Linking open source tools for geophysical simulation and inversion in rugged topographies

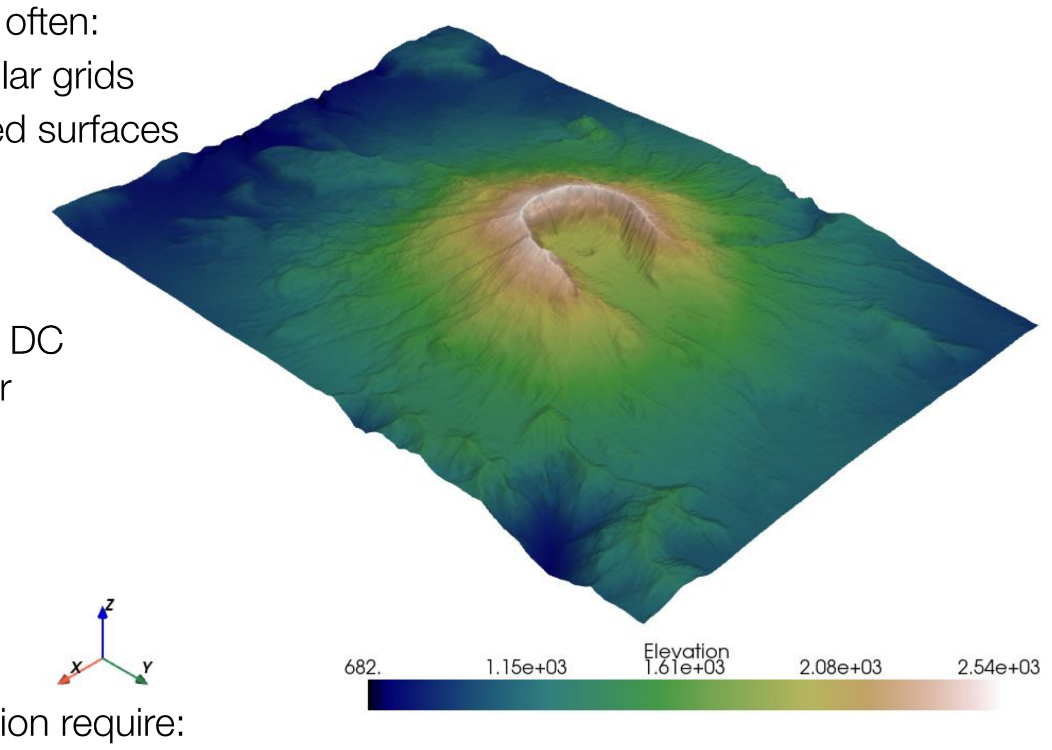
Joseph Capriotti[1*], Johnathan Kuttai[1], Dom Fournier[2], Lindsey J. Heagy[1], and the SimPEG Team

[*] josephrcapriotti@gmail.com
[1] University of British Columbia Geophysical Inversion Facility,
[2] Mira Geoscience Ltd.

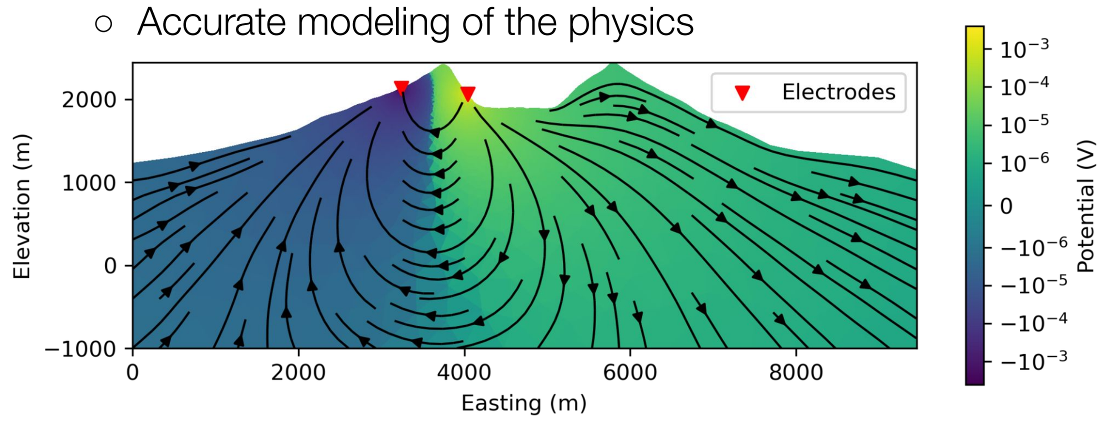
1 Motivation

- Geophysical data are often:
  - Collected on irregular grids
  - Collected on rugged surfaces

