









em.geosci.xyz

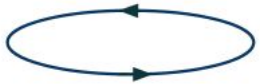
Electromagnetics

Time/Frequency
1D, (2D), 3D
Primary/Secondary
Mu/Sigma inversions
EB, HJ formulations

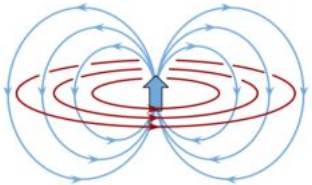
$$\nabla \times \mathbf{e} + \frac{\partial \mathbf{b}}{\partial t} = \mathbf{s_m}$$
$$\nabla \times \mathbf{h} - \mathbf{j} = \mathbf{s_e}$$

$$\nabla \times \mathbf{E} + i\omega\mathbf{B} = \mathbf{S_m}$$
$$\nabla \times \mathbf{H} - \mathbf{J} = \mathbf{S_e}$$

grounded electric



inductive loop

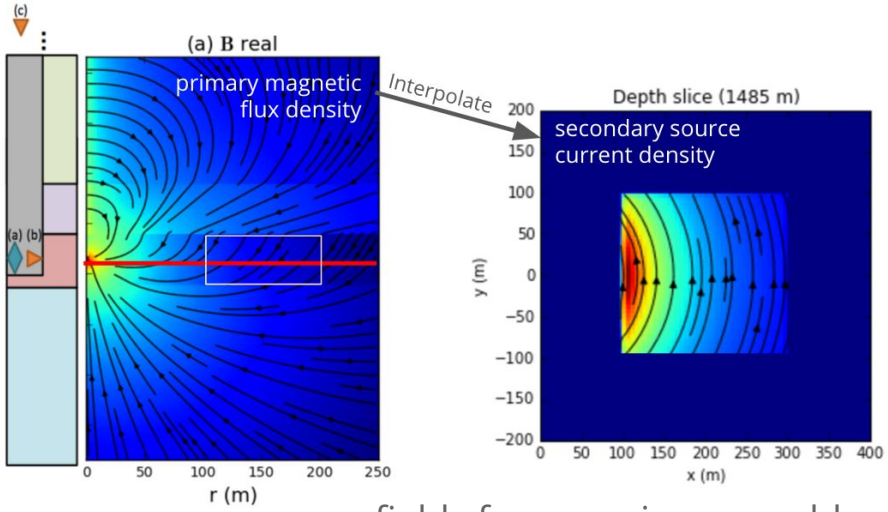


point dipole
(electric or magnetic)

natural source

Data:

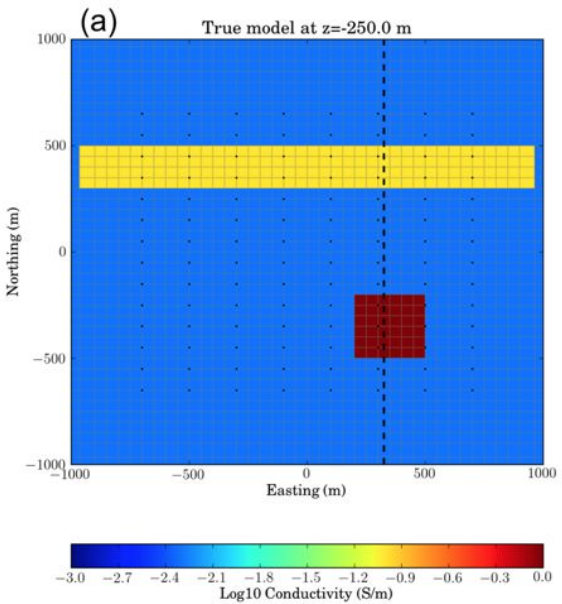
$\mathbf{e_x}^{real}$
$\mathbf{e_x}^{imag}$
$\mathbf{e_y}^{real}$
$\mathbf{e_y}^{imag}$
$\mathbf{b_z}^{real}$
$\mathbf{b_z}^{imag}$
...

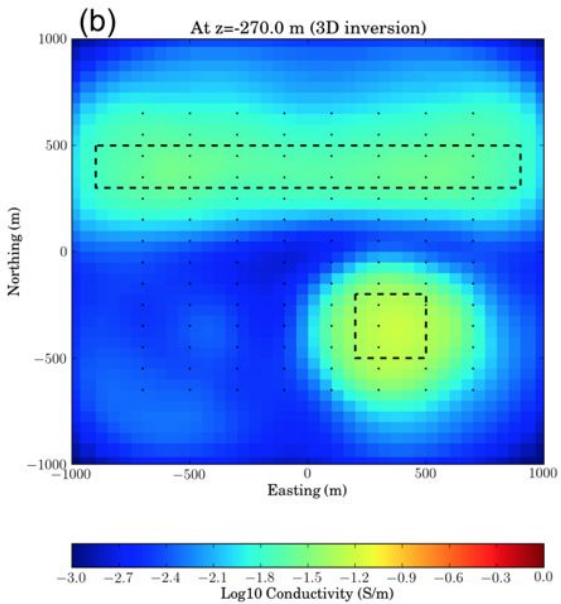


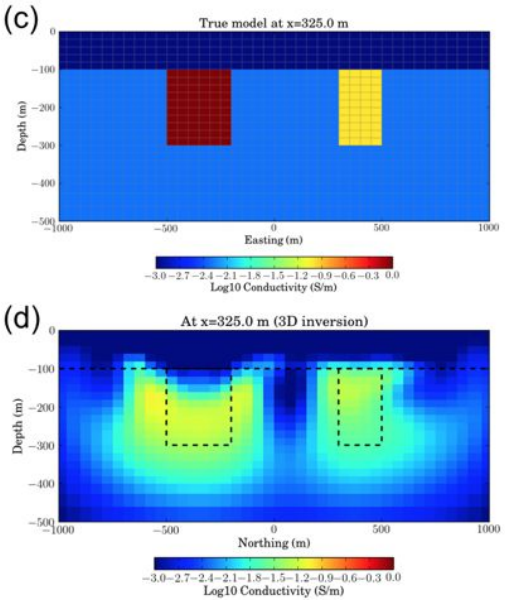
fields from a primary problem

```
In [1]: import SimPEG.EM as EM

In [ ]: EM.FDEM.Src.PrimSecSigma.
        EM.FDEM.Src.PrimSecSigma.eval
        EM.FDEM.Src.PrimSecSigma.evalDeriv
```

Airborne Time Domain EM Inversion







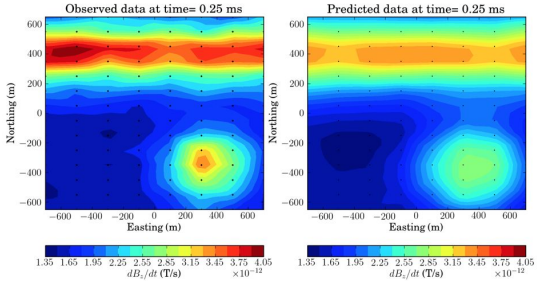
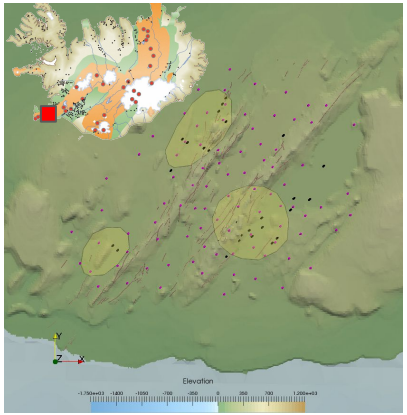


