Open Source

Tools?

- Want software for…
  - EM Simulations: FDEM, TDEM
  - Inversions:1D, 3D, parametric
- Intended use?
  - Production scale inversion
  - Survey design
  - Research
    - Inversion methodology
    - Understanding the physics
- Two options:
  - 1) use existing software
  - 2) develop your own tools

Option 1: use existing software

- Purchase license for software
- request academic license

Pros:	Cons:

• Reduce development overhead
• May have been tested / used for other data sets
• Often fit-for-purpose
	• Proprietary and not accessible
	  – Available as executable
	• Source code may be impenetrable:
	  – optimized and written by one (maybe two) people
	  – spaghetti code in a foreign language
	• Often fit-for-purpose: rigid and not easily extensible
	• Restrictive use (licensing)

Option 2: build your own

- Gather historic codes and merge/modify
- Start from scratch

Pros:	Cons:

• Flexibility
• Knowledge of structure / layout of the code
• Knowledge of which aspects are tested (and not)
• Extensibility
  – Can be designed with future development / research questions in mind
	• Daunting! Need strong background in scientific computing
	  – And software engineering skills
	• Time commitment
	  – Esp. if transferability is a priority
	• Challenging to publish
	• It is not regarded as a scientific contribution

Option 3??

Option 3: wish-list

- Collaboration
  - Development of software
  - Implementing and applying
- Development practices
  - Shared repository
  - Version control
  - Automated testing
  - User and developer documentation
  - Peer review of code
  - Issue tracking
  - Attribution for contributors
  - Licensing

Option 3: Open Source

- Collaboration
  - Development of software
  - Implementing and applying
- Development practices
  - Shared repository
  - Version control
  - Automated testing
  - User and developer documentation
  - Peer review of code
  - Issue tracking
  - Attribution for contributors
  - Licensing

Open source communities already doing this:

astropy
A Community Python Library for Astronomy

253 contributors

scikit learn
machine learning in Python

1,095 contributors

SciPy
Scientific Computing with Python

612 contributors

Tools in the open source ecosystem

python™

julia

R

jupyter

Read the Docs

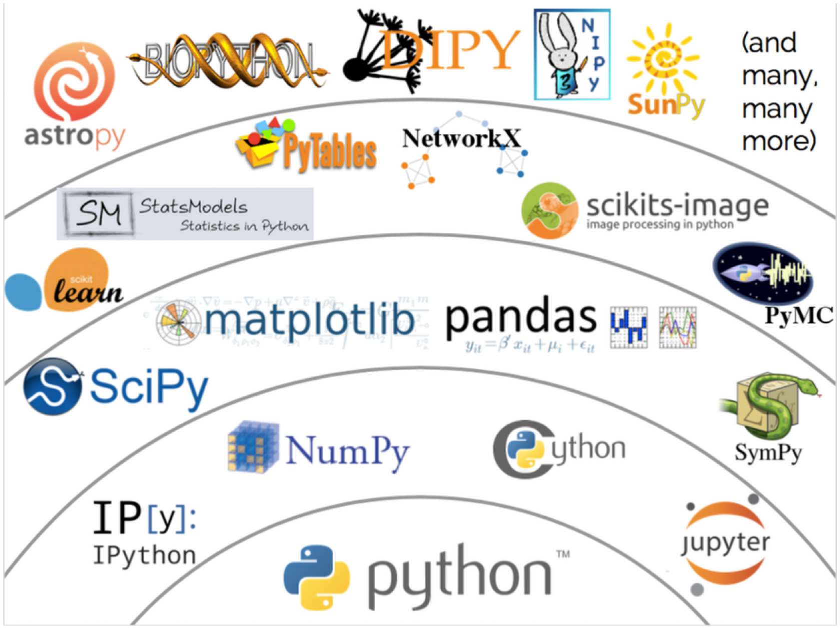
MIT

and many more …

Freely available modern languages

- Facilitates collaboration and reproducibility (by anyone!)
- Easier on-boarding of new users / graduate students
- Communities developing core-packages that can be re-used