Consider collecting DC Resistivity data over Mt. Saint Helens



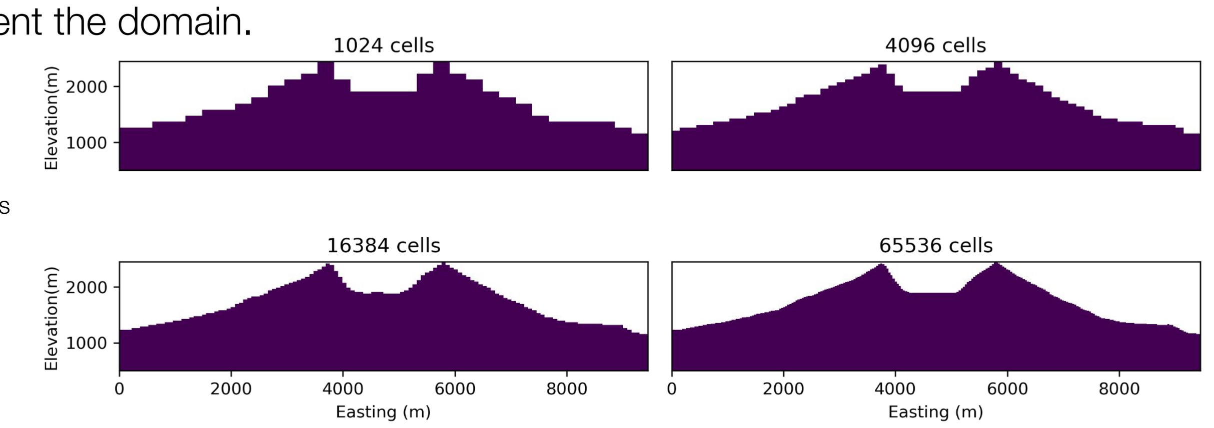
- Simulation and Inversion require:
  - Discretized domain
  - Accurate modeling of the physics



DC Potential due to a dipole shown with electric field lines. Notice the distortions in the flow from the normal dipole response due to topographic effects.

- Simple meshes require large amounts of cells to accurately represent the domain.

2D Regular mesh representing a cross section through Mt. Saint Helens



- Increasingly complex meshes are:
  - Increasingly difficult to construct
  - Require special visualization tools
- Many open source tools are available to tackle individual pieces of the entire problem

2 Objectives

1: Link open-source resources to:

- Create terrain conforming meshes
- Construct numerical PDE operators
- Accurately simulate geophysical methods
- Perform inversion
- Make visual interpretations (pyvista.org)

2: Provide workflows to follow for designing meshes and using them in geophysical inversions.

3 Mesh Design

Design goals

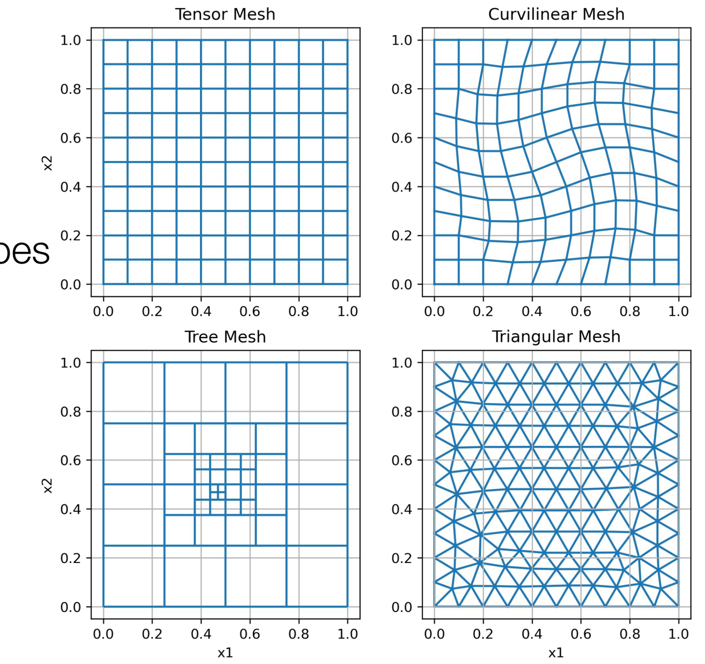
1: Fine cells near topography in the area of the survey, with less need for accurate representation at far distances.

