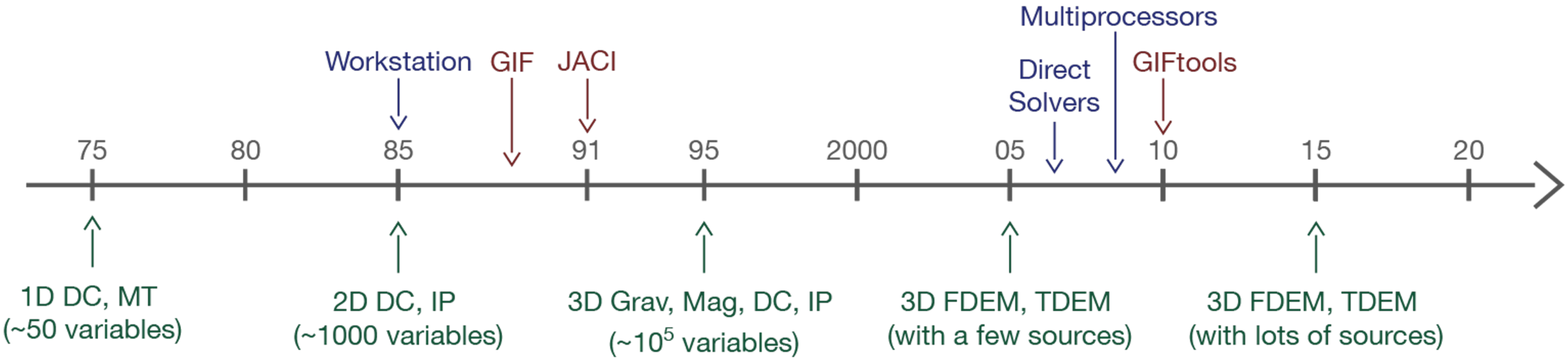
Advances in open-source software for 3D electromagnetics using SimPEG

L. Heagy[1], J. Kuttai[1], D. Cowan[1], J. Capriotti[1,2], S. Kang[3], D. Fournier[4], D. Oldenburg[1]

[1]University of British Columbia Geophysical Inversion Facility
[2]Colorado School of Mines
[3]Stanford Geophysics
[4]Mira Geoscience

some background: UBC-GIF

GIF Fortran Codes & GIF Tools



slide from: [Oldenburg, 2016: Looking back and moving forwards (BIRS)]

some background: UBC-GIF

Transition to open-source

By 2010's…

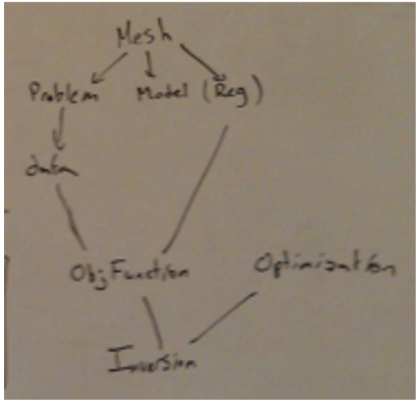
most forward simulations "solved"

next generation of research in inversions

- physical properties: anisotropy, high contrasts, magnetic permeability, IP in EM
- experimenting with inversion methods: choice of norm, including physical property & geologic information
- role of machine learning
- …

In 2013… started SimPEG

goal: common framework for inversions in geophysics to serve researchers



R. Cockett L. Heagy S. Kang

A framework for geophysical inversions

with application to vadose zone parameter estimation

by

Archa Rowan B. Cockett

B.Sc. Applied and Environmental Geology, University of Calgary, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Geophysics)

The University of British Columbia

(Vancouver)

December 2017

© Archa Rowan B. Cockett, 2017

simpeg

simulation and parameter estimation in geophysics

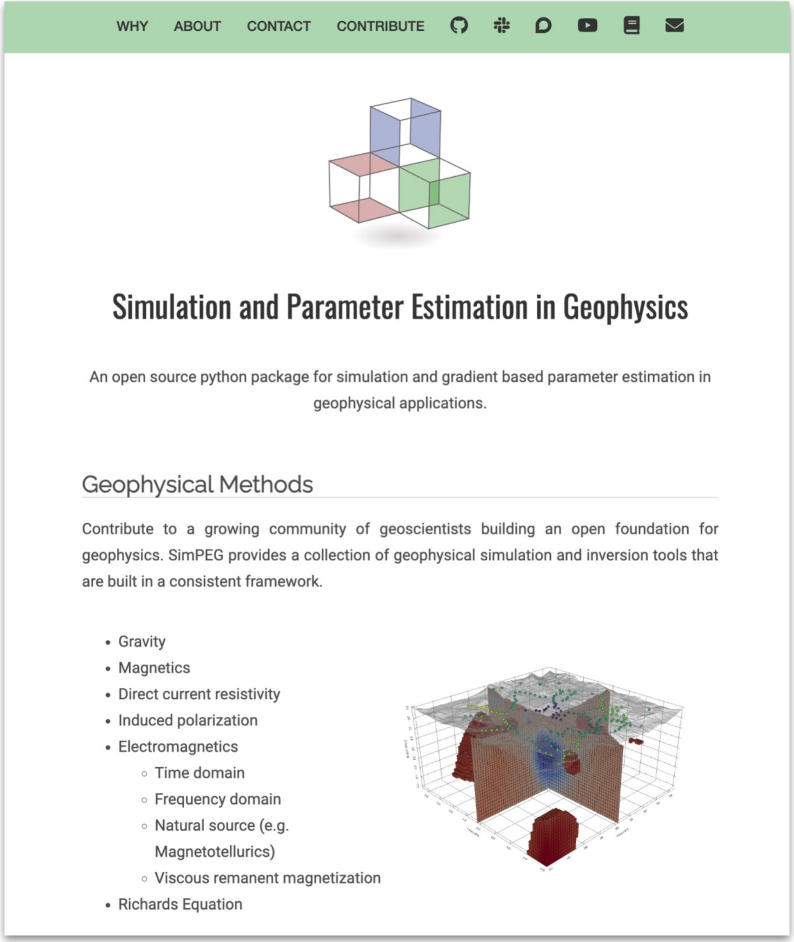
common framework for simulations & inversions

accelerate research: build upon others work

facilitate reproducibility of results

build & deploy in python

open-source



what simpeg solves

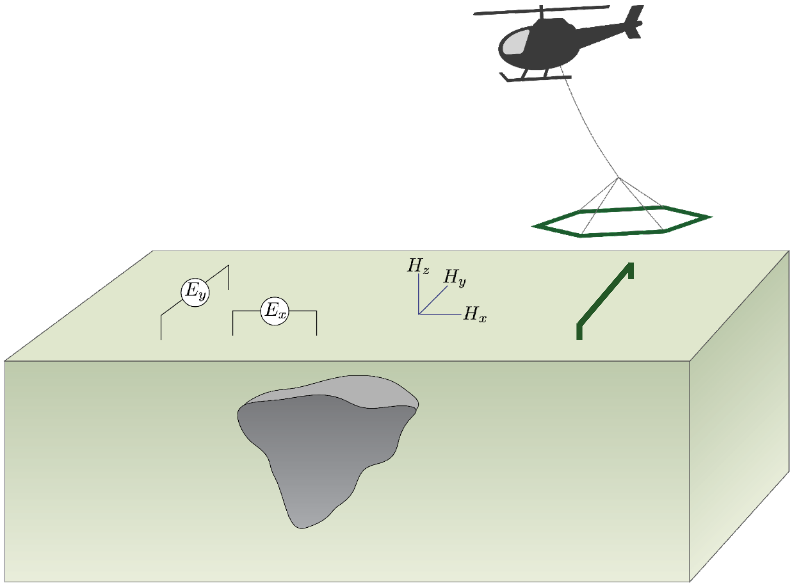
Given

- observations: $d_j^{obs}, \quad j = 1, \ldots, N$
- uncertainties: ϵ_j
- ability to forward model: $\mathcal{F}[m] = d$