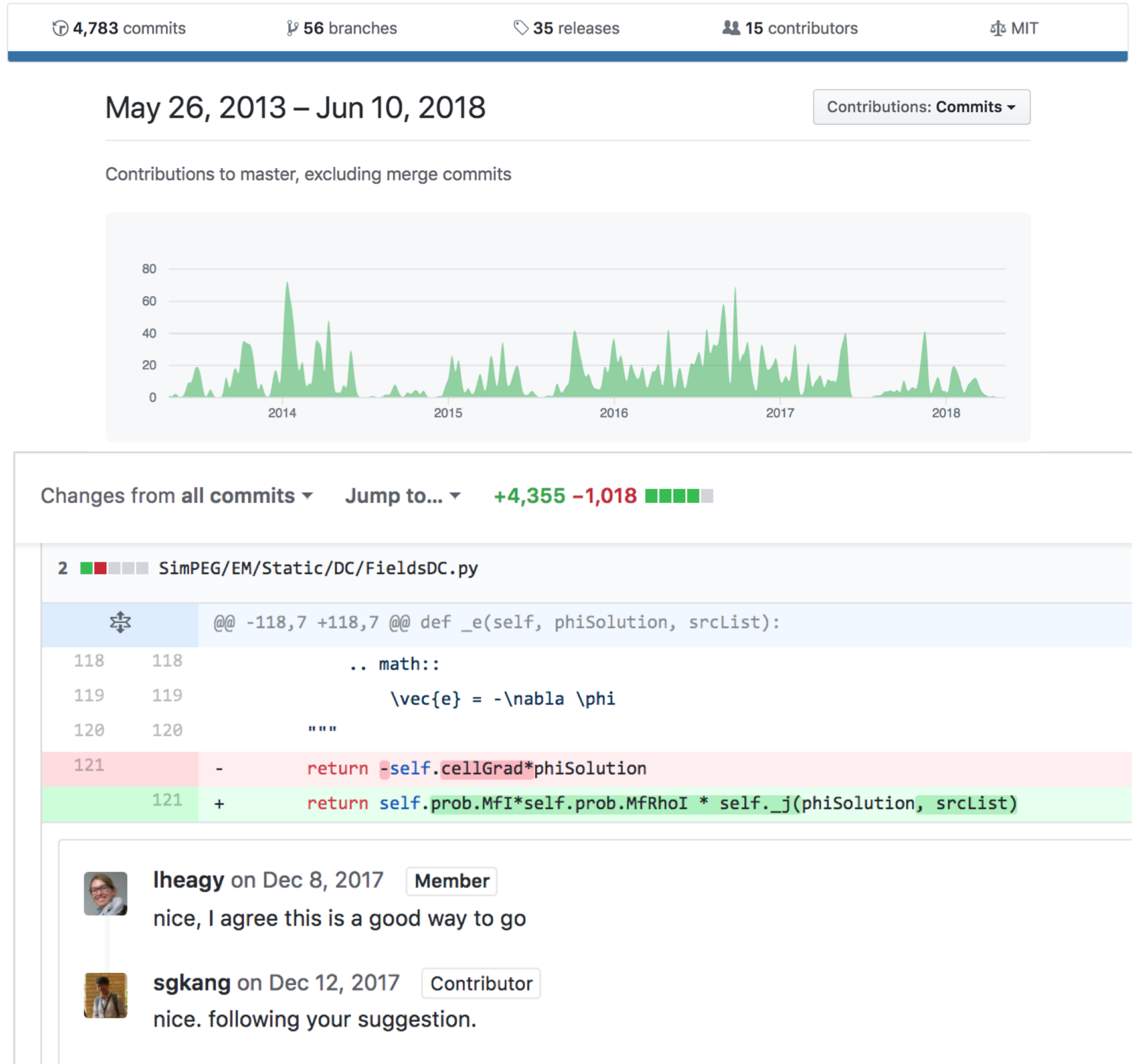
Scientific Python Ecosystem



From Jake VanderPlas "State of the Tools", SciPy 2015

GitHub

- Version control
- Issue tracking
- Code-reviews
- Attribution

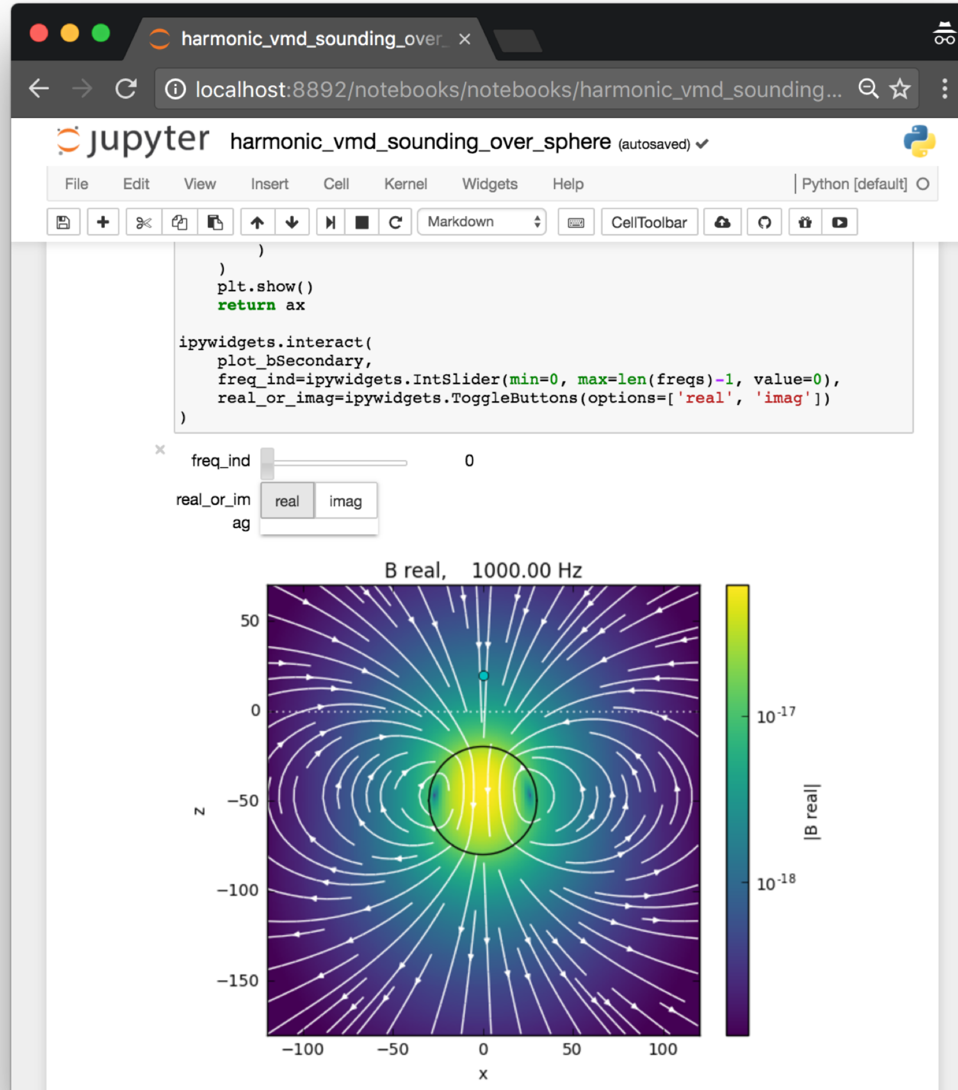


Jupyter notebooks

- Computational narratives / documented workflows
- Interactive computing
- Widgets
- Resources to distribute reproducible notebooks

