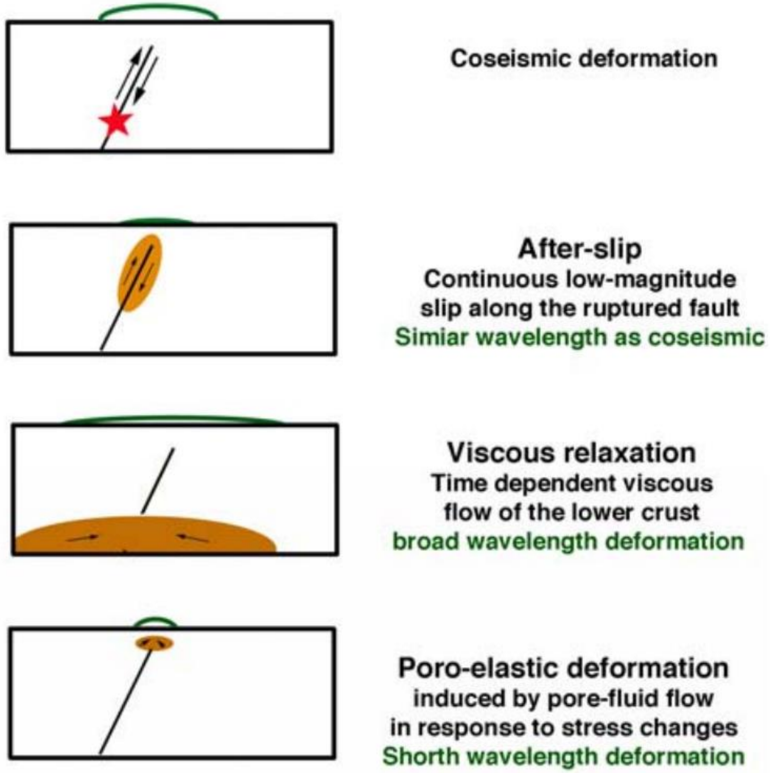
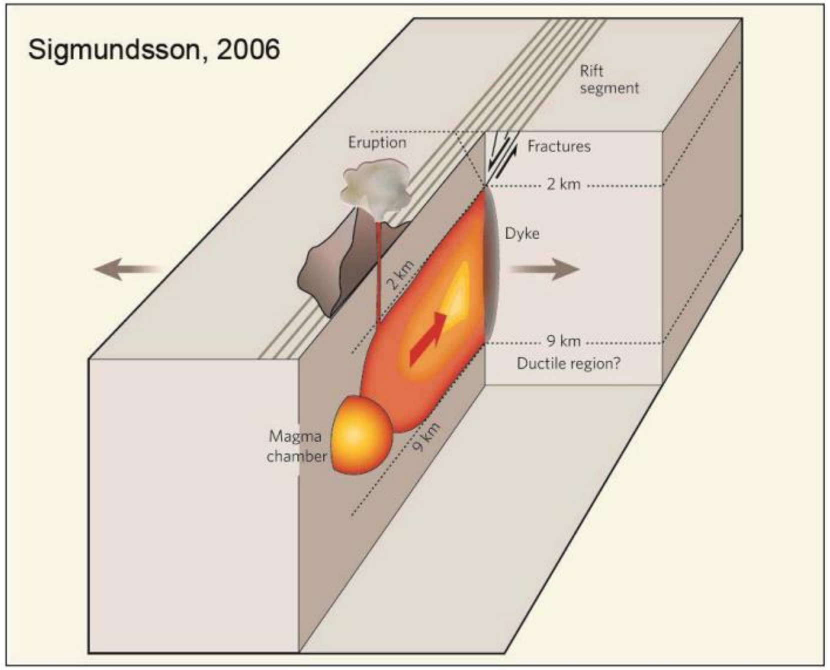
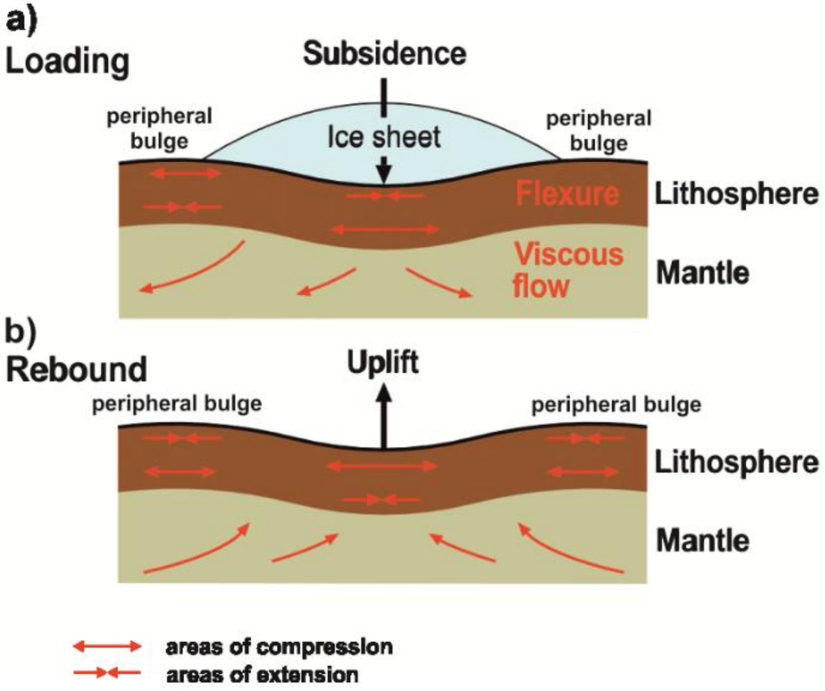
An introduction to geophysical modeling of geodetic data