Inverse problem: Find an Earth model that fits those data and a-priori information

$$\min_{\mathbf{m}} \phi(\mathbf{m}) = \phi_d(\mathbf{m}) + \beta \phi_m(\mathbf{m})$$

$$\text{s.t. } \phi_d \leq \phi_d^* \quad \mathbf{m}_L \leq \mathbf{m} \leq \mathbf{m}_U$$



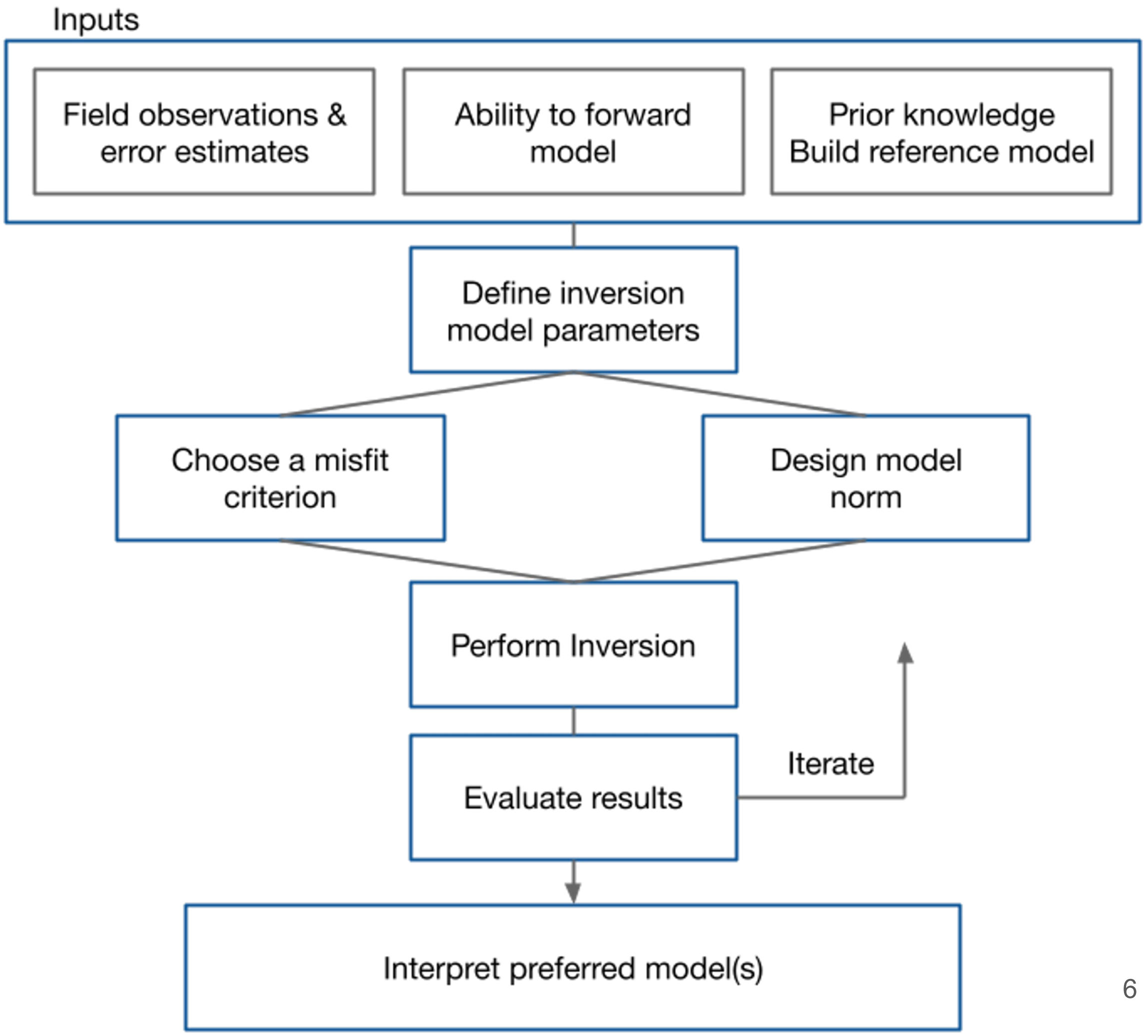
flow chart for the inverse problem

- many components
- iterative process to obtain solution
- each component requires evaluation, adjustment by user



Fundamentals of Inversion – D. Oldenburg
Capturing knowledge in code – L. Heagy

http://www.mtnet.info/EMinars/EMinars.html



electromagnetics: basic equations (quasi-static)

	Time	Frequency
Faraday's Law	$\nabla \times \vec{e} = -\dfrac{\partial \vec{b}}{\partial t}$	$\nabla \times \vec{E} = -i\omega \vec{B}$
Ampere's Law	$\nabla \times \vec{h} = \vec{j} + \dfrac{\partial \vec{d}}{\partial t}$	$\nabla \times \vec{H} = \vec{J} + i\omega \vec{D}$
No Magnetic Monopoles	$\nabla \cdot \vec{b} = 0$	$\nabla \cdot \vec{B} = 0$
Constitutive Relationships (non-dispersive)	$\vec{j} = \sigma \vec{e}$ $\vec{b} = \mu \vec{h}$ $\vec{d} = \varepsilon \vec{e}$	$\vec{J} = \sigma \vec{E}$ $\vec{B} = \mu \vec{H}$ $\vec{D} = \varepsilon \vec{E}$

* Solve with sources and boundary conditions

electromagnetics: frequency domain equations

Use constitutive relations, reduce to two equations, one field, one flux

$$\nabla \times \vec{E} + i\omega \vec{B} = 0$$

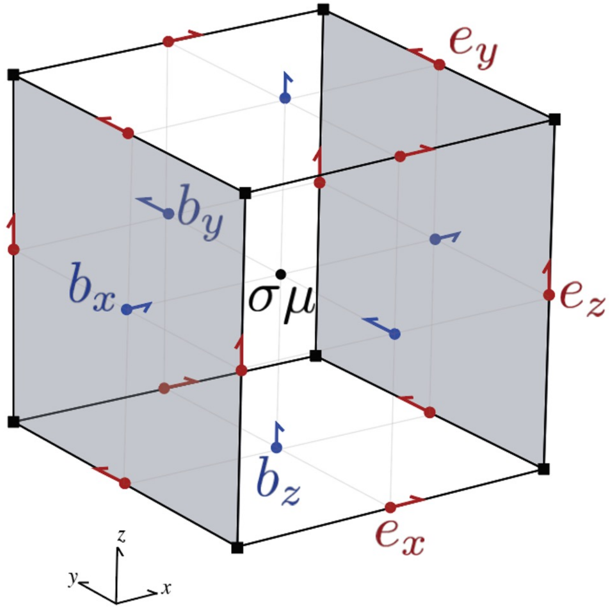
$$\nabla \times \mu^{-1} \vec{B} - \sigma \vec{E} = \vec{J}_s$$

$$\hat{n} \times \vec{B}|_{\partial\Omega} = 0$$

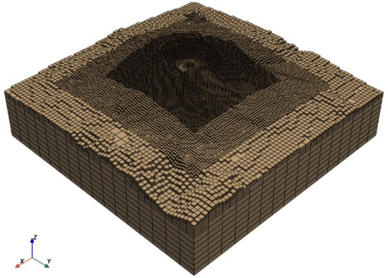
Finite volume discretization
(e.g. Haber, 2014; Cockett et al., 2016)