2: Fine cells radially away from electrode locations, such that the rapidly changing potential field is accurately modeled.

discretize

discretize.simpeg.xyz

- Python package
- Support for several mesh types
  - Tensor
  - Cylindrical
  - Curvilinear
  - Octree
  - Quadtree
  - Triangular
  - Tetrahedral



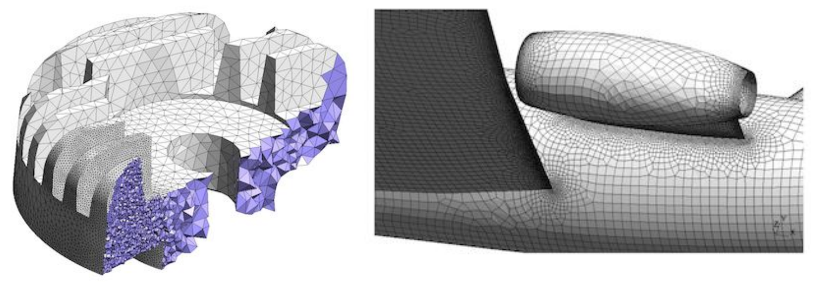
- Mesh creation with tree mesh refinement options:
  - **Balls** - (electrode locations)
  - Lines
  - Boxes
  - Tetrahedrons
  - **Surfaces** - (along topography)

Octree Mesh - 255k nodes, 300k cells

- Fast refinement operators
- Quantized representation of elevation
- Electrode locations must be interpolated

Gmsh

gmsh.info

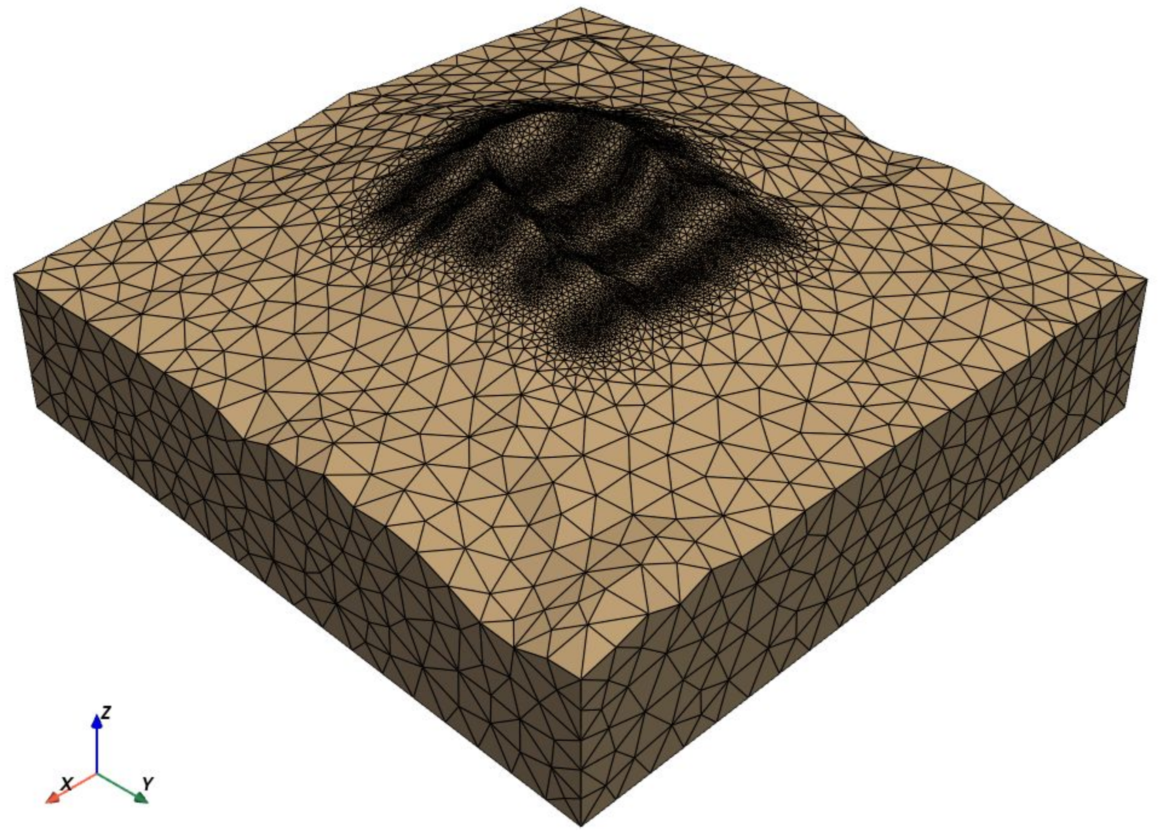


- 2D and 3D mesh generator for triangular and quadrilateral meshes
- Creates geometric entities, and subsequently meshes these.
- Python interface to compiled C++ codes