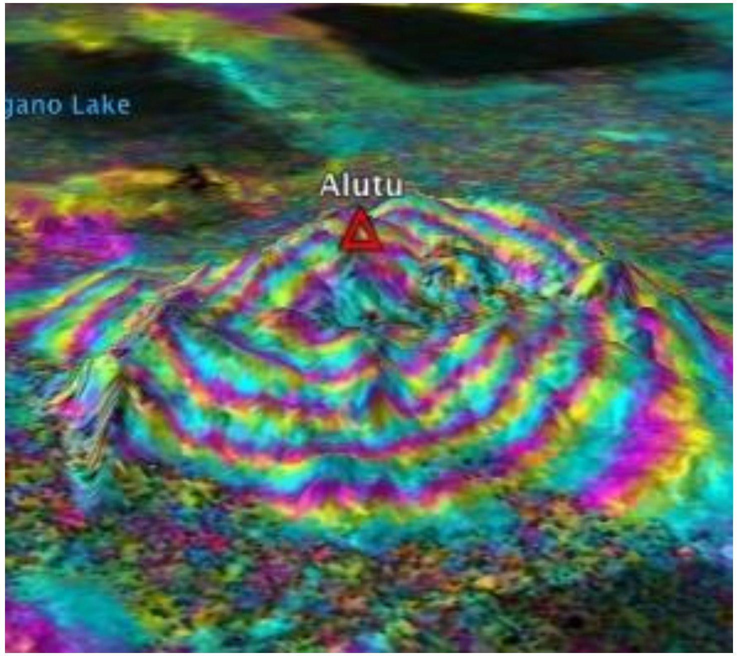
Gareth Funning, University of California, Riverside