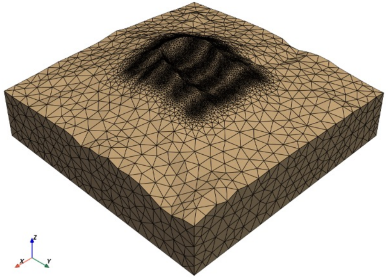
$$\mathbf{C}\mathbf{e} + i\omega \mathbf{b} = 0$$

$$\mathbf{C}^\top \mathbf{M}^f_{\mu^{-1}} \mathbf{b} - \mathbf{M}^e_\sigma \mathbf{e} = \mathbf{M}^e \mathbf{j}_s$$



Solving an FDEM problem





J. Capriotti

Mesh types:

- Tensor
- OcTree
- Curvilinear
- Tetrahedral

$$\underbrace{(\mathbf{C}^\top \mathbf{M}^f_{\mu^{-1}} \mathbf{C} + i\omega \mathbf{M}^e_\sigma)}_{\mathbf{A}(\sigma,\omega)} \underbrace{\mathbf{e}}_{\mathbf{u}} = \underbrace{-i\omega \mathbf{M}^e \mathbf{j_s}}_{\mathbf{q}(\omega)}$$

```
ω = 2 * pi * frequency

C = mesh.edge_curl
Mfμi = mesh.get_face_inner_product(1/mu_0)
Meσ = mesh.get_edge_inner_product(sigma)

A = C.T * Mfμi * C + i * ω * Meσ
Ainv = Solver(A)  # acts like A inverse

Me = mesh.get_edge_inner_product()
q = -i * ω * Me * js

u = Ainv * q
```

```python
from SimPEG import electromagnetics
```

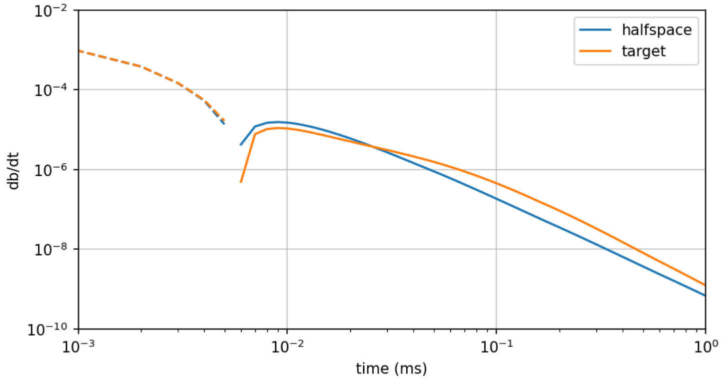
modules:

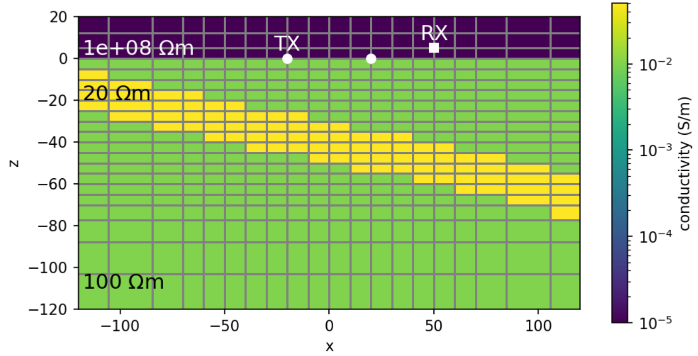
- frequency domain
- natural sources
- time domain

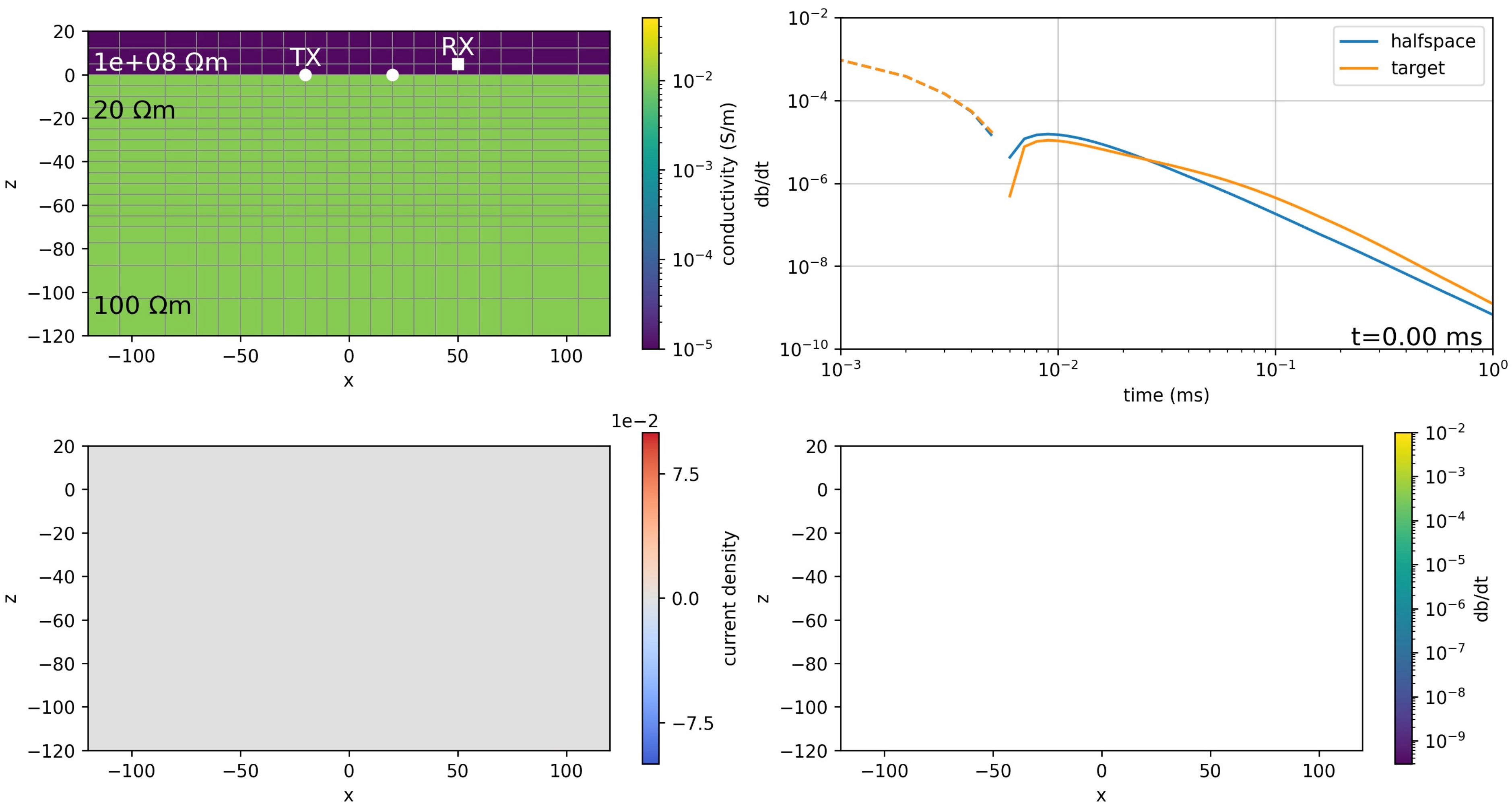
physical properties:

- supports anisotropic conductivity
- variable magnetic permeability
- IP parameters (1D fdem, tdem)

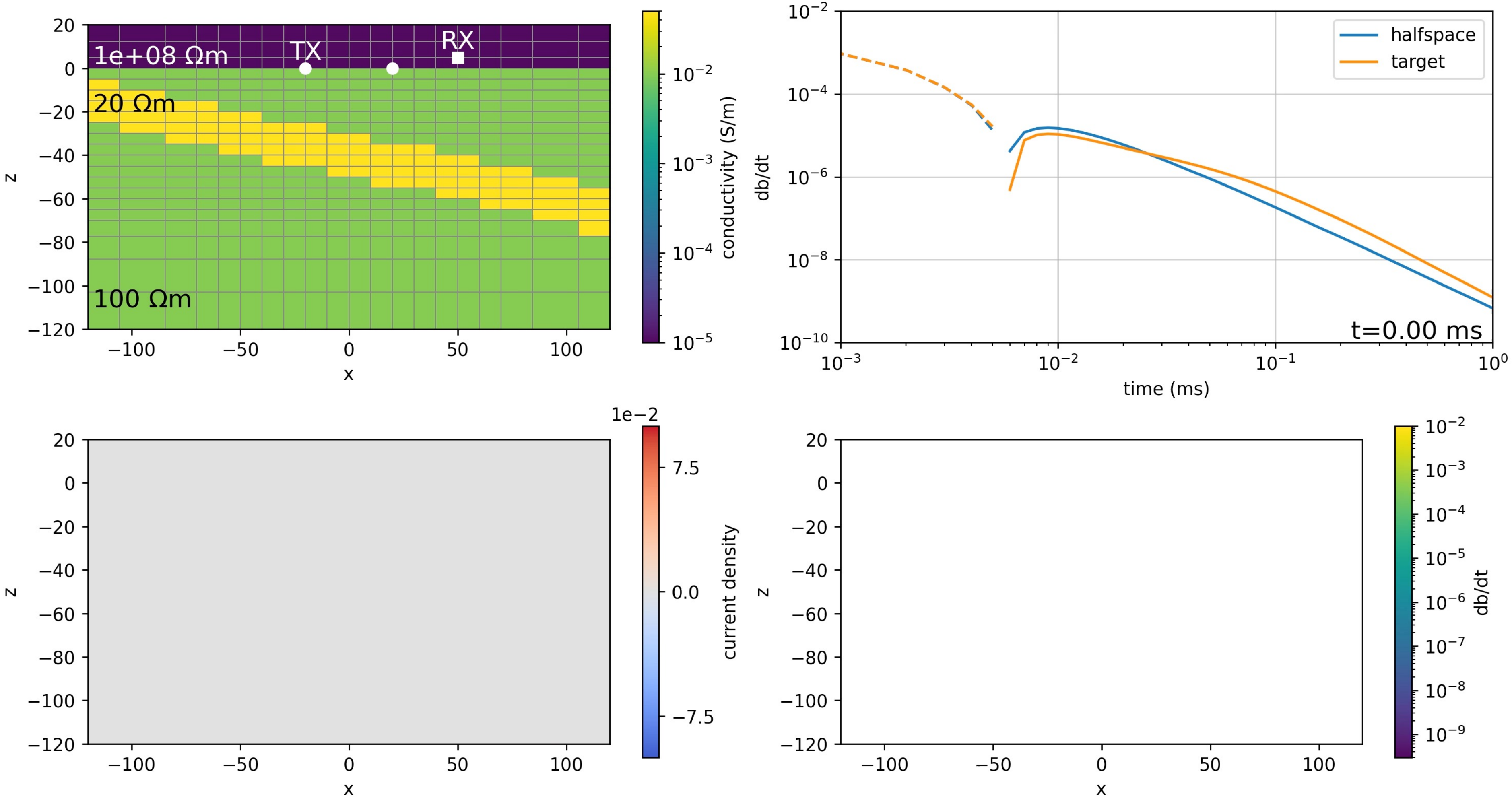
multiple formulations: E-B or H-J







1e+08 Ωm
TX
RX
20 Ωm
100 Ωm
z
x
conductivity (S/m)
halfspace
target
db/dt
time (ms)
t=0.00 ms
1e−2
current density
z
x
z
x
db/dt



1e+08 Ωm
TX
RX
20 Ωm
100 Ωm
z
x
conductivity (S/m)
halfspace
target
db/dt
time (ms)
t=0.00 ms
1e−2
current density
z
x
db/dt

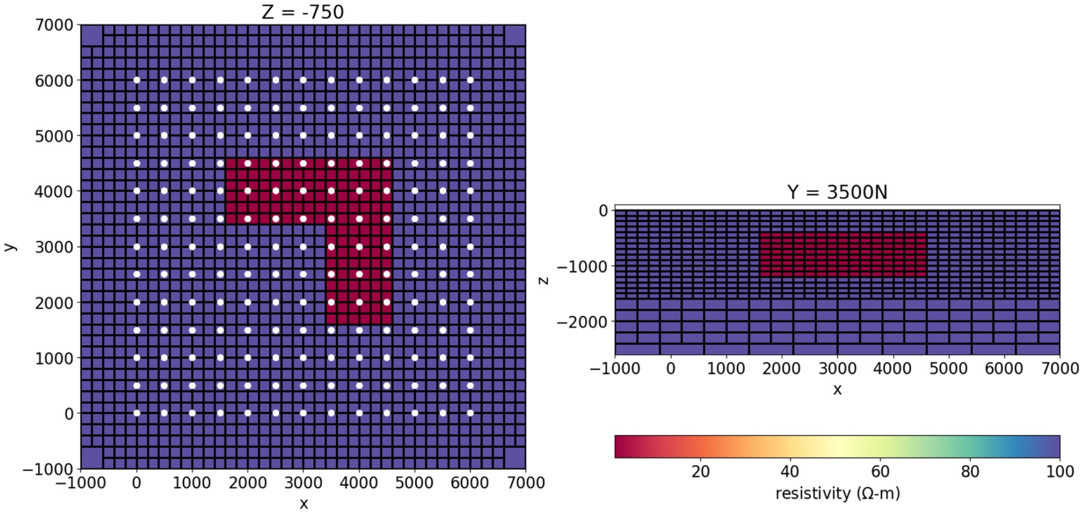
example: 3D simulation & inversion

model:

- conductive L-block (1 Ωm)
- half-space (100 Ωm)

survey:

- natural source EM
- frequencies: 10, 50, 200Hz
- data: ZTEM



simulation

model:

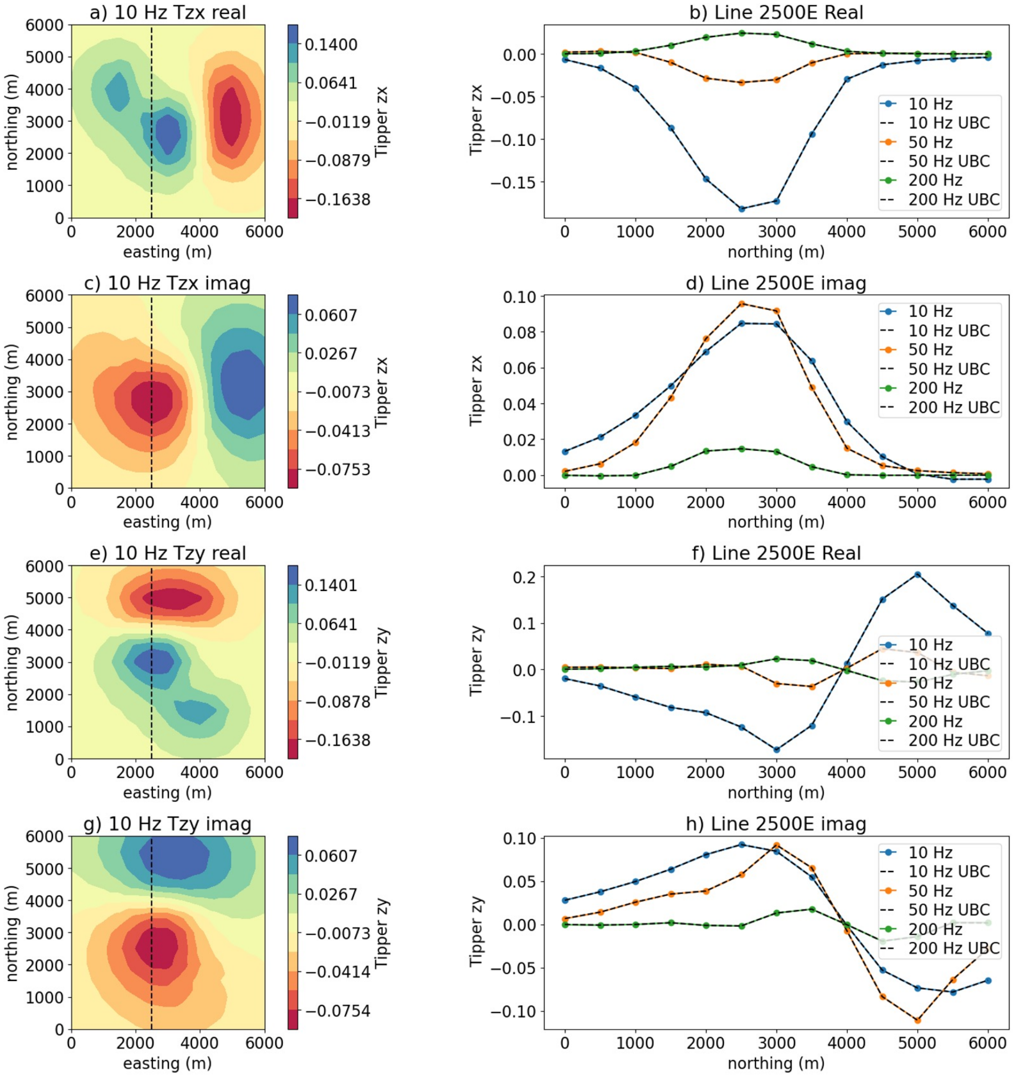
- conductive L-block (1 Ωm)
- half-space (100 Ωm)

survey:

- natural source EM
- frequencies: 10, 50, 200Hz
- data: ZTEM

simulation:

- OcTree mesh
- run SimPEG & UBC E3DMT



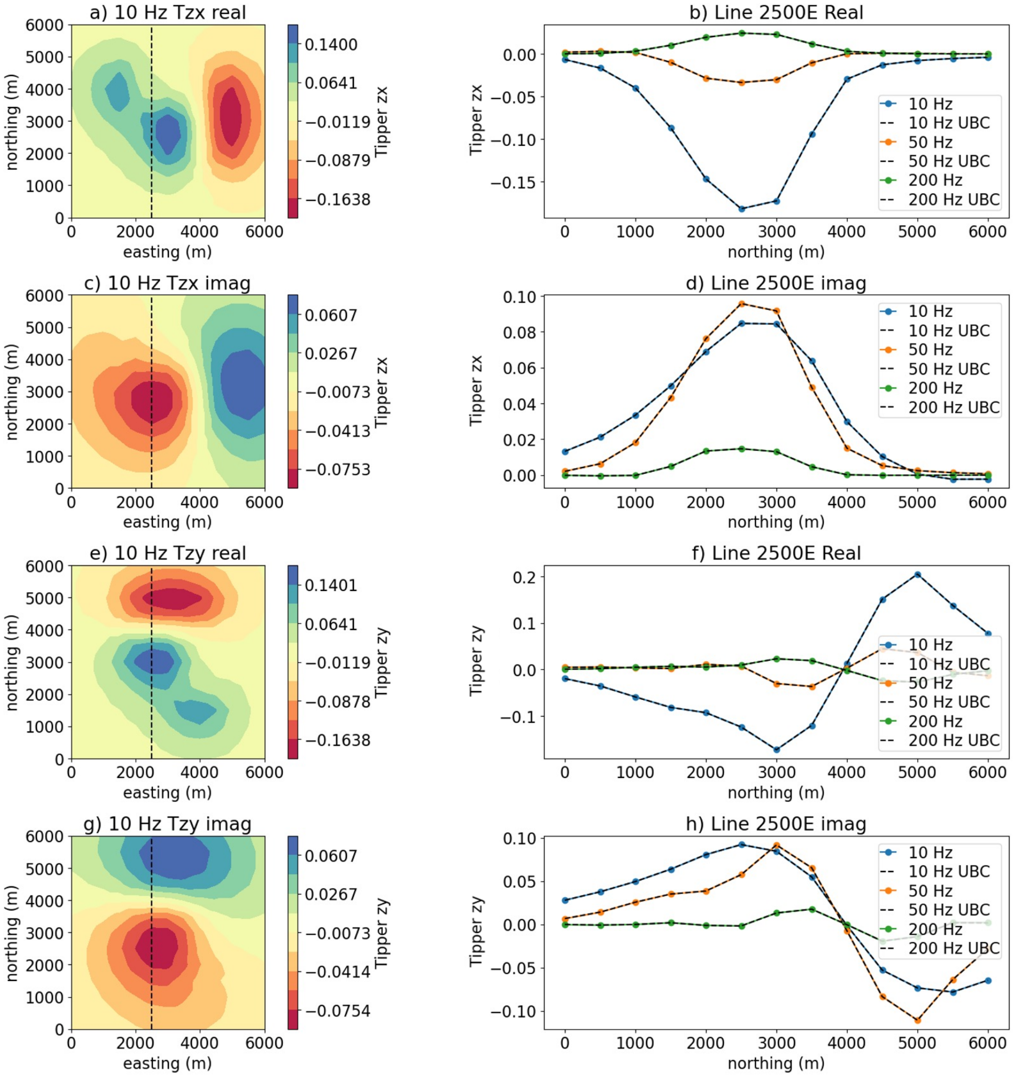
inversion

data

- 2028 data
- uncertainties:
  - 5% relative error
  - floor: 0.01 for 10 Hz, 50 Hz data, 0.005 for 200 Hz data