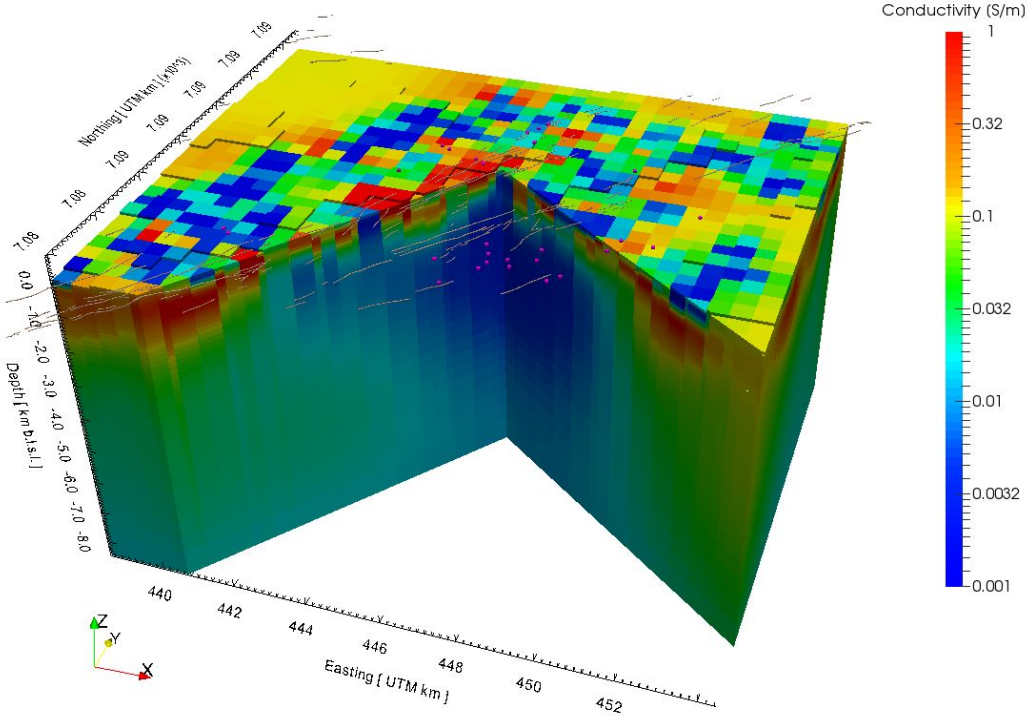
Figure above shows plan views of (a) the true conductivity model and (b) the recovered model, and section views of (c) the true and (d) the recovered conductivity models. Figure to the left shows observed and predicted data. Core cell size: 50×50×20 m, The number of cell: 50×50×48 = 120,000; Reference model: Half-space model with conductivity value, 0.005 S/m; Inexact Gauss-Newton: 13 iterations; Cpu time: 48hrs; Maximum memory usage: 51.2GB; Cpu: Intel(R)Xeon(R) CPU 2.80 GHz; Ram: 64 GB

SimPEG MT: Magneto Tellurics



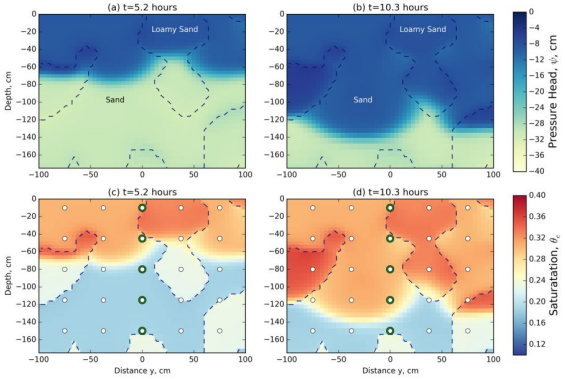
```
In [1]: import simpegMT as MT

In [ ]: MT.
```

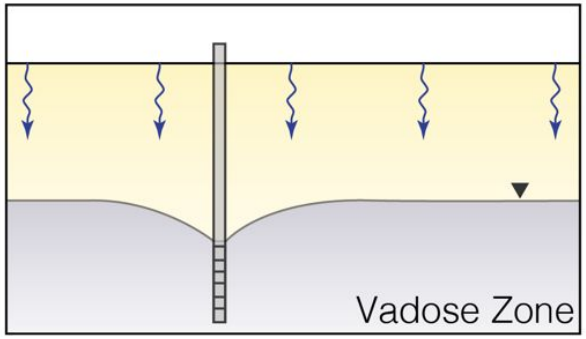
```
MT.BaseMT
MT.DataMT
MT.FieldsMT
MT.ProblemMT1D
MT.ProblemMT2D
MT.ProblemMT3D
MT.Sources
MT.SurveyMT
MT.Utils
```



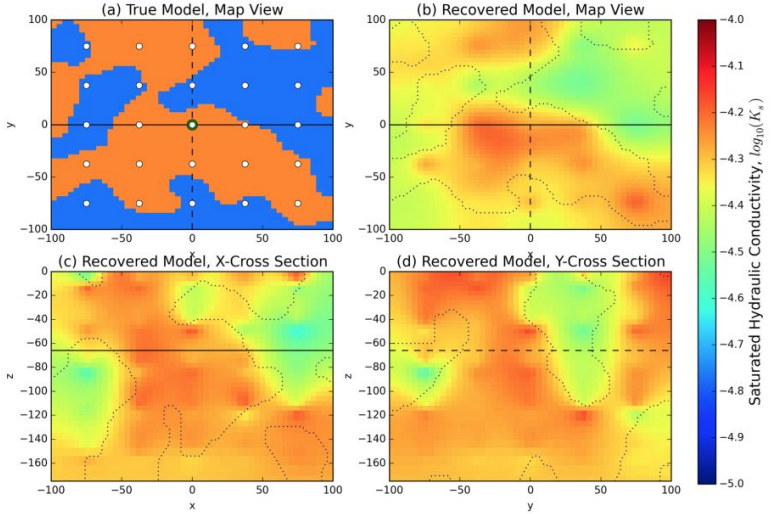
Rosenkjær *et al* 2015

Fluid Flow in the Vadose Zone: Richards Eqn.





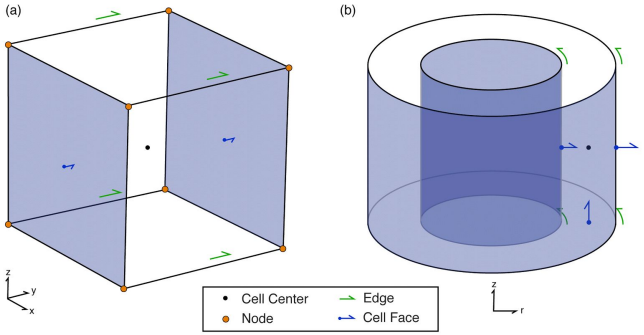
$$\frac{\partial \theta(\psi)}{\partial t} - \underbrace{\nabla \cdot K(\psi)\nabla\psi}_{\text{Diffusion}} - \underbrace{\frac{\partial K(\psi)}{\partial z}}_{\text{Gravity}} = 0$$

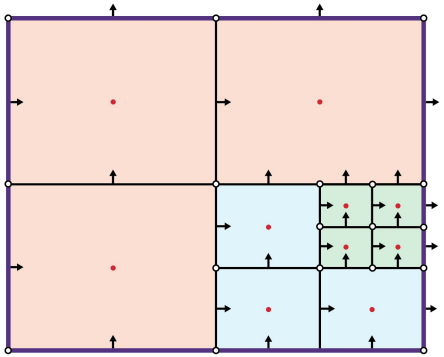


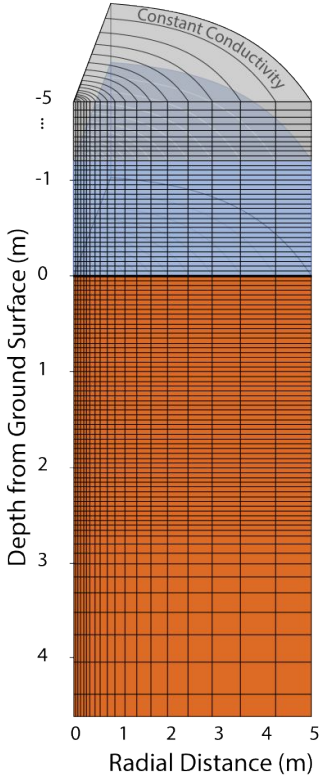
Cockett & Haber (2016)