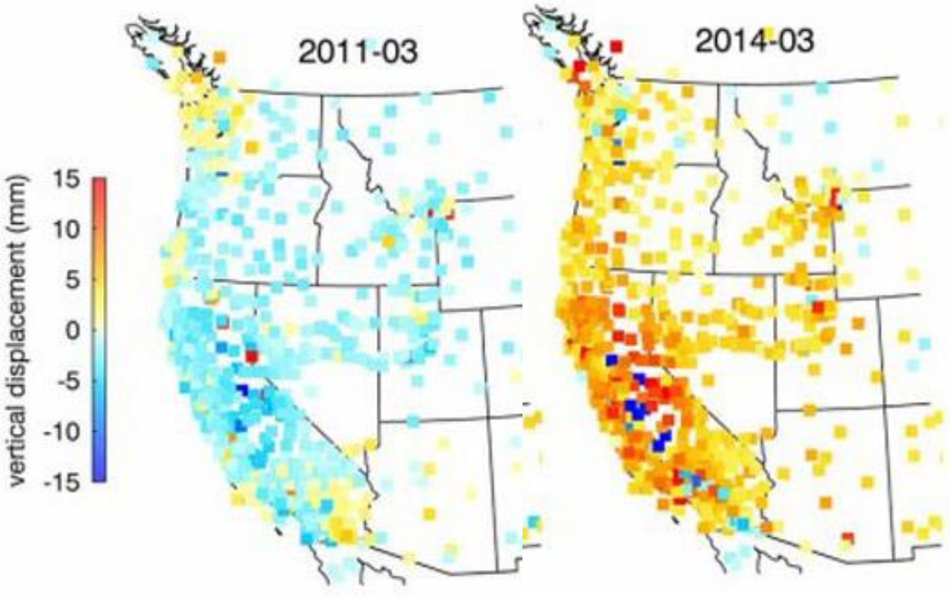
By **geophysical modeling**, we mean using idealized representations of the Earth to gain insight into its properties and processes