regularization

- L_2 smooth model
- identical beta-cooling schedule



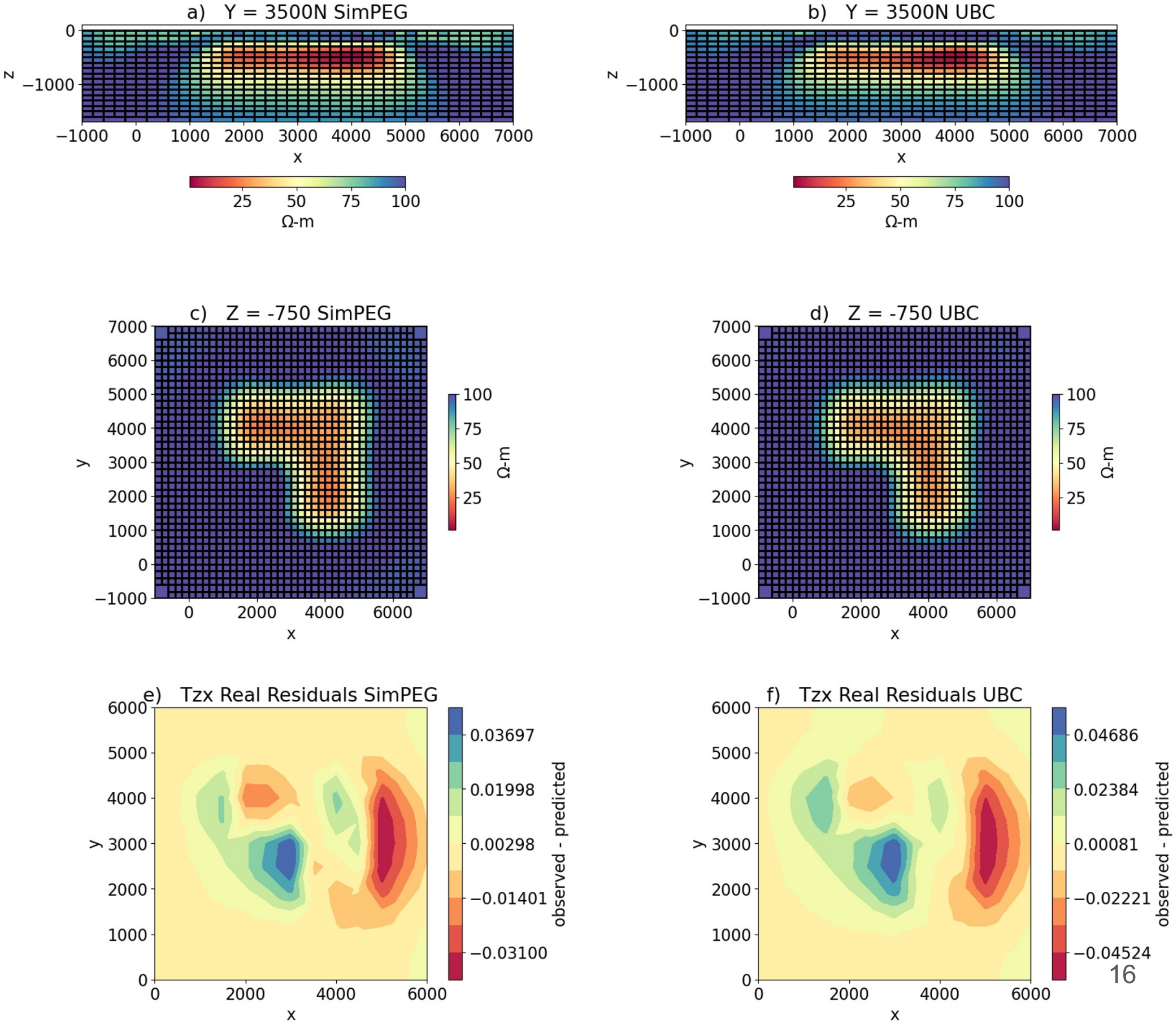
inversion

SimPEG

- 7 iterations
- final χ-factor of 0.47
- parallelized with Python multiprocessing
- runtime: 15 minutes

UBC E3DMT

- 8 iterations
- final χ-factor of 0.55
- parallelized with MPI
- runtime: 11 minutes