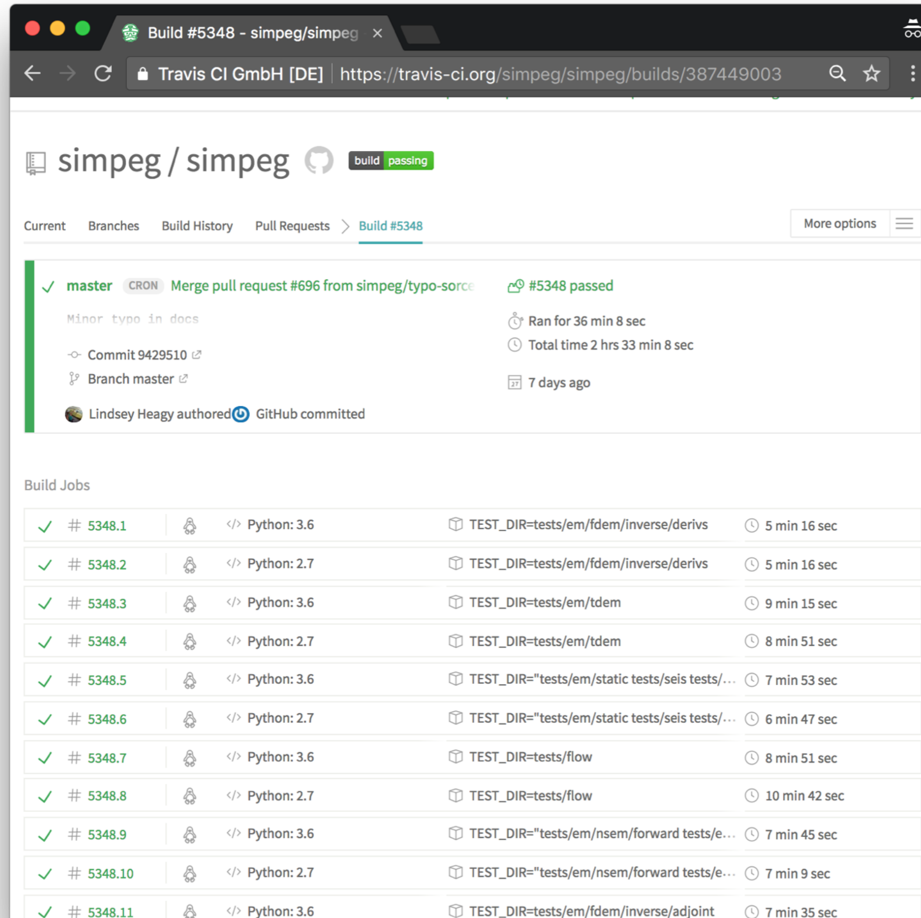
binder
(beta)

Microsoft Azure Notebooks



Testing

- TravisCI, CircleCI, …
- Automated testing
- Runs on every change to the codebase
- Test changes / new contributions before distributing

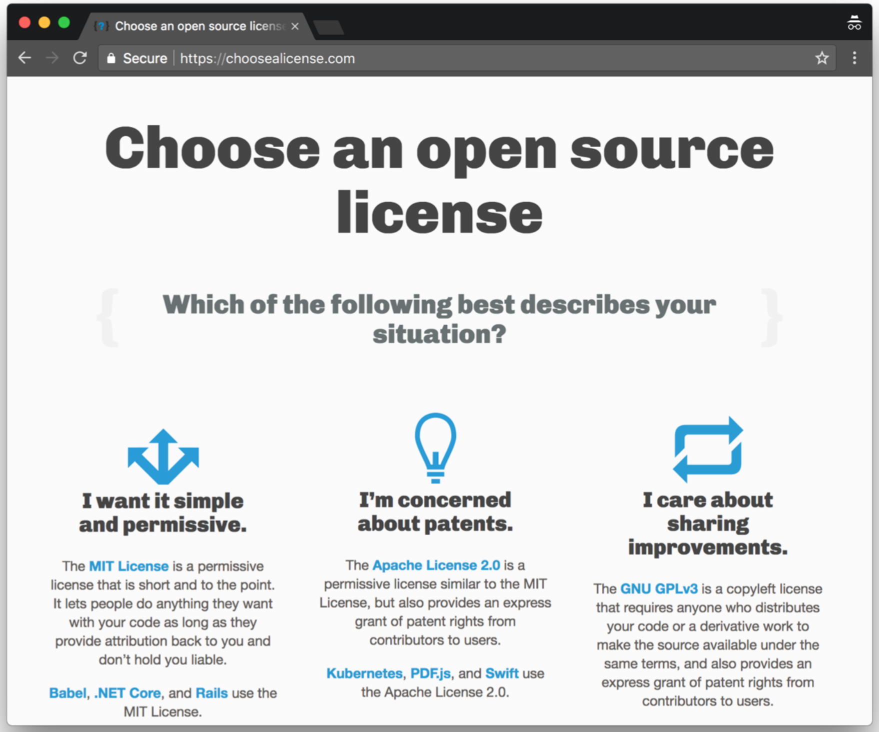


Documentation

- Read the Docs
- Documentation hosting
- Updated with changes to distributed version of code
- Searchable
- uses documentation strings within codebase

Licenses

- Specifies how others can use / adapt software
- Most Python projects use permissive licenses (MIT or BSD-3)
  - Facilitates collaboration
  - allows industry and academic use



https://choosealicense.com

Some big benefits

- Collaboration
- Ease of testing
- Useable software
- Problem solved faster
- Modification is easier
- Learn by reading existing codes
- Share analysis and results
- Reproducibility
- Attribution
- Continually improving set of tools in the open-source ecosystem
- Building a community

Sampling of modern open-source projects

- For EM
  - empymod
  - jInv
  - pyGIMLi
  - Fatiando
  - SimPEG
  - …

Sampling of modern open-source projects

- For EM
  - empymod
  - jInv
  - pyGIMLi
  - Fatiando
  - SimPEG
  - …

EM forward modelling of 1D VTI media with 3D EM sources in Python

Sampling of modern open-source projects

- For EM
  - empymod
  - jInv
  - pyGIMLi
  - Fatiando
  - SimPEG
  - …

Framework for PDE parameter estimation in Julia. Contains building blocks to assemble your own simulation and inversion.

Sampling of modern open-source projects

- For EM
  - empymod
  - jInv
  - pyGIMLi
  - Fatiando
  - SimPEG
  - …

Multi-method library for solving inverse and forward tasks related to geophysical problems in C++ and Python. 3D DC/IP, potential fields are highlights