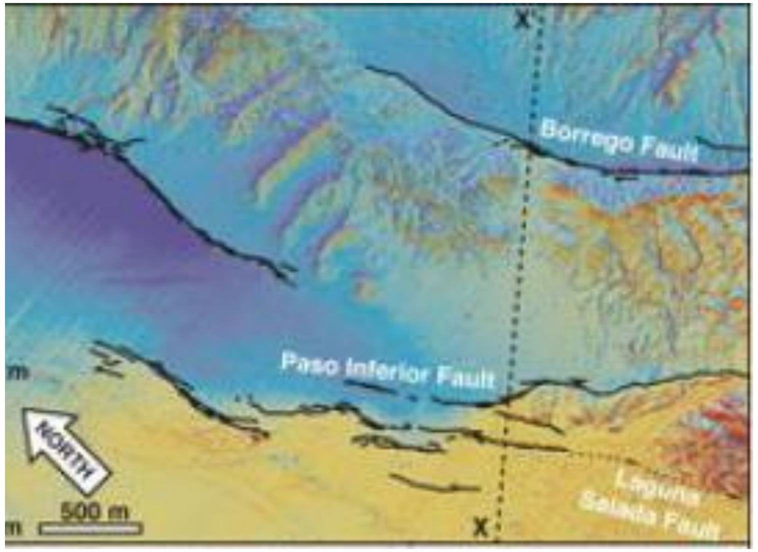






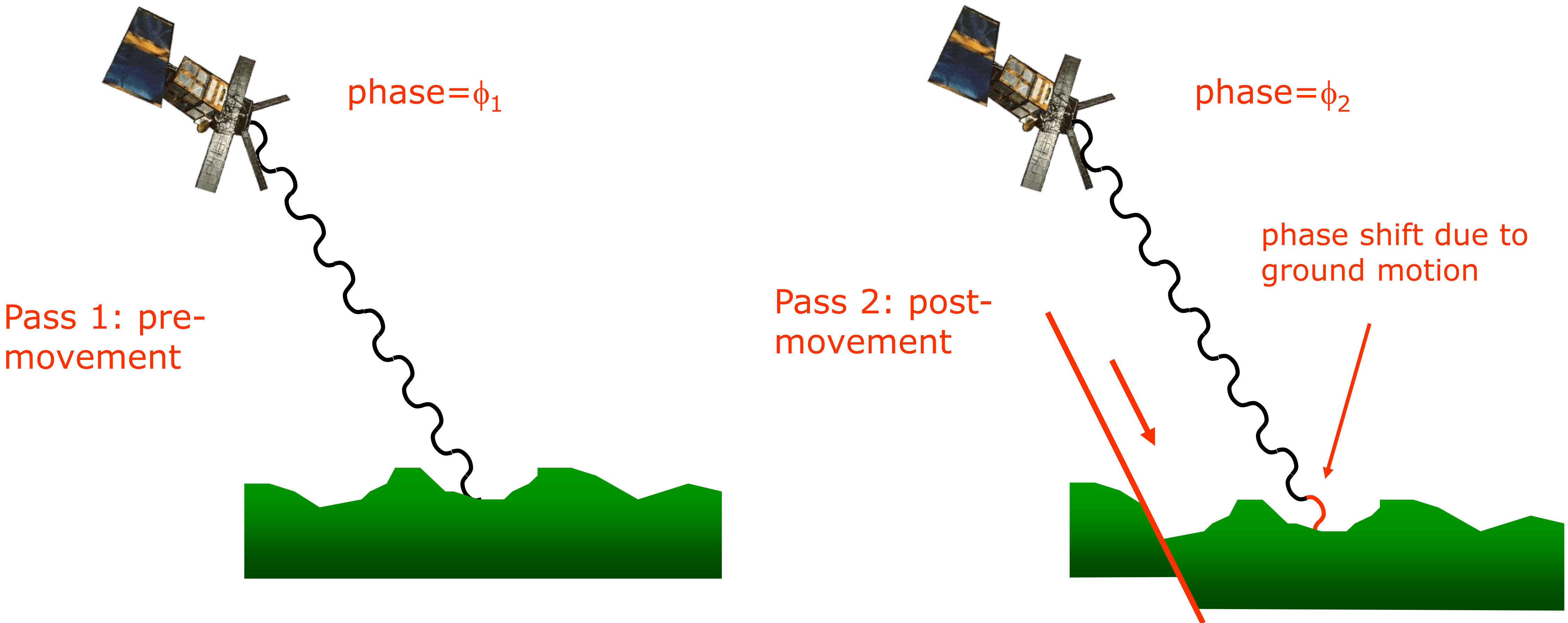




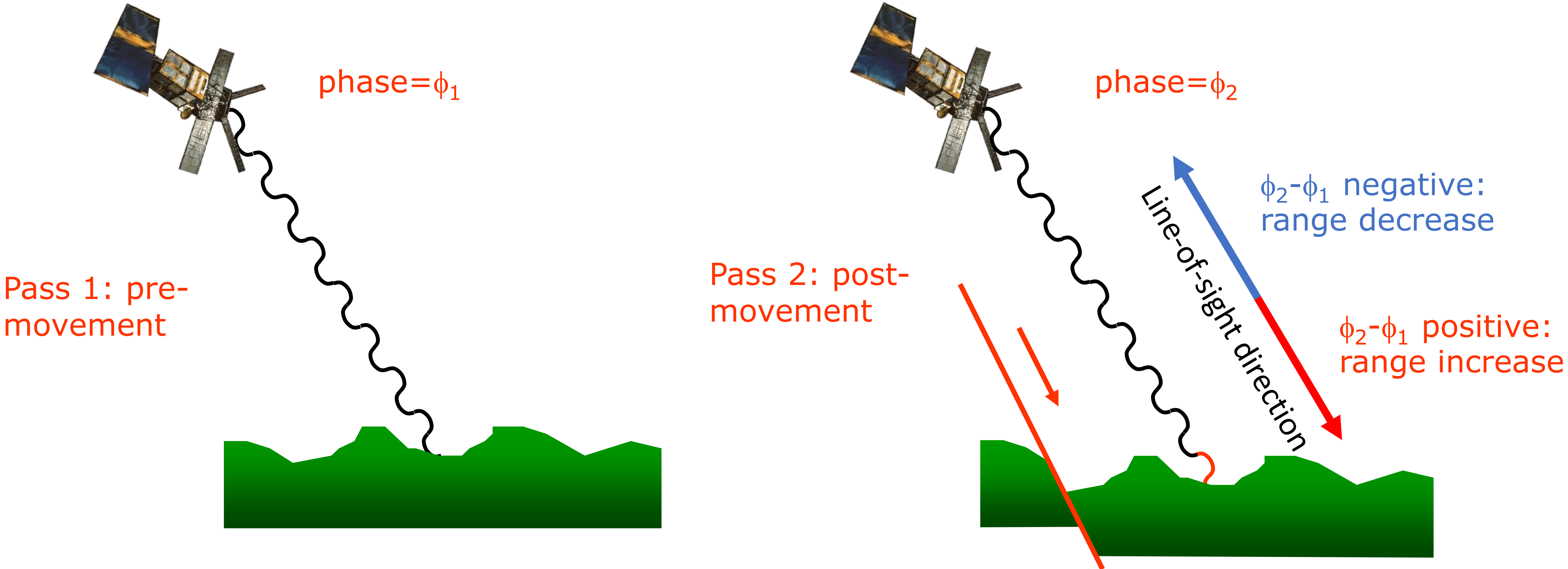