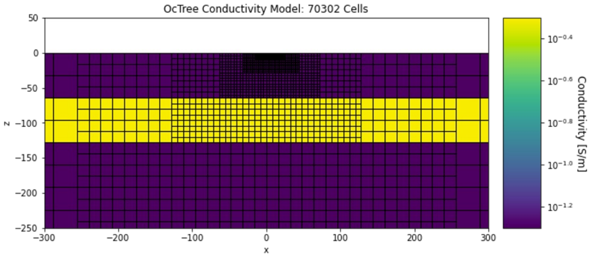
code validations

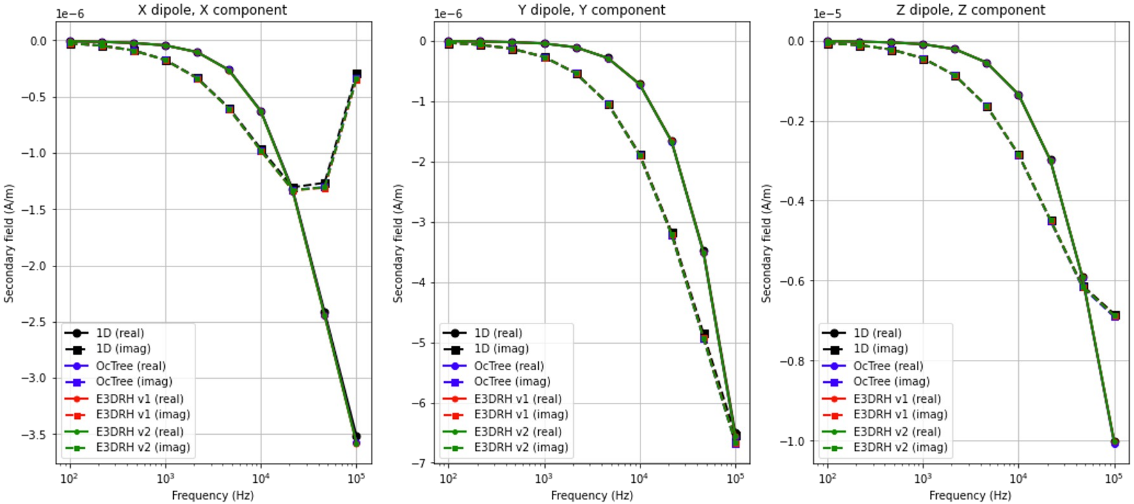
compare analytic, UBC Fortran, SimPEG

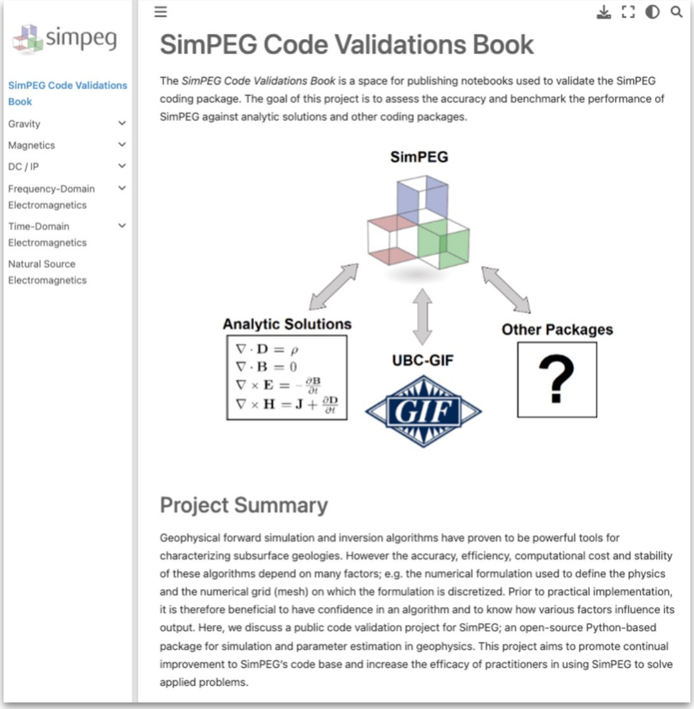
fdem dipole:

- z=5m
- separation: 10m

D. Cowan







simpeg.xyz/validations

example: AEM inversion with sparse norms

S. Kang

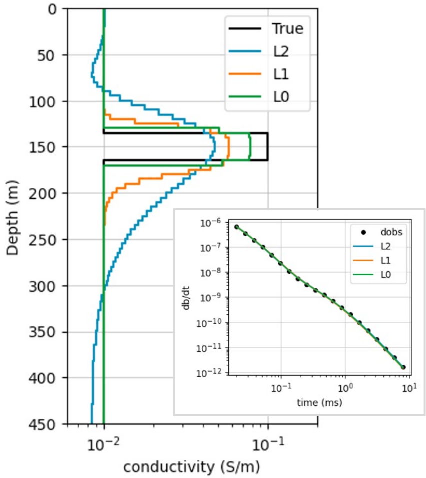
seawater intrusion: TDEM over Salinas Valley in 2017, 2019

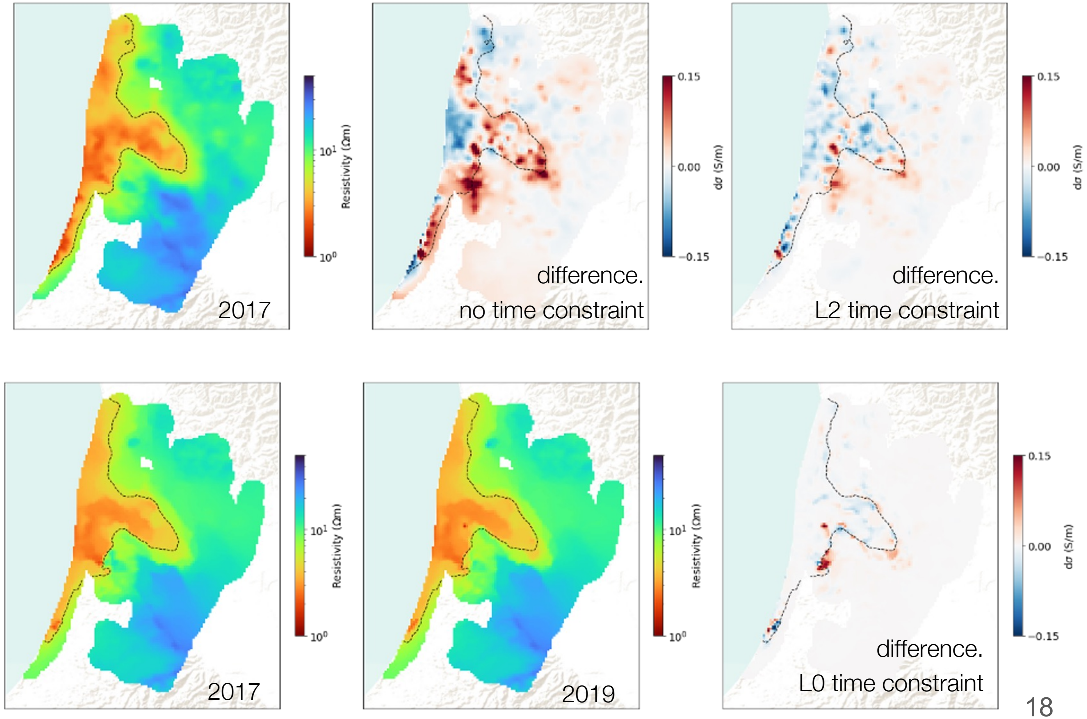
spatially constrained 1D inversions

- introduce time-constraint
- L_0 in space & time

$$\phi_m = \alpha_r \int_V \left|\frac{dm}{dr}\right|^{p_r} + \alpha_z \int_V \left|\frac{dm}{dz}\right|^{p_z} + \alpha_t \int_V \left|\frac{dm}{dt}\right|^{p_t}$$

lateral — vertical — time

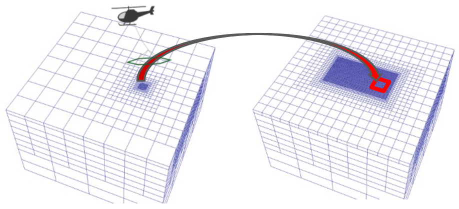




Kang & Knight, 2022; Kang BCGS talk 2022

electromagnetics: advanced capabilities

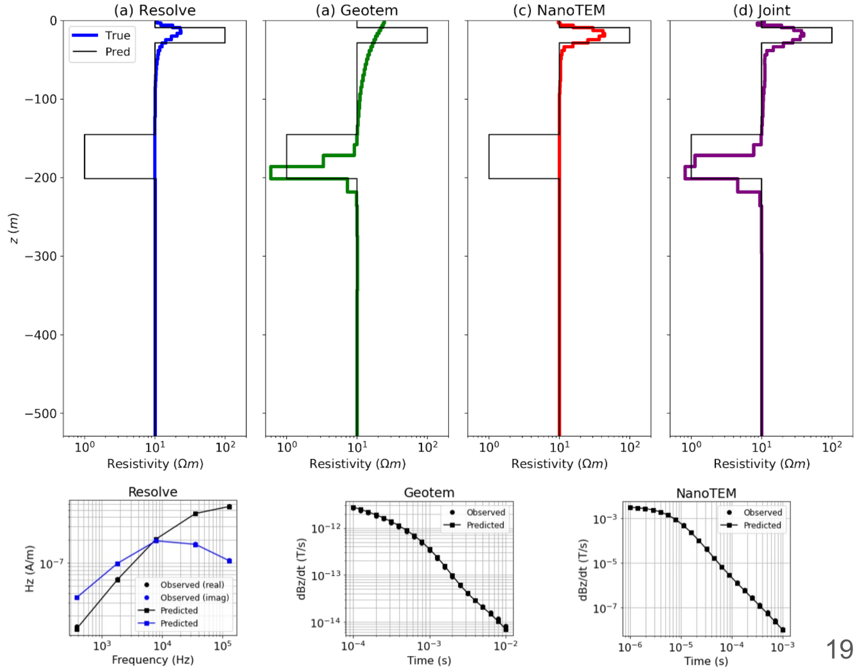
- Parallelization via domain decomposition