- Mesh creation steps:
  - Represent topography as a surface
  - Directly embed points at electrode locations
  - Specify mesh size as a function of distance from electrode locations
- Output linked to tetrahedral mesh in `discretize`

Tetrahedral Mesh - 48k nodes, 250k cells

- Fewer nodes needed (smaller system matrices)
- Elevation satisfied at all surface nodes
- Electrode locations guaranteed to fall on nodes



4 Simulation and Inversion

discretize

- Weak-form finite volume operators on all meshes
  - Nodal Gradients
  - Edge based inner products
  - Interpolation functions
- Code looks like the math
  - DC continuous PDE:

$$\nabla \cdot \sigma \nabla \phi = -I\delta$$

  - DC finite volume discrete form:

$$\mathbf{G}^T \mathbf{M}_{e,\sigma} \mathbf{G} v = e$$

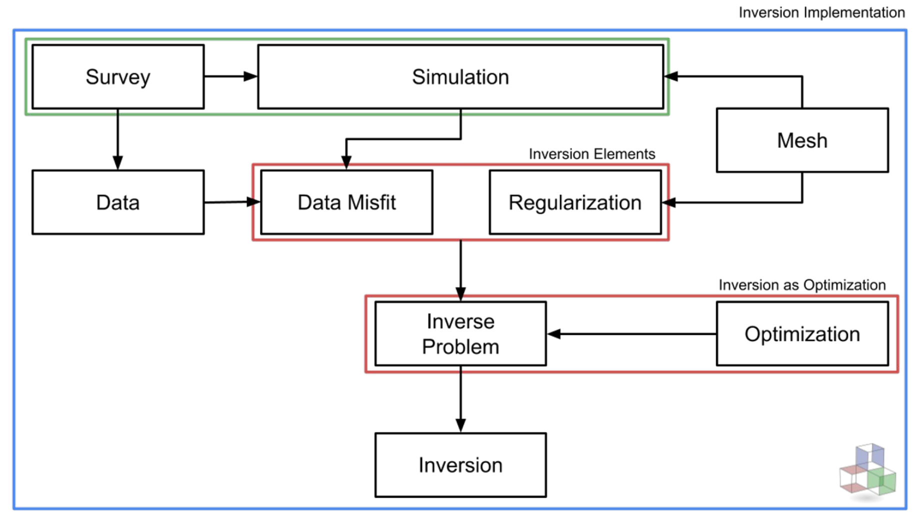
  - Python code:

```
G = mesh.nodal_gradient
Me_sig = mesh.get_edge_inner_product(sigma)

A = G.T @ Me_sig @ G
e = mesh.get_interpolation_matrix(source.location).sum(axis=0)

v = Solver(A) @ e
```

- Provides derivative operations with respect to mesh physical properties.

simpeg

- Simulation and inversion framework for many geophysical methods.

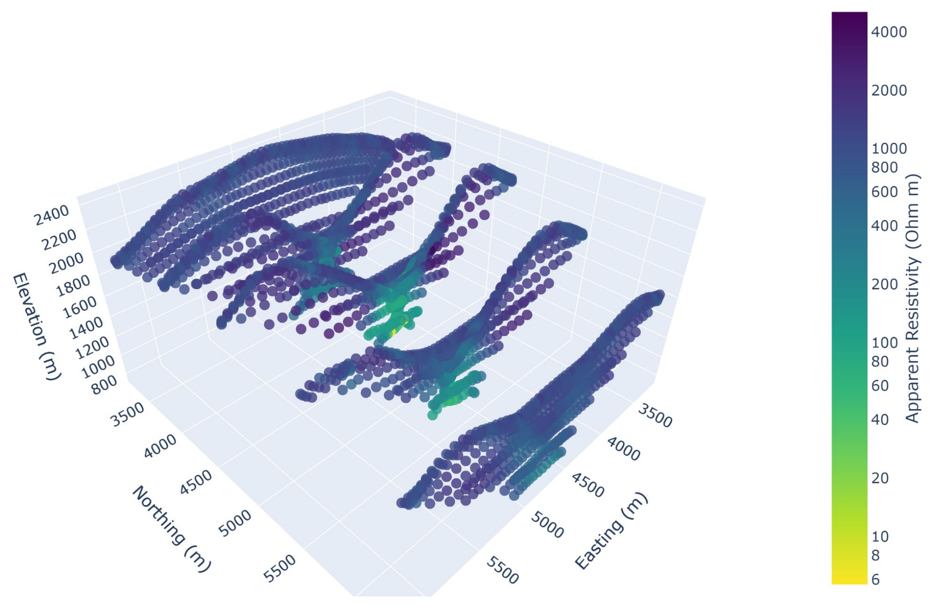


- Readily implemented geophysical methods:
  - Mesh independent definitions
  - Jacobian matrix operations
- Mesh independent spatial derivative regularization