By **geodetic data**, we mean data that measure deformation (changes in shape) of the Earth's surface – e.g. InSAR, GPS/GNSS, differential lidar, optical image correlation…

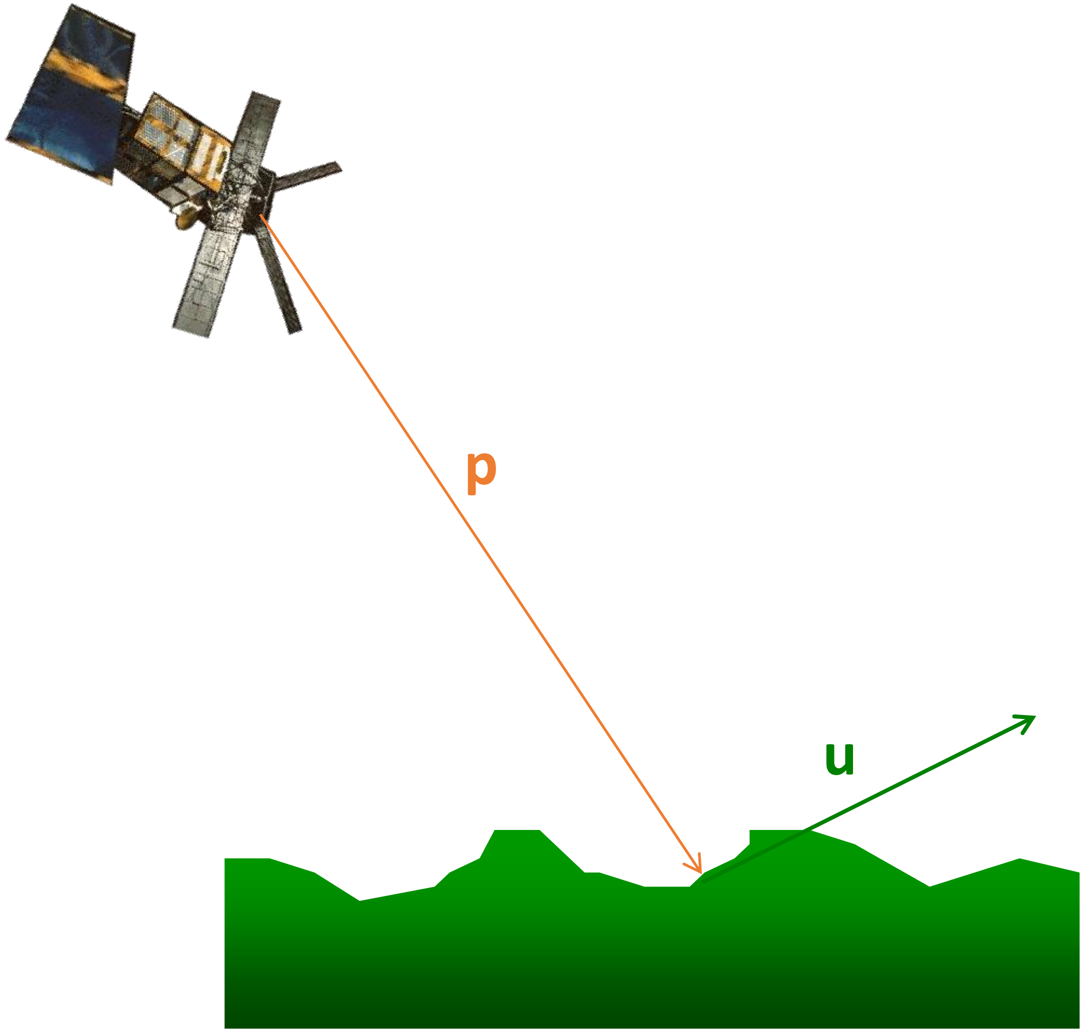
An individual SAR interferogram measures deformation in one dimension, in the radar line-of-sight