Sampling of modern open-source projects

- For EM
  - empymod
  - jInv
  - pyGIMLi
  - Fatiando
  - SimPEG
  - …

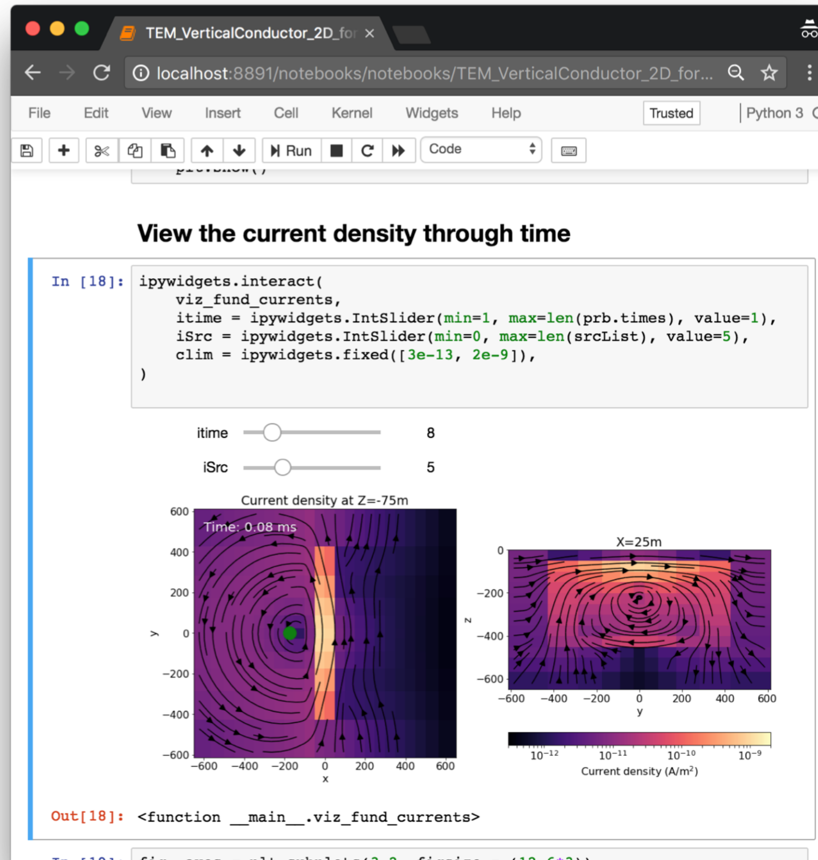
Simulation and gradient based parameter estimation in geophysical applications, including DC/IP, EM. Written in Python

Sampling of modern open-source projects

- For EM
  - empymod
  - jInv
  - pyGIMLi
  - Fatiando
  - SimPEG
  - …
- They differ in objectives, capabilities, structure, interactivity, license, and language

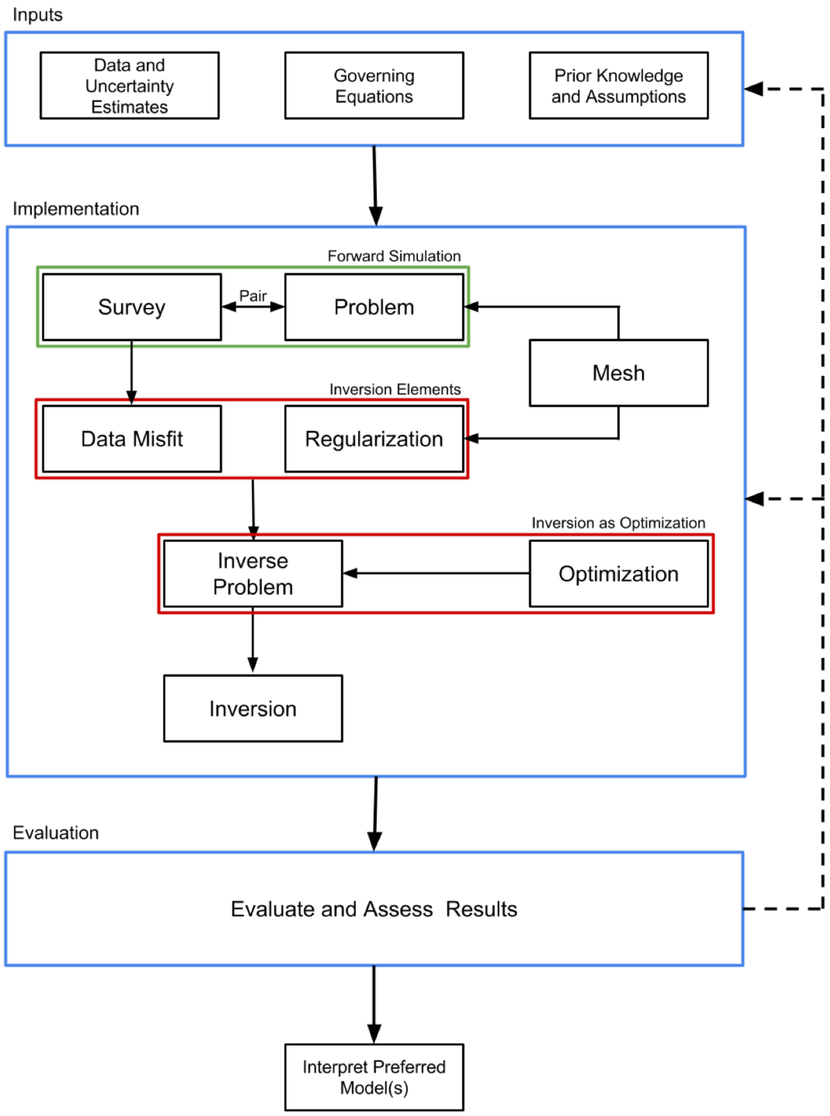
simpeg

- Modular framework for simulation and inversion of geophysical data
  - gravity, magnetics, vadose flow, DC/IP, FDEM, TDEM
- Open source
- Written in Python
- Specific to electromagnetics
  - Quasi-static Maxwell
  - Tensor, OcTree, Curvilinear and Cylindrical meshes
  - Easily visualize fields, fluxes, charges