- Sparse / compact norms
- Parametric inversions
- Petrophysically & Geologically Guided Inversion (PGI)
- Joint inversions
    - same physical property
    - cross gradient, joint total variation, PGI

$$\phi(\mathbf{m}) = \underbrace{\phi_d^{\text{Resolve}} + \phi_d^{\text{Geotem}} + \phi_d^{\text{NanoTEM}}}_{\phi_d(\mathbf{m})} + \beta\phi_m(\mathbf{m})$$

```
phi_d = phi_d_resolve + phi_d_geotem + phi_d_nanotem
```



(Oldenburg et al, 2020)

now and the future

where we are…

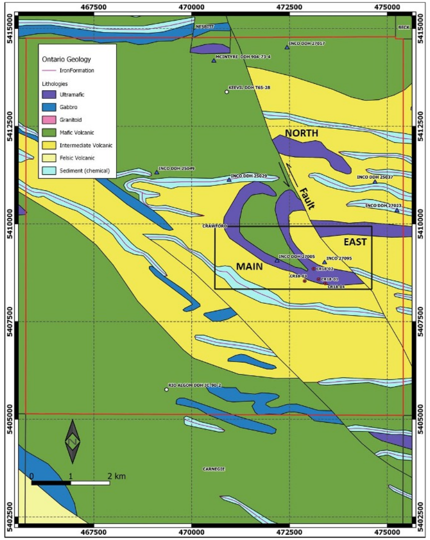
- 1D inversions: spatial constraints, sparse norms
- 3D inversions: fdem & nsem comparable to UBC fortran, tdem still some efficiency improvements
- joint inversions: cross-gradient, joint total variation, PGI

where we are going…

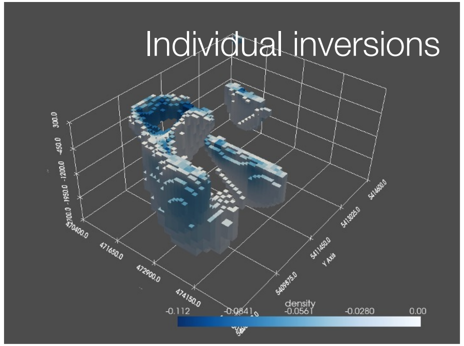
- tutorials, efficiency, simplifying code
- lots of research!

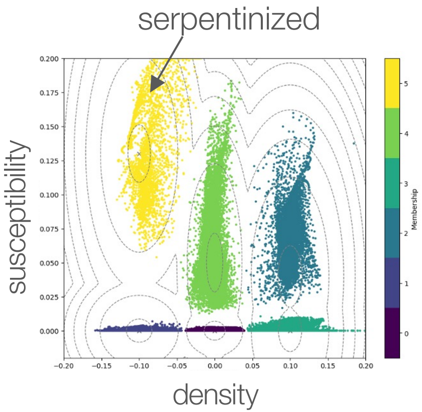
model

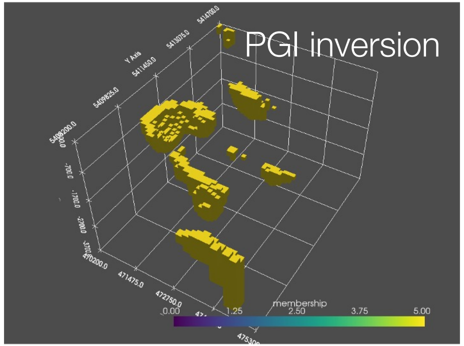
- research drives development
- partnerships with industry
- MIT license: use, adapt, commercialize



locating serpentinized rock: quasi-geology model from multiple geophysical data sets



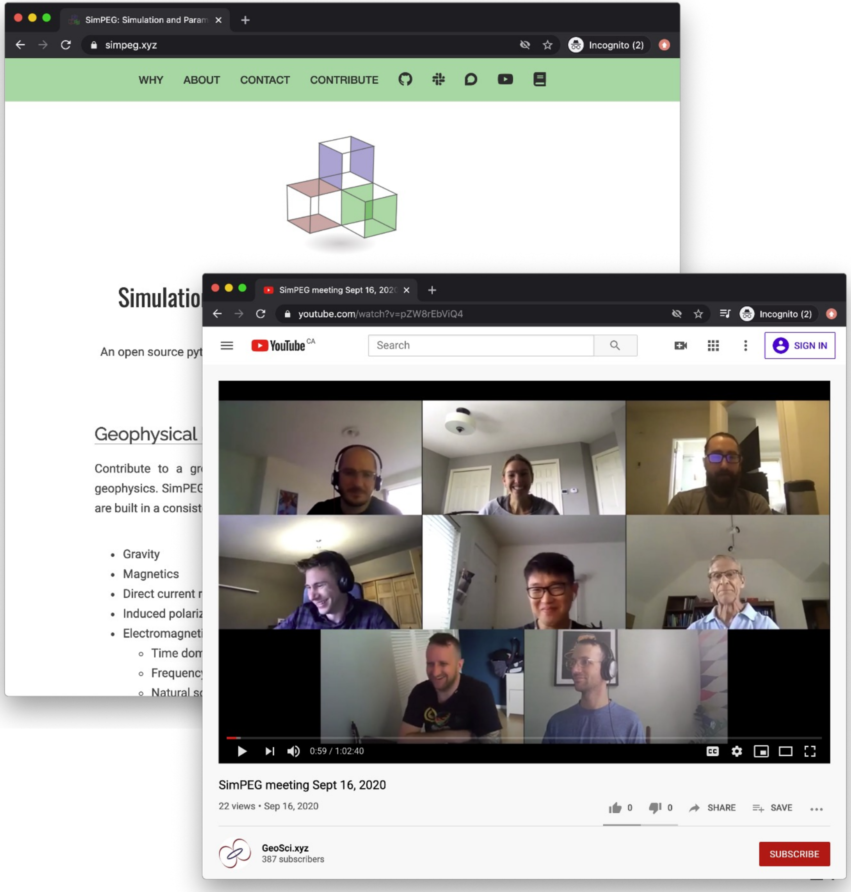




getting connected

```
conda install -c conda-forge simpeg
```

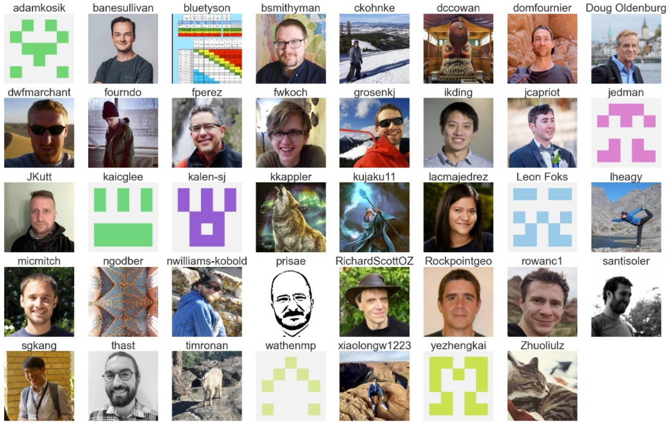
docs: docs.simpeg.xyz
newsletter: newsletter.simpeg.xyz
slack group: slack.simpeg.xyz



thank you!

email: lheagy@eoas.ubc.ca

slides: bit.ly/heagy-2023-aem



… and join us at 3DEM-7

https://3dem-7.geosci.xyz