An individual SAR interferogram measures deformation in one dimension, in the radar line-of-sight



Vector description of InSAR

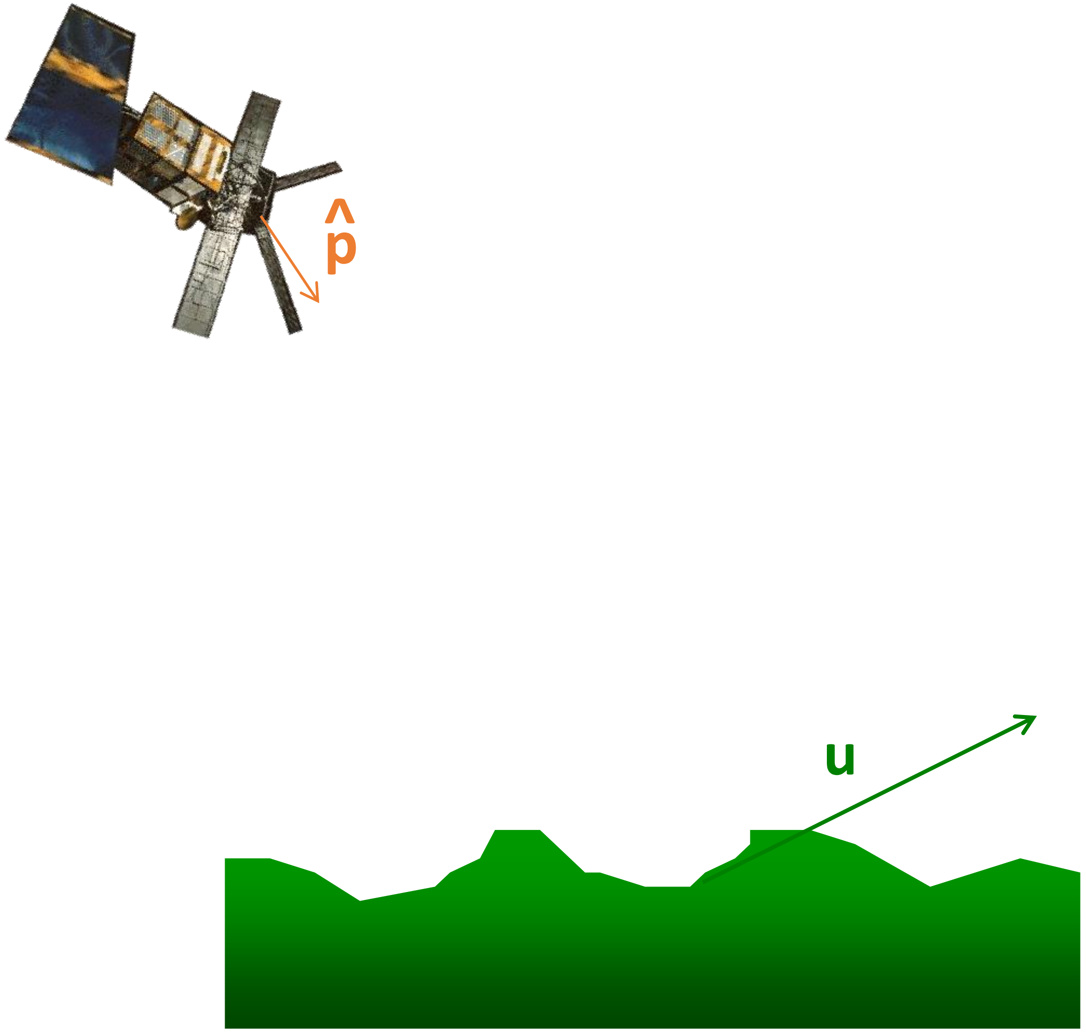


u = ground displacement vector

p = pointing vector (from satellite to ground target)

p is controlled by the satellite trajectory, beam mode (incidence angle) and position of the pixel within the swath

The unit pointing vector

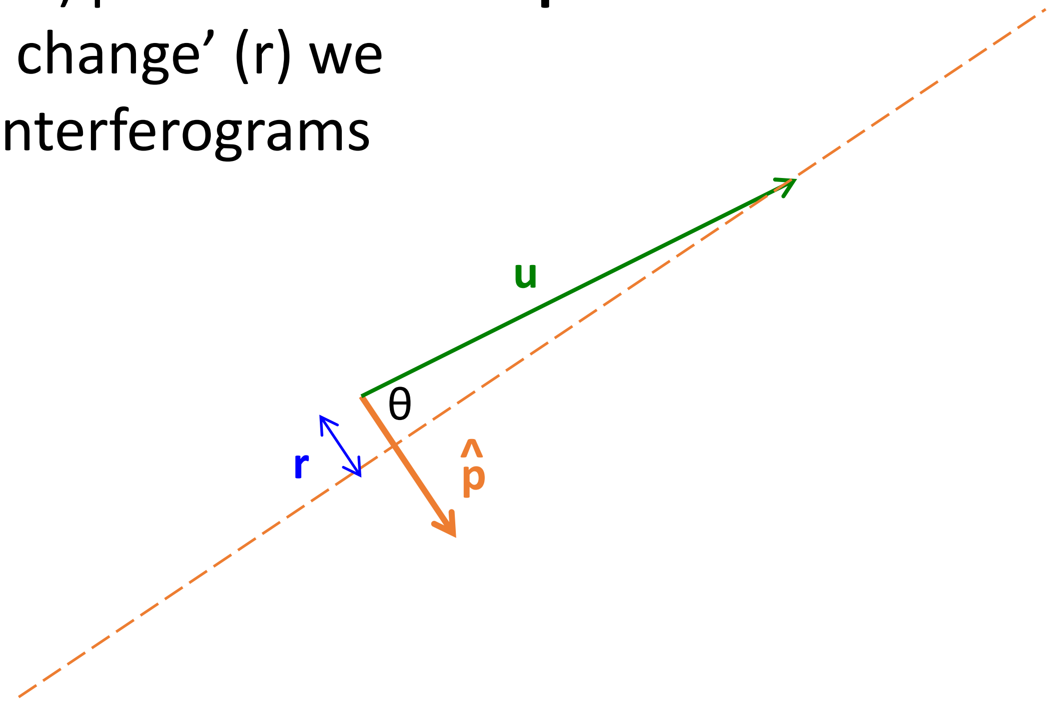


u = ground displacement vector

$\hat{\mathbf{p}}$ = unit pointing vector (from satellite to ground target)