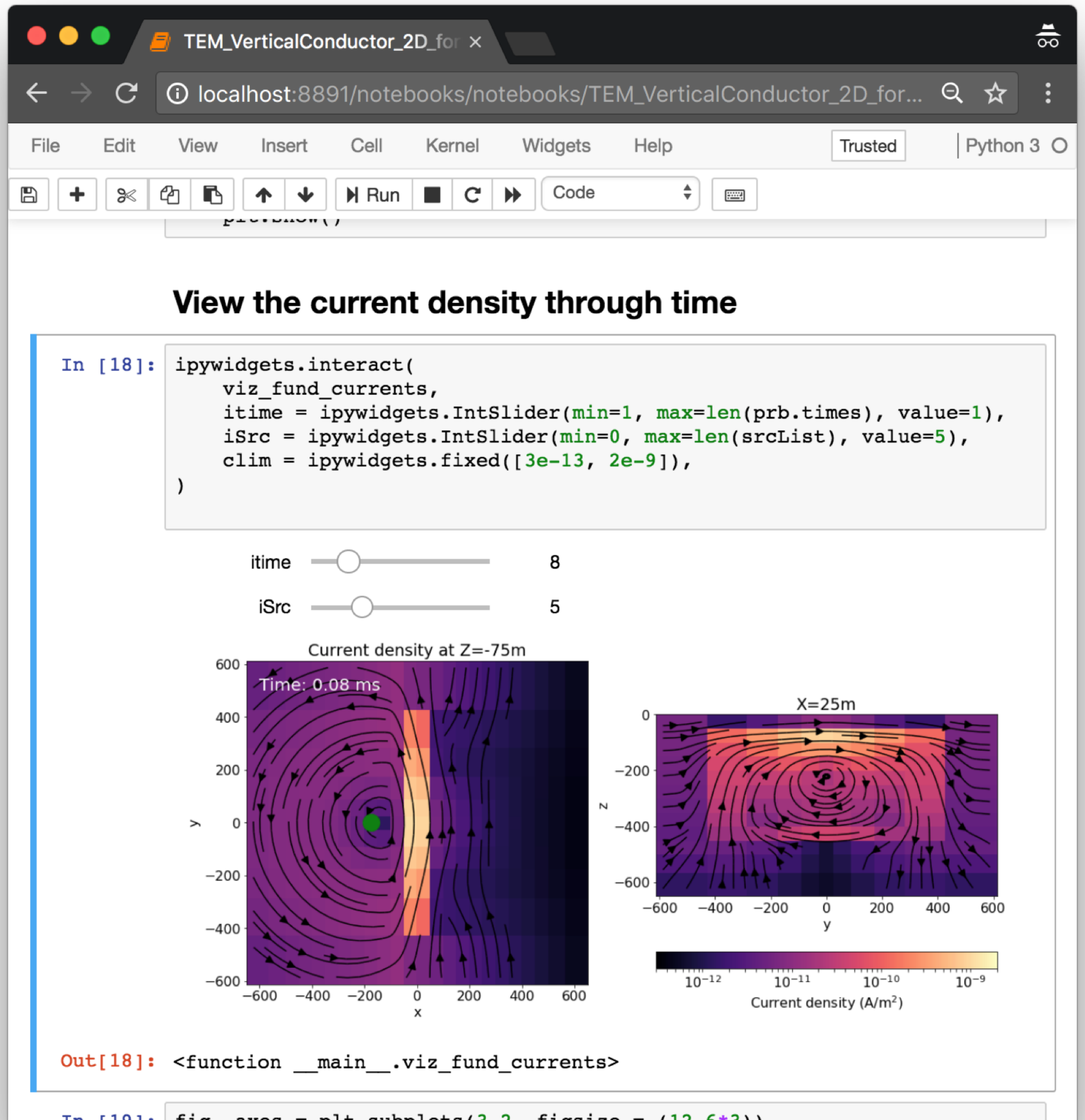


http://simpeg.xyz

SimPEG: Inversion framework

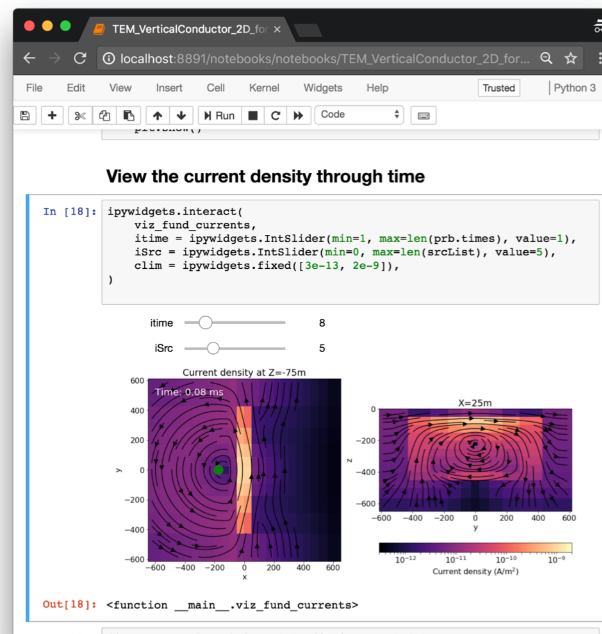


Notebook Demo: Plate in a Halfspace



Summary

- Need for Open source
- Open source eco-system
- SimPEG
- Jupyter notebooks and demos
  - Forward simulation
  - Inversion
- Benefits
  - Collaborating
  - Reproducibility,
  - Learning

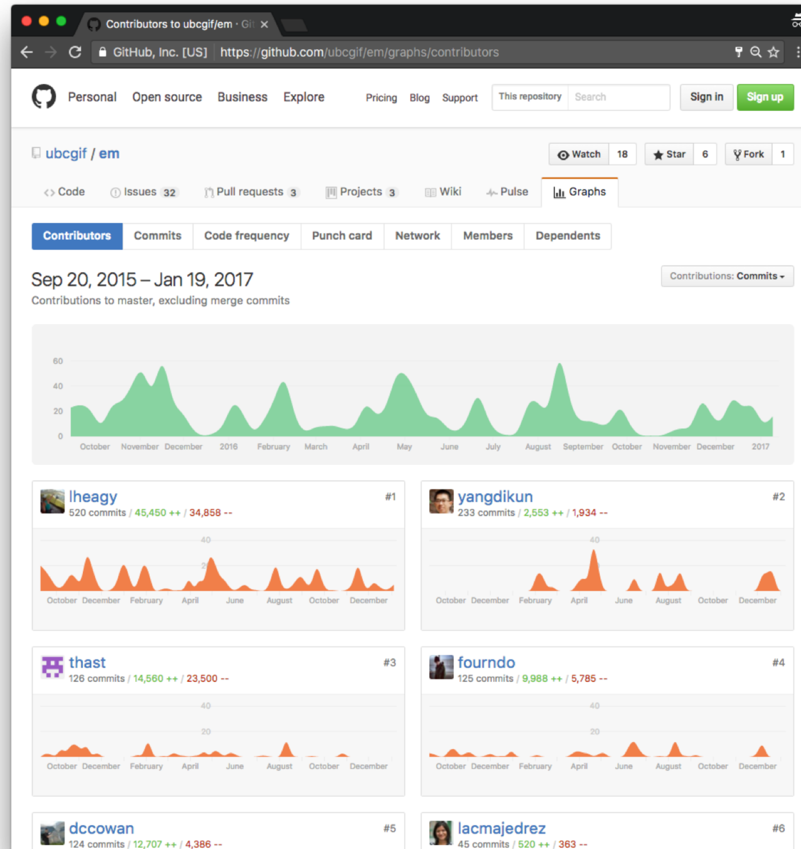


Web and Open Source Resources

- Open source development: Software and resources
  - Collaborate
  - Share
  - Test changes
  - Interactive computing