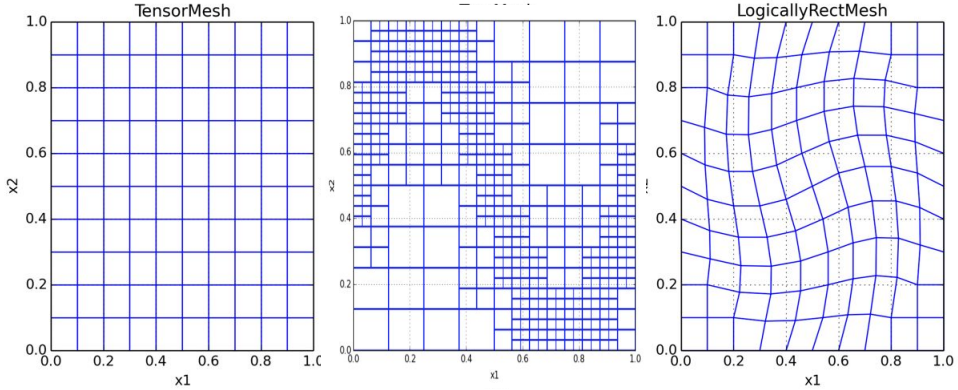
Finite Volume

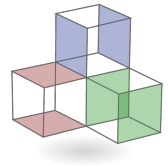
1D, 2D, 3D, 4D
Tensor, Cylinder, QuadTree, OcTree, Curvilinear





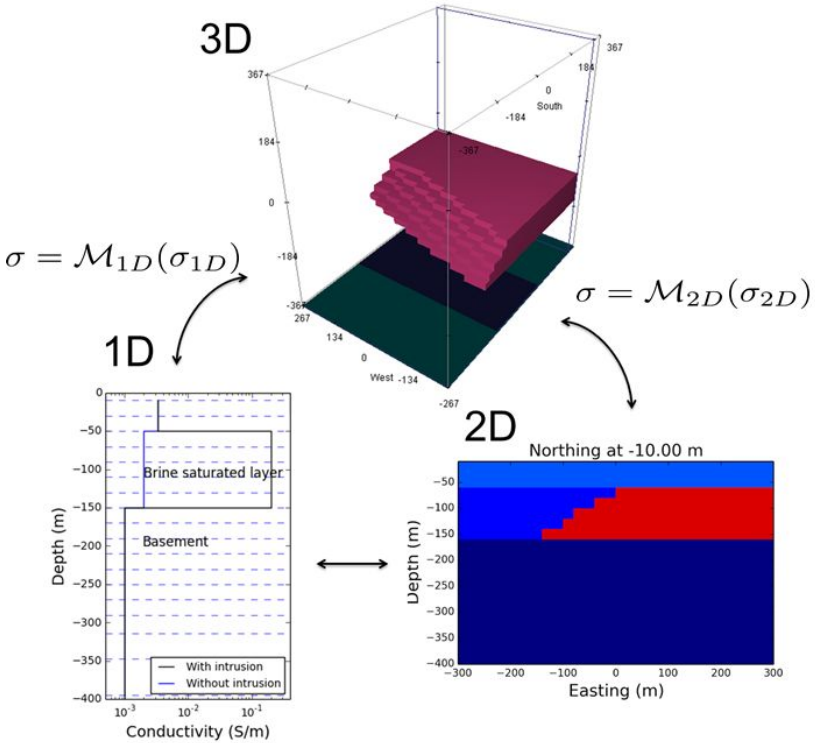


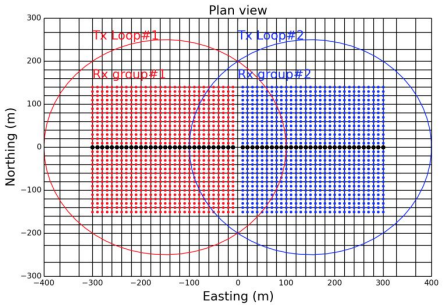


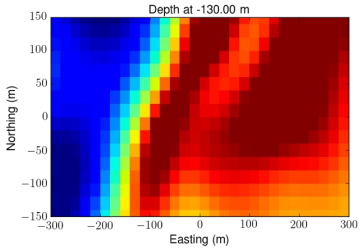


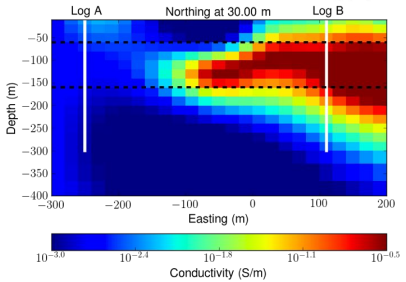
Exploring Model Spaces

Saltwater intrusions

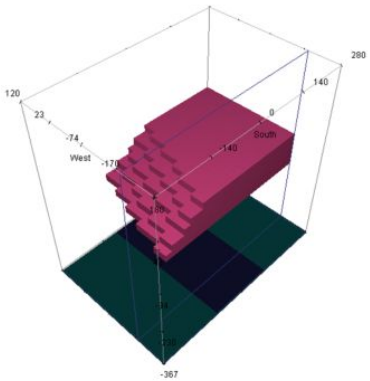




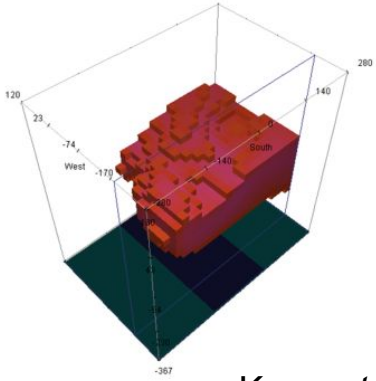




True intruded seawater



Estimated intruded seawater



Kang *et al* (2015)

Education

Problem: DC Resistivity??

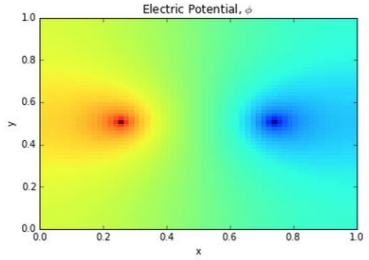
$$\nabla \cdot \sigma \nabla \phi = -q$$