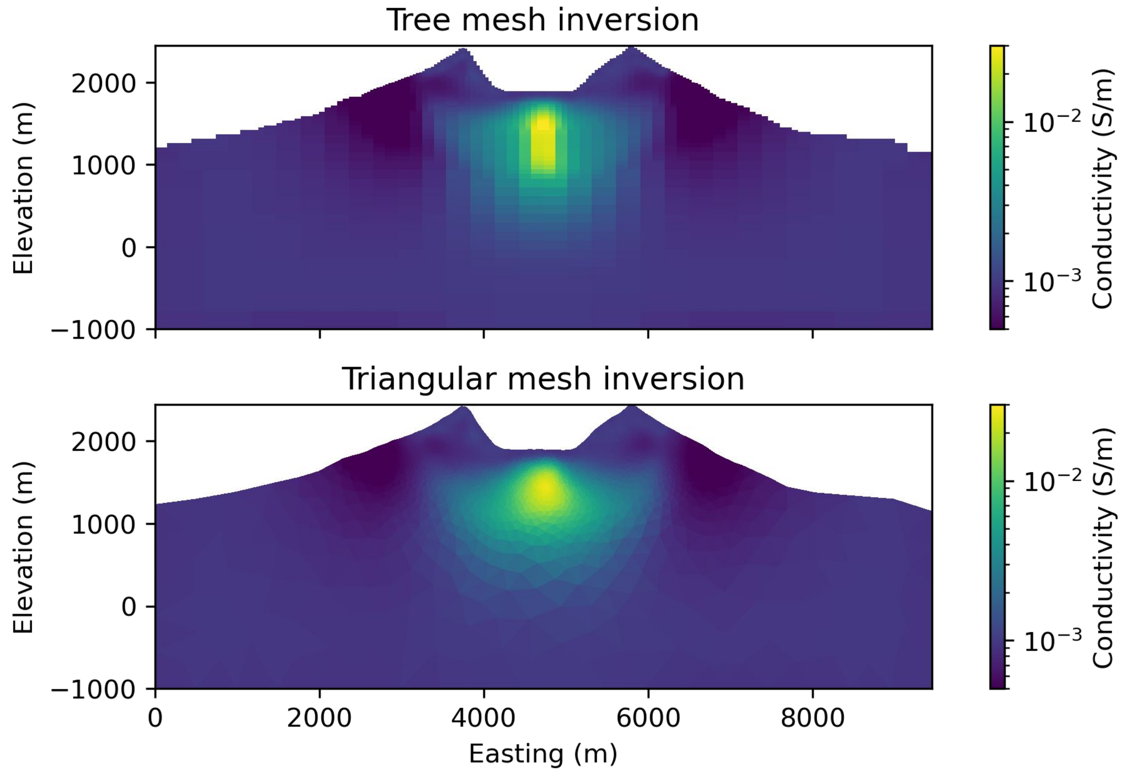
$$\phi_m(m) = \int_V |\nabla m|^2 \partial v$$

- Provides optimization routines

- We simulate resistivity data for a high conductivity anomaly centered below the crater



- We invert the simulated data on both the tree mesh and triangular mesh



5 Summary

- `discretize` and `Gmsh` used to design structured and unstructured meshes
- `discretize` constructs finite volume operators
- `simpeg` to perform simulation and inversion
- `pyvista` for visualization

- Notebook workflows are available at: simpeg-terrain.curve.space

Join us and get involved!

- Attend our weekly meetings

simpeg.xyz

github.com/simpeg

slack.simpeg.xyz

for Python users

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

References:

Cockett R., S. Kang, L. J. Heagy, A. Pidlisecky, and D. W. Oldenburg, 2015, An Open Source Framework for Simulation and Gradient Based Parameter Estimation in Geophysical Applications. Computers and Geosciences 85(A) pp. 124-154.

Geuzaine, C. and J.-F. Remacle, 2009, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331.

Sullivan, C. B., and A. Kaszynski, 2019, PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK): Journal of Open Source Software, 4(37), 1450.