simpeg

Simulation and Parameter Estimation in Geophysics
http://simpeg.xyz



Github
versioning, collaborating

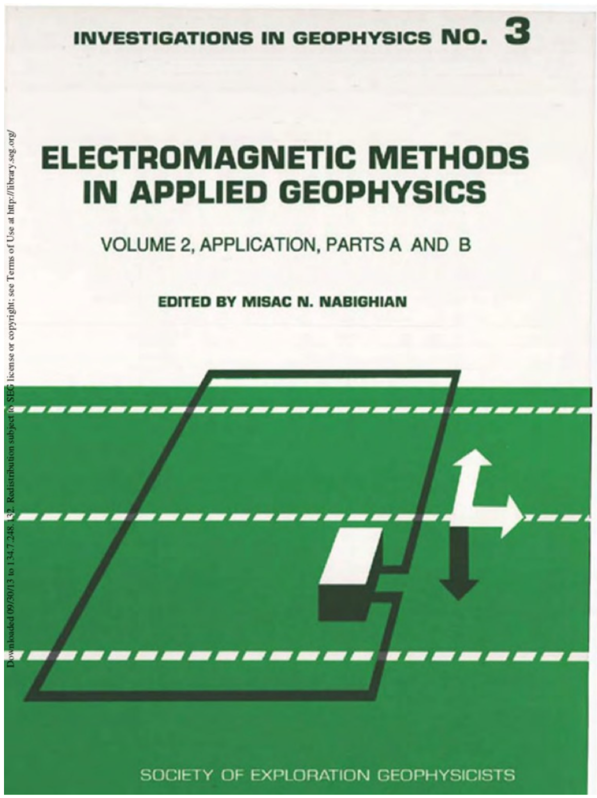
Travis CI
testing, deploy

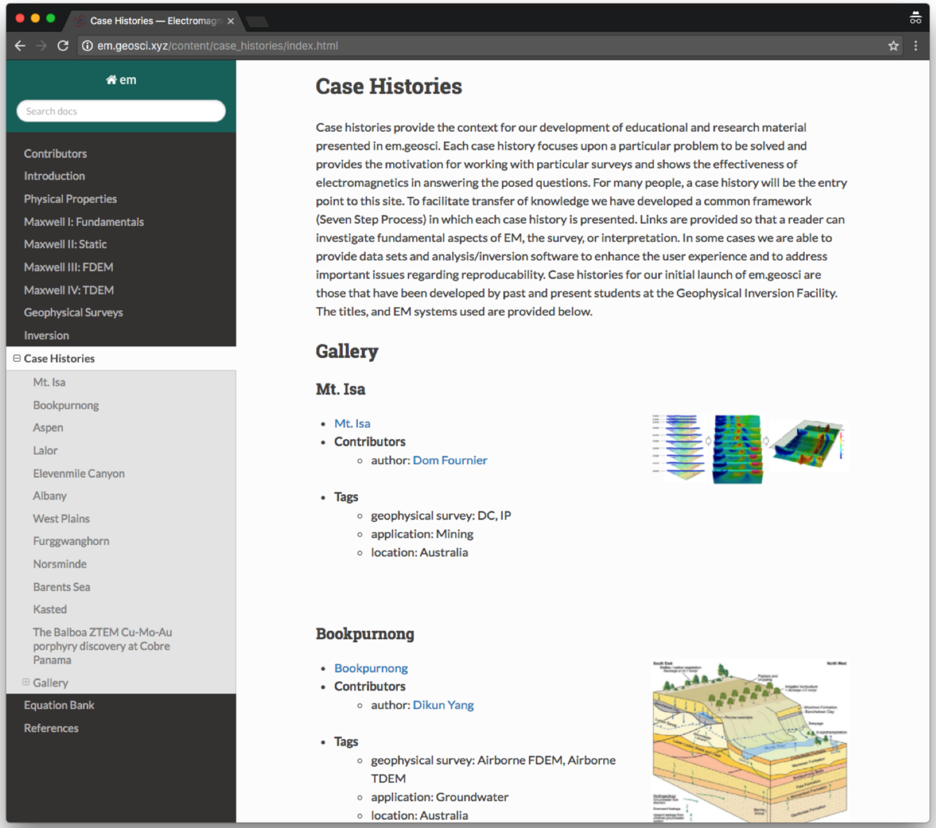
Jupyter
interactive computing

Creative Commons
licensing, reuse

Python
computation

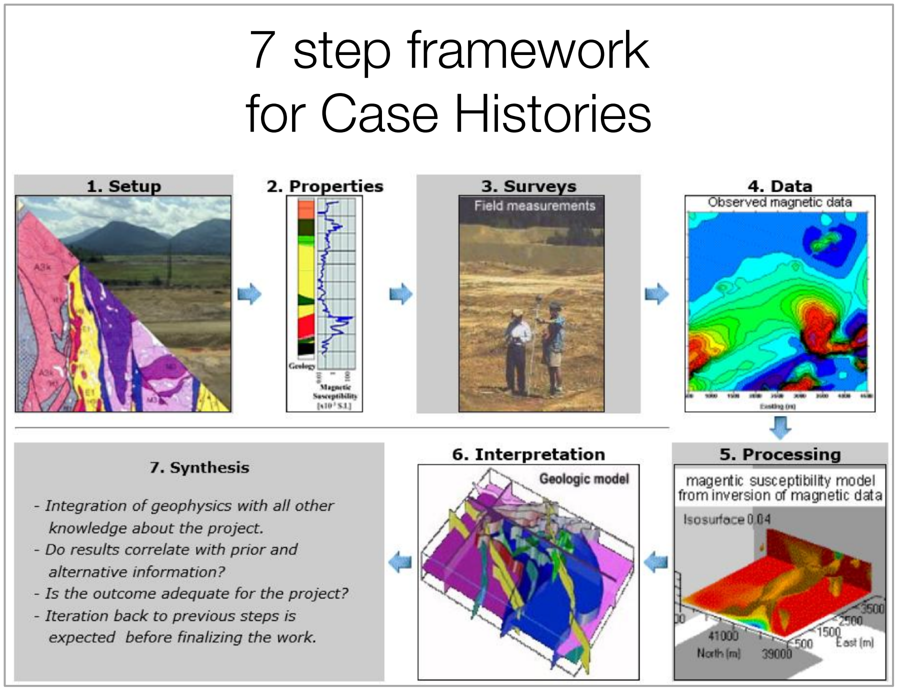
Resources: EM.geosci