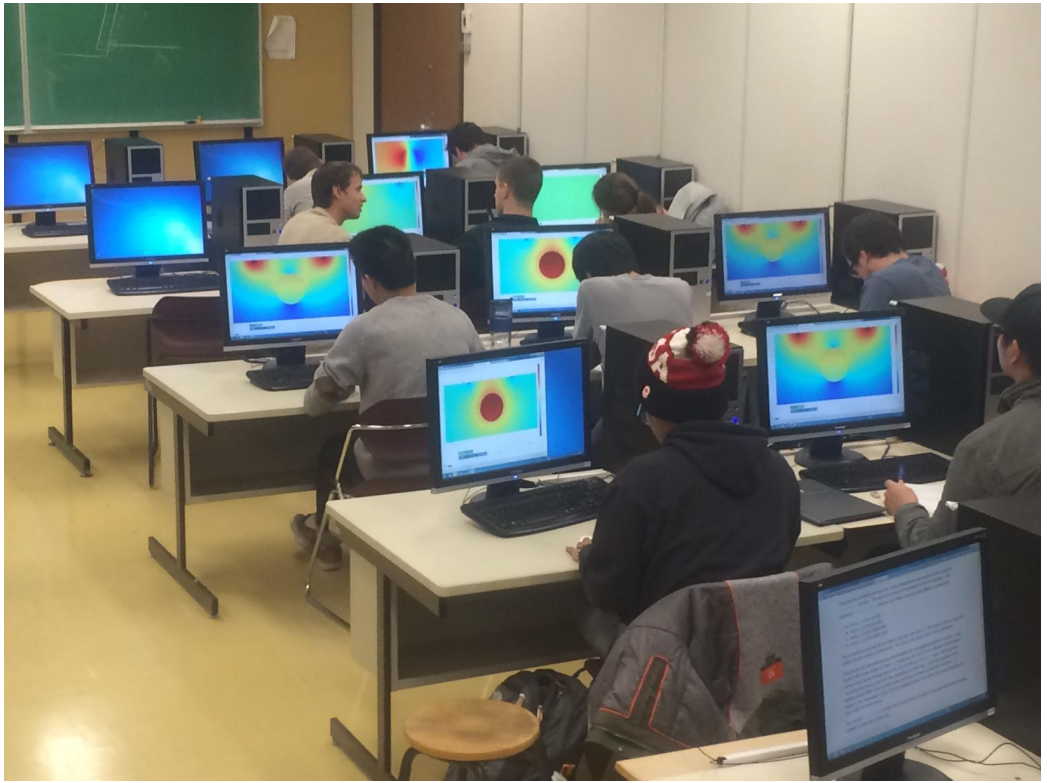
IMAGE NOT AVAILABLE

DC Resistivity

$$\nabla \cdot \sigma \nabla \phi = -s$$

```
In [9]: # Construct A Matrix
        Div, Sigma, Grad = getOperators(mesh,sigma)
        A = Div * Sigma * Grad # looks like the equation!
        Ainv = Solver(A)

In [10]: phi = Ainv * -s

In [11]: mesh.plotImage(phi)
         plt.title('Electric Potential, $\phi$')

Out[11]: <matplotlib.text.Text at 0x108b6e210>
```



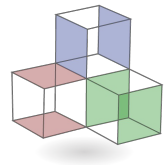



scikit
learn

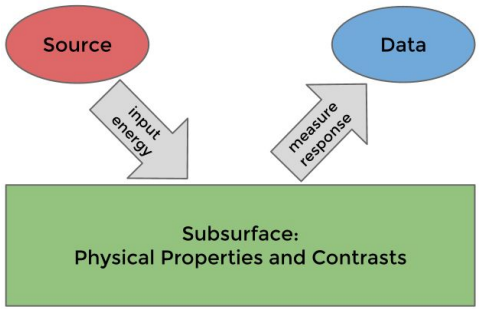

20,299 commits
15 branches
67 releases
522 contributors

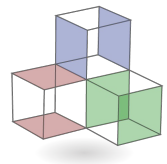

astropy
A Community Python Library for Astronomy


14,232 commits
10 branches
38 releases
146 contributors



Using: Math, Physics, Geology, Computer Science (Tools!)
Goals: Explore & Experiment (Research!)
With: Other people (Collaborate!)





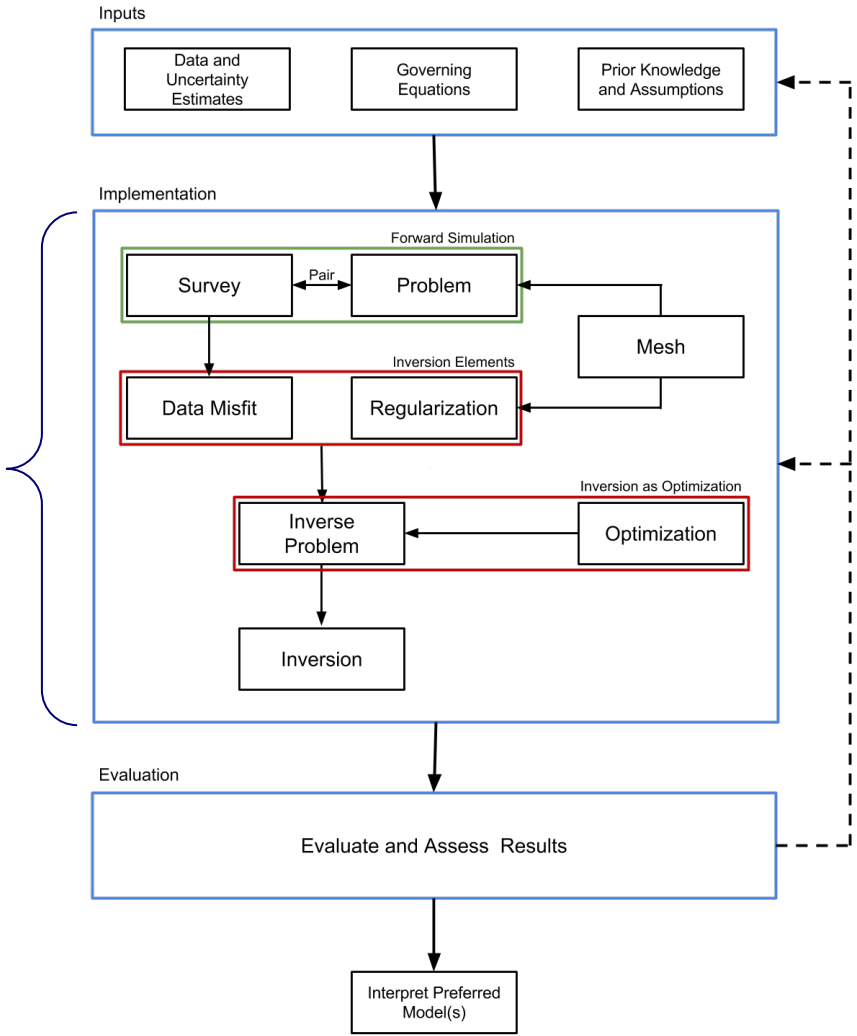
A framework

Implemented in Python!

```
In [1]: import SimPEG

In [ ]: SimPEG.
```

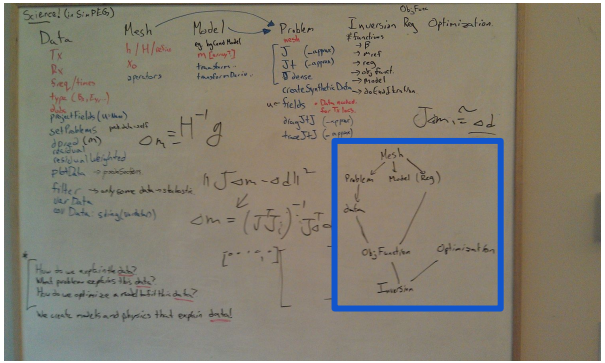
SimPEG.DataMisfit
SimPEG.Directives
SimPEG.Fields
SimPEG.InvProblem
SimPEG.Inversion
SimPEG.Maps
SimPEG.Mesh
SimPEG.Models
SimPEG.Optimization
SimPEG.Problem