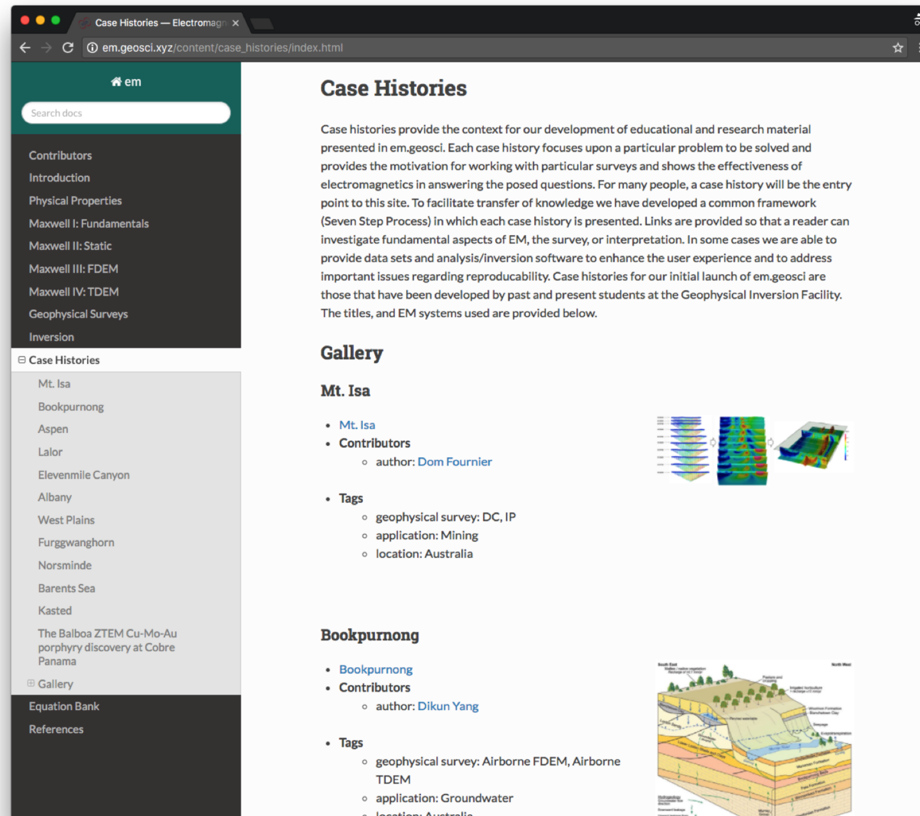




http://em.geosci.xyz

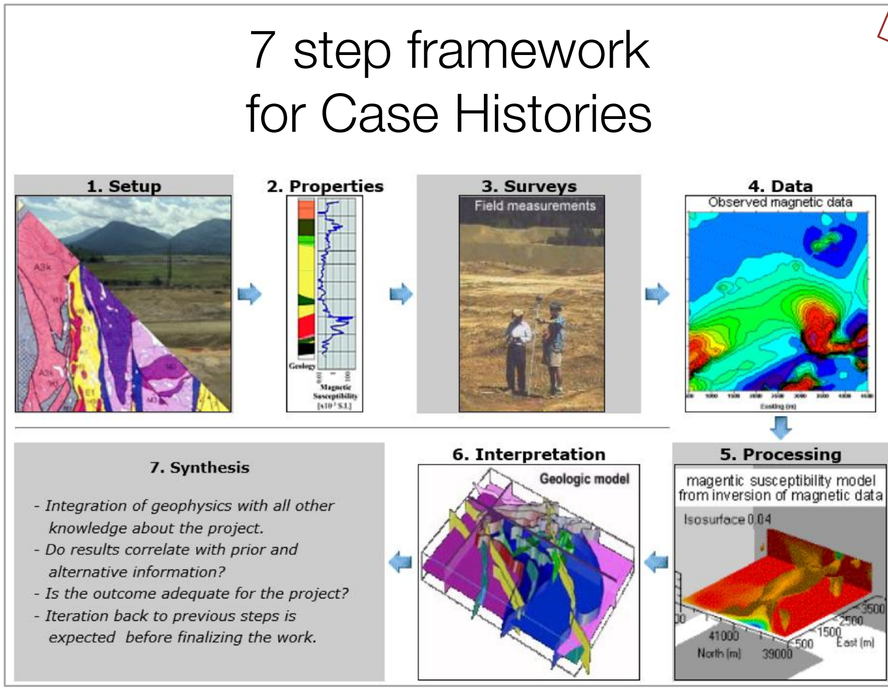
Resources: EM.geosci





http://em.geosci.xyz

Resources: EM.geosci

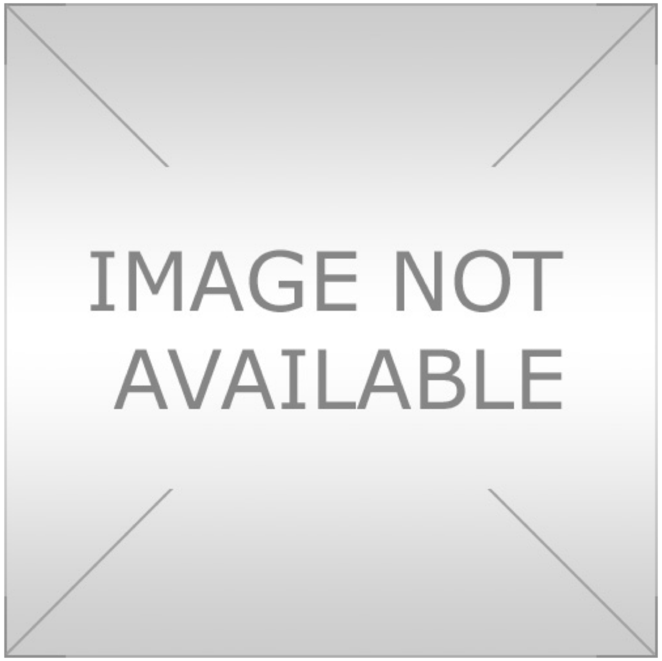


http://em.geosci.xyz

Why Apps

$$\nabla \times \mathbf{e} = -\frac{\partial \mathbf{b}}{\partial t}$$

$$\nabla \times \mathbf{h} = \mathbf{j} + \frac{\partial \mathbf{d}}{\partial t}$$