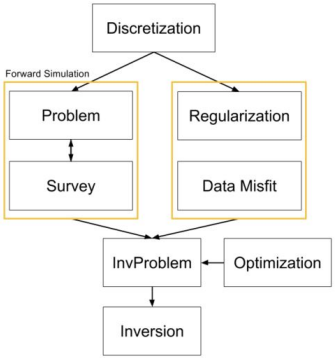
jupyter

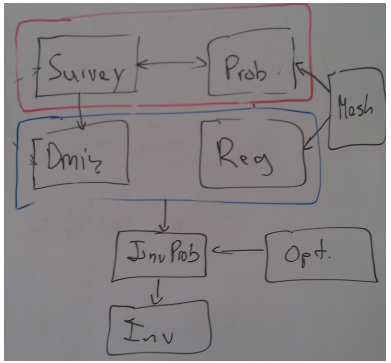


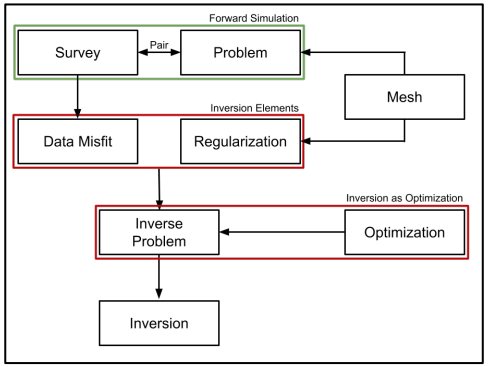
Cockett *et al* 2015

A framework







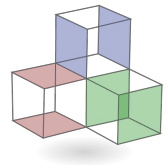


Time

Terminology and Structure is important

Changing the picture means the code changes

Implementation is ‘close’ to how you talk about the problem

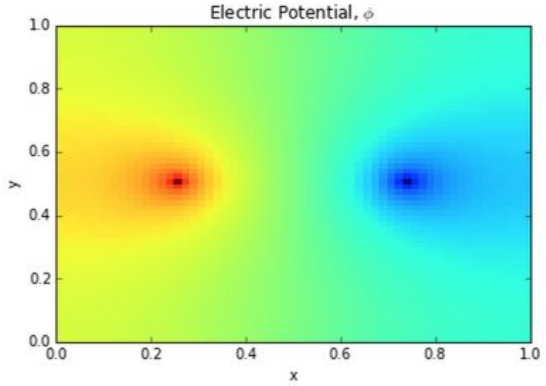


Python

DC Resistivity

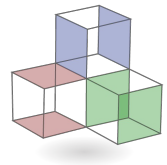
$$\nabla \cdot \sigma \nabla \phi = -s$$

```
In [9]: # Construct A Matrix
        Div, Sigma, Grad = getOperators(mesh,sigma)
        A = Div * Sigma * Grad # looks like the equation!
        Ainv = Solver(A)

In [10]: phi = Ainv * -s

In [11]: mesh.plotImage(phi)
         plt.title('Electric Potential, $\phi$')

Out[11]: <matplotlib.text.Text at 0x108b6e210>
```



Open source
Large & growing
Interactive
High level abstractions
Interfaces to fast things

Think then implement
Consistent terminology



jupyter Richards 3D Last Checkpoint: 07/06/2015 (autosaved)

File Edit View Insert Cell Kernel Help | Python 2

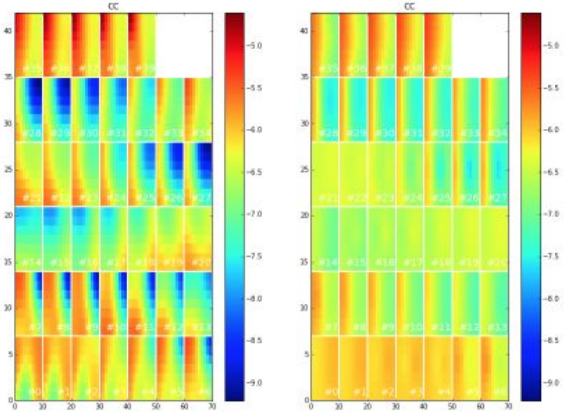
Code | Cell Toolbar: None

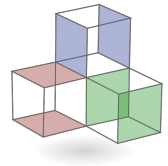
In [13]:
```
%%time
m0 = np.ones_like(mtrue)*mtrue.mean()
mopt = inv.run(m0)
```

```
SimPEG.InvProblem will set Regularization.mref to m0.
SimPEG.InvProblem is setting bfgsH0 to the inverse of the eval2Deriv.
                    ***Done using same solver as the problem***
SimPEG.l2_DataMisfit is creating default weightings for Wd.
========================= Inexact Gauss Newton ==========================
  #     beta     phi_d     phi_m       f      |proj(x-g)-x|  LS    Comment
-------------------------------------------------------------------------
   0  1.79e+02  1.39e+04  1.20e-01  1.39e+04    2.73e+03      0
   1  1.79e+02  2.92e+03  1.17e-01  2.94e+03    7.80e+02      0
   2  1.79e+02  8.83e+02  2.50e+00  1.33e+03    9.90e+01      0   Skip BFGS
   3  2.24e+01  7.80e+02  2.71e+00  8.40e+02    1.13e+02      0
   4  2.24e+01  3.57e+02  9.16e+00  5.63e+02    3.78e+02      0
   5  2.24e+01  2.11e+02  9.33e+00  4.20e+02    3.10e+01      0
   6  2.80e+00  1.76e+02  9.77e+00  2.04e+02    6.39e+01      0
   7  2.80e+00  1.34e+02  1.20e+01  1.68e+02    1.17e+02      0
   8  2.80e+00  1.22e+02  1.29e+01  1.58e+02    1.03e+02      0
------------------------- STOP! -------------------------
1 : |fc-fOld| = 9.8167e+00 <= tolF*(1+|f0|) = 1.3903e+03
1 : |xc-x_last| = 2.4832e+00 <= tolX*(1+|x0|) = 3.4781e+01
0 : |proj(x-g)-x|    = 1.0346e+02 <= tolG          = 1.0000e-01
0 : |proj(x-g)-x|    = 1.0346e+02 <= 1e3*eps       = 1.0000e-02
1 : maxIter   =       8    <= iter          =      8
------------------------- DONE! -------------------------
CPU times: user 16min 1s, sys: 52.3 s, total: 16min 53s
Wall time: 4min 28s
```

In [14]:
```
plt.figure(figsize=(14, 10))
clim = [mtrue.min(),(mtrue).max()]
ax = plt.subplot(121)
colorbar(M.plotImage(mtrue,ax=ax,clim=clim))
ax = plt.subplot(122)
colorbar(M.plotImage(mopt,ax=ax,clim=clim))
```

Out[14]: <matplotlib.colorbar.Colorbar instance at 0x10d7c8368>

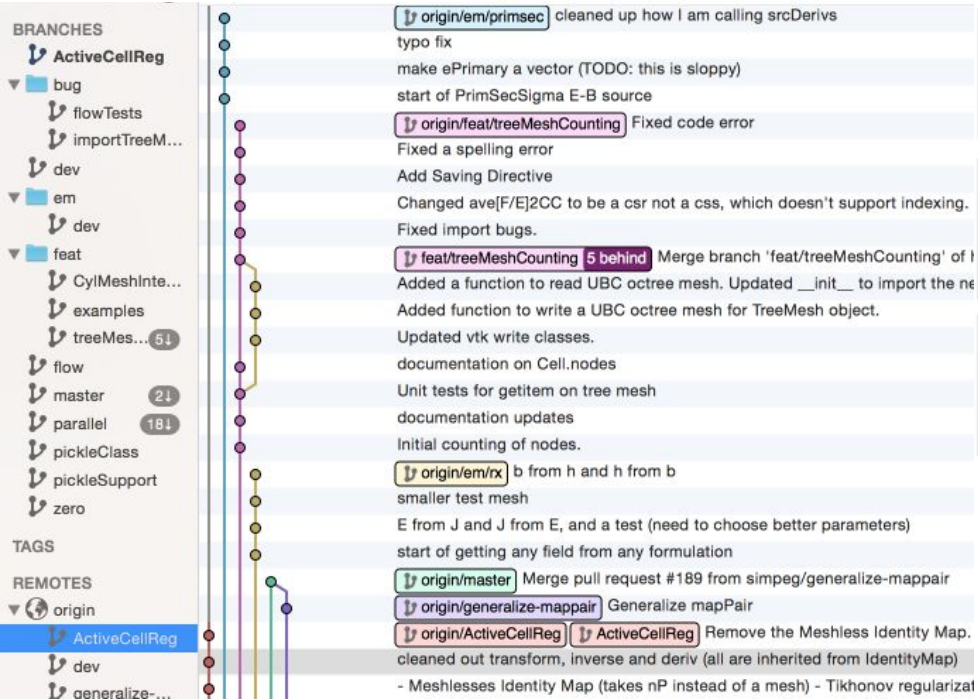


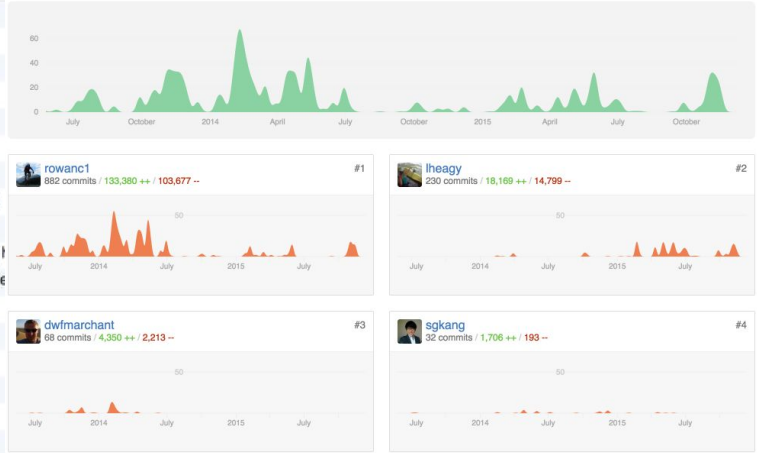


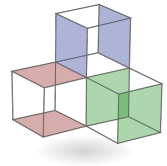
Version control
History (iterations)

Work in parallel
Tinker without fear
Stats

GitHub







simpeg / simpeg build passing

Current Branches Build History Pull Requests Settings

✓ master Merge pull request #189 from simpeg/generalize-mappair

Generalize mapPair

Commit 8b53fd5

Compare ca3b44b..8b53fd5

Rowan Cockett authored and committed

#707 passed

Elapsed time 43 min 59 sec

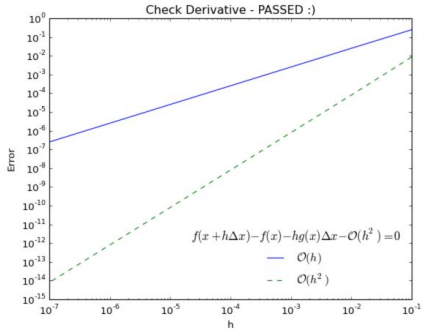
Total time for 1 hr 38 min 36 sec

9 days ago

Build Jobs

✓ # 707.1	Python: 2.7	TEST_DIR=tests/em/examples	2 min 22 sec
✓ # 707.2	Python: 2.7	TEST_DIR=tests/em/fdem/forward	22 min 38 sec
✓ # 707.3	Python: 2.7	TEST_DIR=tests/em/fdem/inverse/derivs	43 min 59 sec
✓ # 707.4	Python: 2.7	TEST_DIR=tests/em/fdem/inverse/adjoint	15 min 18 sec
✓ # 707.5	Python: 2.7	TEST_DIR=tests/em/tdem	1 min 56 sec
✓ # 707.6	Python: 2.7	TEST_DIR=tests/mesh	6 min 23 sec
✓ # 707.7	Python: 2.7	TEST_DIR=tests/flow	2 min 48 sec
✓ # 707.8	Python: 2.7	TEST_DIR=tests/utils	59 sec
✓ # 707.9	Python: 2.7	TEST_DIR=tests/base	1 min 15 sec
✓ # 707.10	Python: 2.7	TEST_DIR=tests/examples	58 sec

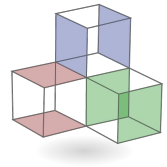
```
test_order (tests.mesh.test_operators.TestCurl2D) ... ok
test_order (tests.mesh.test_operators.TestFaceDiv2D) ... ok
test_order (tests.mesh.test_operators.TestFaceDiv3D) ... ok
test_order (tests.mesh.test_operators.TestNodalGrad) ... ok
test_order (tests.mesh.test_operators.TestNodalGrad2D) ... ok
test_area_3D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_cent_neg (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_centering (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_edge_2D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_edge_3D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_negative (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_oneCell (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_printing (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_tensor (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_vectorCC_2D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_vectorN_2D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_vol_2D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_vol_3D (tests.mesh.test_tensorMesh.BasicTensorMeshTests) ... ok
test_orderBackward (tests.mesh.test_tensorMesh.TestPoissonEqn) ... ok
test_orderForward (tests.mesh.test_tensorMesh.TestPoissonEqn) ... ok
test_both (tests.utils.test_Zero.Tests) ... ok
test_mat_one (tests.utils.test_Zero.Tests) ... ok
test_mat_shape (tests.utils.test_Zero.Tests) ... ok
test_mat_zero (tests.utils.test_Zero.Tests) ... ok
test_numpy_one (tests.utils.test_Zero.Tests) ... ok
test_one (tests.utils.test_Zero.Tests) ... ok
test_zero (tests.utils.test_Zero.Tests) ... ok
test_rotateMatrixFromNormals (tests.utils.test_coordutils.coorUtilsTest) ... ok
test_rotatePointsFromNormals (tests.utils.test_coordutils.coorUtilsTest) ... ok
test_rotationMatrixFromNormals (tests.utils.test_coordutils.coorUtilsTest) ... ok
test_simpleFail (tests.utils.test_utils.TestCheckDerivative) ... ok
test_simpleFunction (tests.utils.test_utils.TestCheckDerivative) ... ok
test_simplePass (tests.utils.test_utils.TestCheckDerivative) ... ok
test_simpleFail (tests.utils.test_utils.TestCounter) ... ok
testOnes (tests.utils.test_utils.TestDiagEst) ... ok
test_TensorType2D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_TensorType3D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_asArray_N_x_Dim (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_ind2sub (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_indexCube_2D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_indexCube_3D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_invPropertyTensor2D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_invPropertyTensor3D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_invXXXBlockDiagonal (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_isScalar (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_mkvc1 (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_mkvc2 (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_mkvc3 (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_ndgrid_2D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_ndgrid_3D (tests.utils.test_utils.TestSequenceFunctions) ... ok
test_sub2ind (tests.utils.test_utils.TestSequenceFunctions) ... ok

----------------------------------------------------------------------
Ran 430 tests in 245.984s

OK
```




```
========== checkDerivative ==========
iter    h         |ft-f0|    |ft-f0-h*J0*dx|  Order
-------------------------------------------------------
 0   1.00e-01    7.989e-06     7.299e-07       nan
 1   1.00e-02    8.212e-07     7.714e-09       1.976
 2   1.00e-03    8.239e-08     7.666e-11       2.003
========== PASS! ==========
You are awesome.
```

Testing
Consistency
Reproducibility

Rigour
Ensure assumptions
Accountability



COVERALLS

build passing coverage 77%

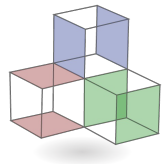
ALL 64 | CHANGED 1 | SOURCE CHANGED 1 | COVERAGE CHANGED 0 | SHOW 10 ENTRIES

COVERAGE	FILE	LINES	RELEVANT	COVERED	MISSED	HITS/LINE
91.62	SimPEG/Survey.py	379	191	175 +7	16 -7	1.0
91.59	SimPEG/Mesh/TensorMesh.py	568	321	294	27	1.0
91.28	SimPEG/EM/FDEM/FieldsFDEM.py	460	321	293 +206	28 -206	1.0 +1.0
90.63	SimPEG/Utils/SolverUtils.py	157	96	87 +1	9 -1	1.0
88.77	SimPEG/Mesh/BaseMesh.py	596	187	166	21	1.0
88.24	SimPEG/EM/Analytics/FDEMcasing.py	98	68	60 +45	8 -45	1.0 +1.0
87.18	SimPEG/EM/Base.py	186	78	68 +22	10 -22	1.0 +1.0
87.04	SimPEG/Mesh/DiffOperators.py	749	494	430	64	1.0
85.87	SimPEG/Problem.py	217	92	79	13	1.0
85.5	SimPEG/Mesh/CylMesh.py	398	200	171 +11	29 -11	1.0
84.62	SimPEG/EM/FDEM/SurveyFDEM.py	181	91	77 +54	14 -54	1.0 +1.0

```
15  def __init__(self, mesh, survey, **kwargs):                                  9×
16      self.survey = survey                  all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
17      self.mesh = mesh                      all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
18      Utils.setKwargs(self, **kwargs)       all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
19      self._fields = {}                     all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
20
21      if self.knownFields is None:          all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
22          raise Exception('knownFields cannot be set to None')                  !
23      if self.aliasFields is None:          all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
24          self.aliasFields = {}             only TEST_DIR=tests/base, TEST_DIR=tests/em/examples, and
                                              TEST_DIR=tests/em/tdem - 3×
25
26      allFields = [k for k in self.knownFields] + [a for a in   all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
self.aliasFields]                             TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
27      assert len(allFields) == len(set(allFields)), 'Aliased fields and   all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
Known Fields have overlapping definitions.'   TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
28      self.startup()                        all except TEST_DIR=tests/examples, TEST_DIR=tests/mesh,
                                              TEST_DIR=tests/utils, and TEST_DIR=tests/flow - 5×
29
30  def startup(self):                                                            9×
31      pass                                  only TEST_DIR=tests/base, TEST_DIR=tests/em/examples, and
                                              TEST_DIR=tests/em/tdem - 3×
```

> *I have gotten all the IO for the meshes to work, both for octree and tensor. **I will write up tests this afternoon** so hopefully we can merge that into the SimPEG.*

See what is tested
Culture of testing



Your comments are now readable, searchable documentation.