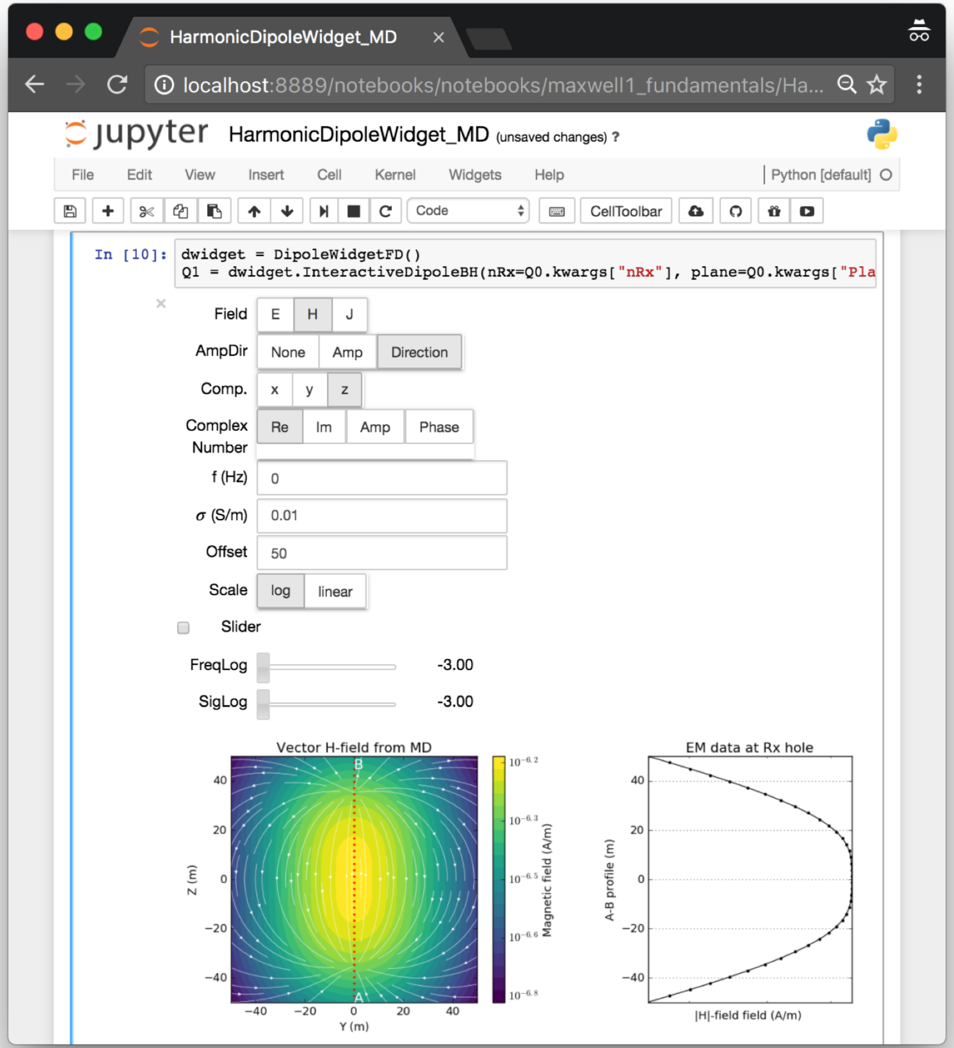


http://em.geosci.xyz/apps.html

Why Apps

$$\nabla \times \mathbf{e} = -\frac{\partial \mathbf{b}}{\partial t}$$

$$\nabla \times \mathbf{h} = \mathbf{j} + \frac{\partial \mathbf{d}}{\partial t}$$



http://em.geosci.xyz/apps.html

Goals

- Motivation (questions supplemented with images)
  - We've seen Aarhus workshop; great examples; much information obtained with LCI;
  - What happens if the situation is 3D
  - how do the fields behave; do we need to work in 3D
  - If we decide to work in 3D what are we up against;
  - Forward modelling
  - Inversion

- Open Source plays a critical role in achieving goals
  - Talk more about this on Tuesday
  - Today: illustrate the benefits and needs of adopting this paradigm

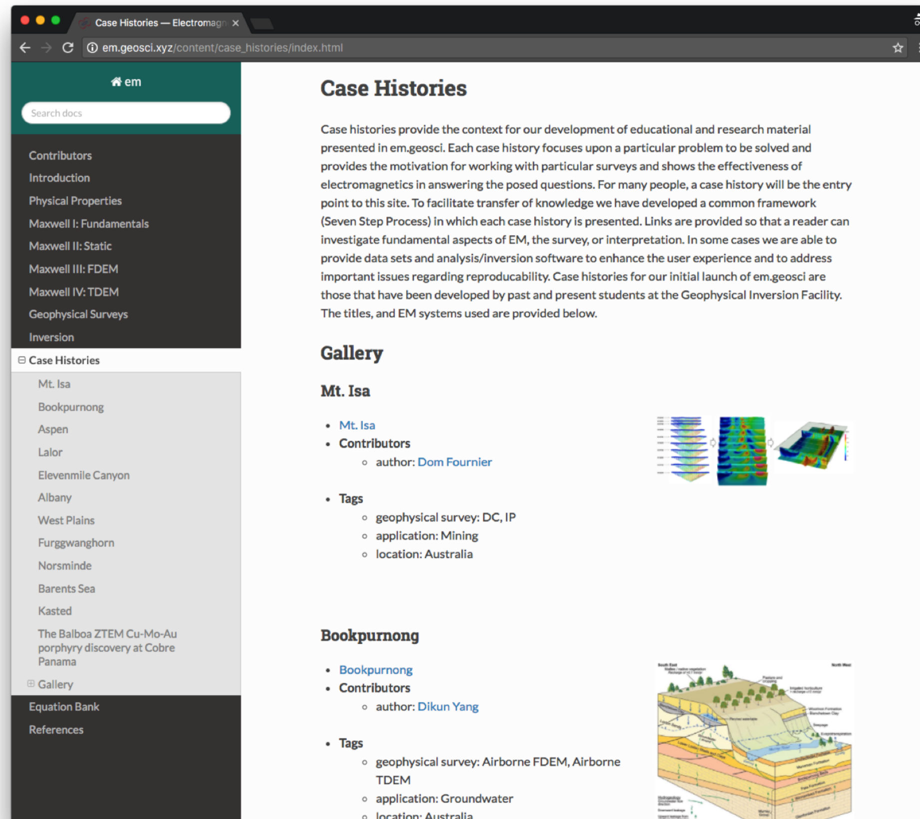
Connecting & Contributing

- Today: Slack
  - http://slack.geosci.xyz/

Join **GeoSci** on Slack.
3 users online now of **9** registered.

you@yourdomain.com

GET MY INVITE



- Contributing:
  - EM GeoSci
    - Case histories
    - Content
  - SimPEG
    - Software