Internal communication
Accessibility

```python
def getInterpolationMat(self, loc, locType, zerosOutside=False):
    """ Produces interpolation matrix

    :param numpy.ndarray loc: Location of points to interpolate to
    :param str locType: What to interpolate (see below)
    :rtype: scipy.sparse.csr.csr_matrix
    :return: M, the interpolation matrix

    locType can be::

        'Ex'    -> x-component of field defined on edges
        'Ey'    -> y-component of field defined on edges
        'Ez'    -> z-component of field defined on edges
        'Fx'    -> x-component of field defined on faces
        'Fy'    -> y-component of field defined on faces
        'Fz'    -> z-component of field defined on faces
        'N'     -> scalar field defined on nodes
        'CC'    -> scalar field defined on cell centers
    """
    if self._meshType == 'CYL' and self.isSymmetric and locType in ['Ex','Ez',
        raise Exception('Symmetric CylMesh does not support %s interpolation,

    loc = Utils.asArray_N_x_Dim(loc, self.dim)

    if zerosOutside is False:
        assert np.all(self.isInside(loc)), "Points outside of mesh"
    else:
        indZeros = np.logical_not(self.isInside(loc))
        loc[indZeros, :] = np.array([v.mean() for v in self.getTensor('CC')])
```

getInterpolationMat(*loc, locType, zerosOutside=False*) [source]

Produces interpolation matrix

Parameters:
- **loc** (*numpy.ndarray*) – Location of points to interpolate to
- **locType** (*str*) – What to interpolate (see below)

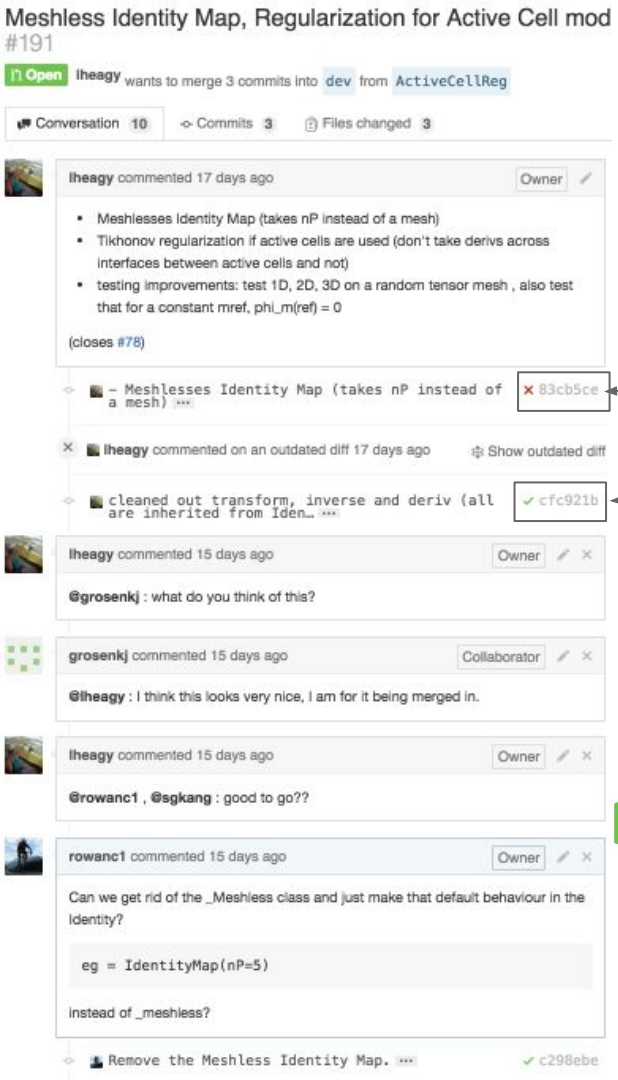
Return type: scipy.sparse.csr.csr_matrix

Returns: M, the interpolation matrix

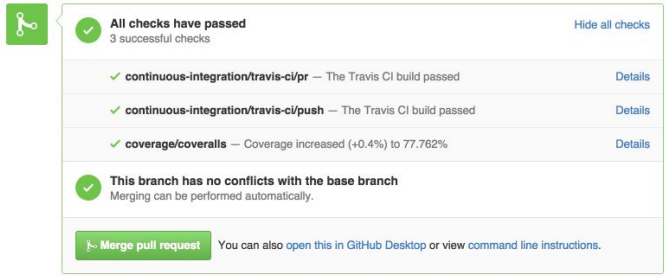
locType can be:

```
'Ex'    -> x-component of field defined on edges
'Ey'    -> y-component of field defined on edges
'Ez'    -> z-component of field defined on edges
'Fx'    -> x-component of field defined on faces
'Fy'    -> y-component of field defined on faces
'Fz'    -> z-component of field defined on faces
'N'     -> scalar field defined on nodes
'CC'    -> scalar field defined on cell centers
```

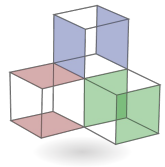
All together now



Tested every time!



Peer review
Commenting
Immediate feedback
Confidence



MIT

Can

- Commercial Use
- Modify
- Distribute
- Sublicense
- Private Use

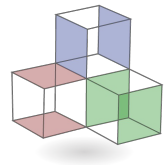
Cannot

- Hold Liable

Must

- Include Copyright
- Include License

Opens up the community
Input is more diverse



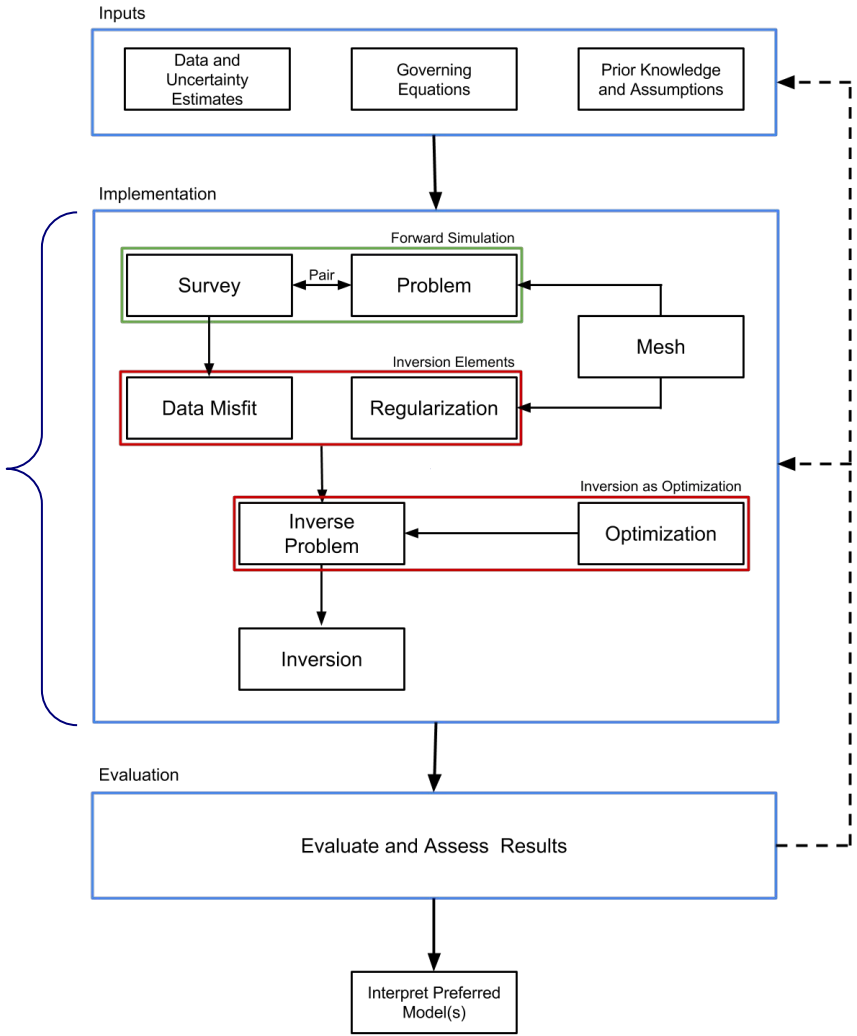
A framework

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



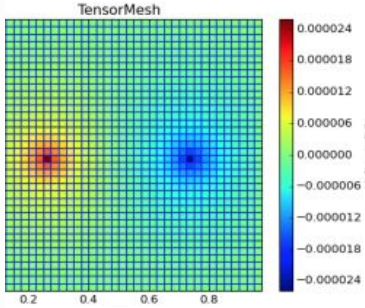
Cockett *et al* 2015

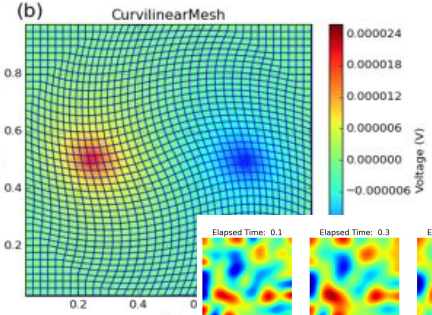
Examples

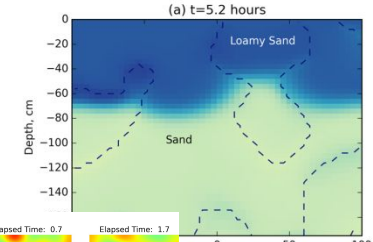
- EM: FDEM: 1D: Inversion
- FLOW: Richards: 1D: Celia1990
- Forward BasicDirectCurrent
- Inversion: Linear Problem
- Mesh: Basic: PlotImage
- Mesh: Basic: Types
- Mesh: Operators: Cahn Hilliard
- Mesh: QuadTree: Creation
- Mesh: QuadTree: FaceDiv
- Mesh: QuadTree: Hanging Nodes
- Mesh: Tensor: Creation

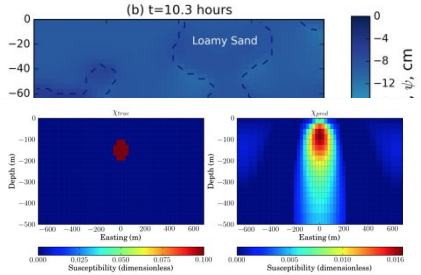
External Notebooks

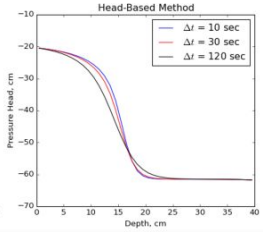
- Example 1: Direct Current
- Example 2: Seismic-Acoustic

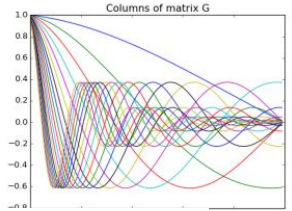


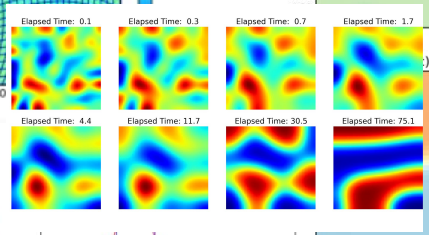


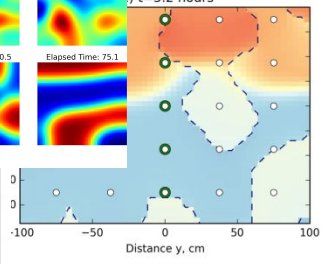


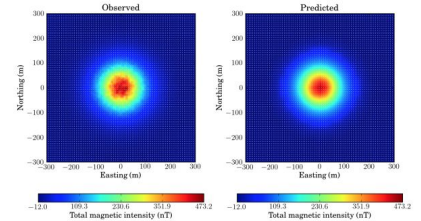


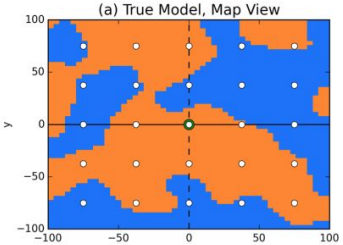


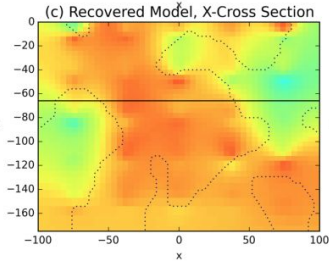


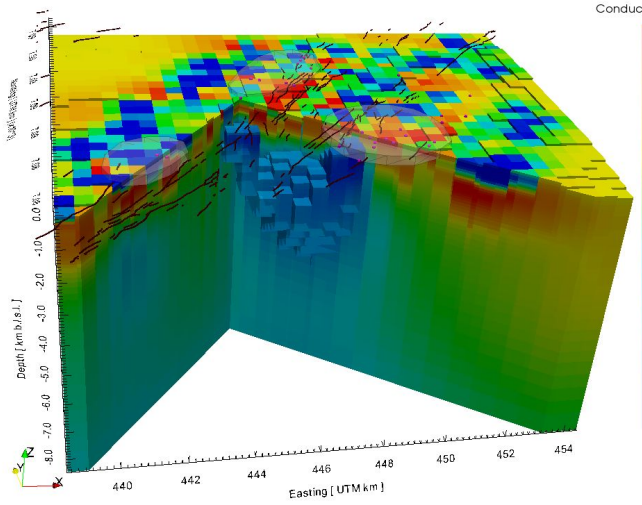


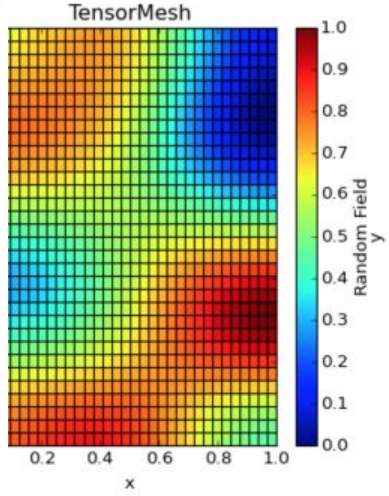


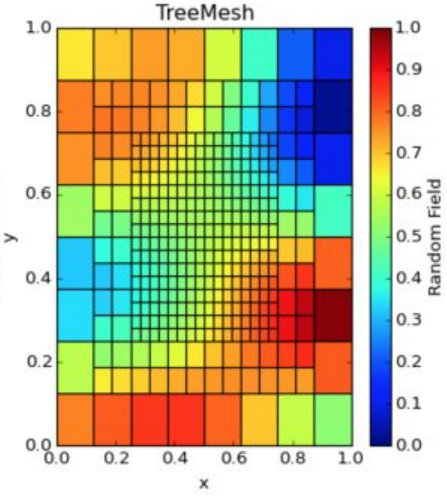


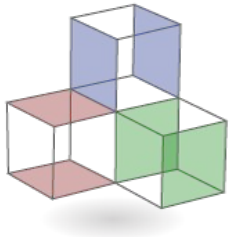












simpeg

Creating a **toolbox** and **framework** for geophysics
Focusing on flexibility and speed ***for the researcher***
Starting to build a **community**

Thank You!
&
Questions?

simpeg.xyz

github.com/simpeg

Except where noted, this work is licensed under a Creative Commons Attribution 4.0 International License