Range change

the scalar (dot) product of **u** and $\hat{\mathbf{p}}$ is the 'range change' (r) we measure in interferograms



cross-section view

$$
\begin{aligned}
r &= \mathbf{u} \times \hat{\mathbf{p}} \\
&= |\mathbf{u}||\hat{\mathbf{p}}|\cos q \\
&= |\mathbf{u}|\cos q
\end{aligned}
$$

therefore, the key to modeling InSAR data is having a code that can simulate the displacements **u**

WARNING

Historically, people did not all use the same sign conventions in InSAR (including me…)

- Check whether your interferograms are ‘range change’ or ‘ground LOS displacement’
- Check if your pointing vectors are consistent with your interferograms (pointing from satellite to ground, or ground to satellite?)

If your reference image is the earlier of the two images, your interferogram should be in range change

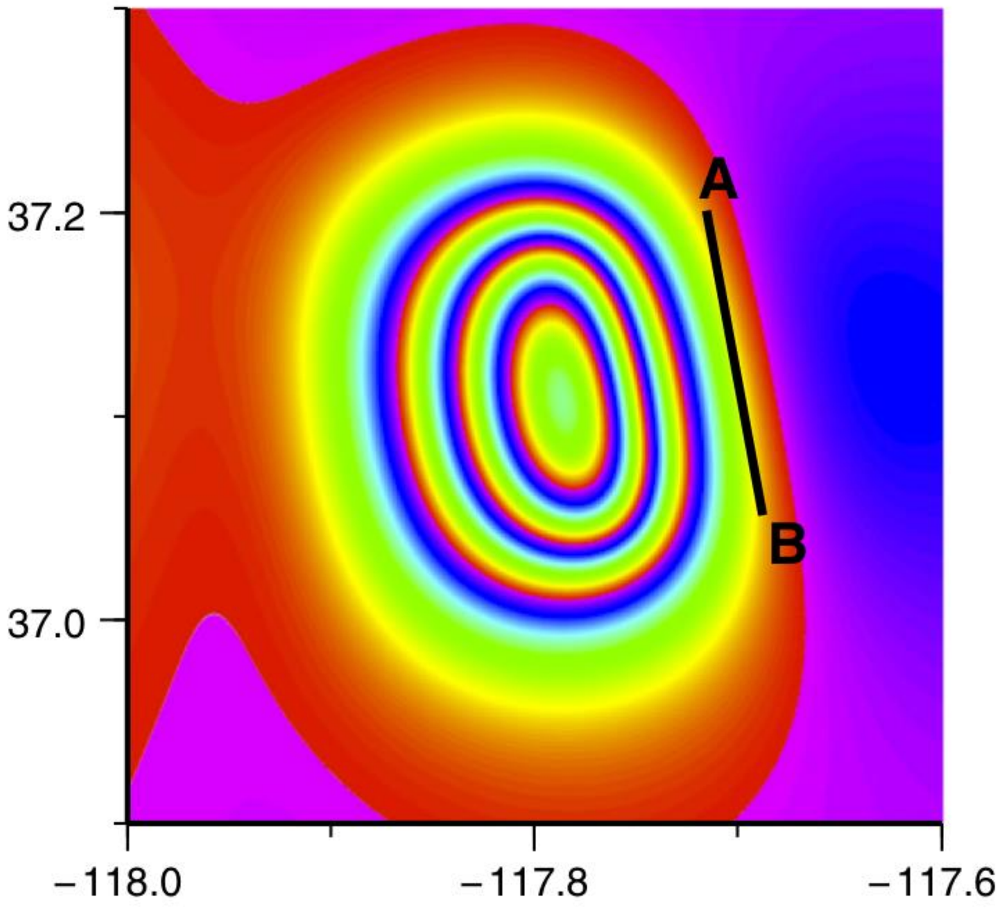
A **forward model** is a simulation of what InSAR would measure for a given set of model parameters

$$\mathbf{d}' = G(\mathbf{m})$$

simulated data

model theory (and InSAR geometry)

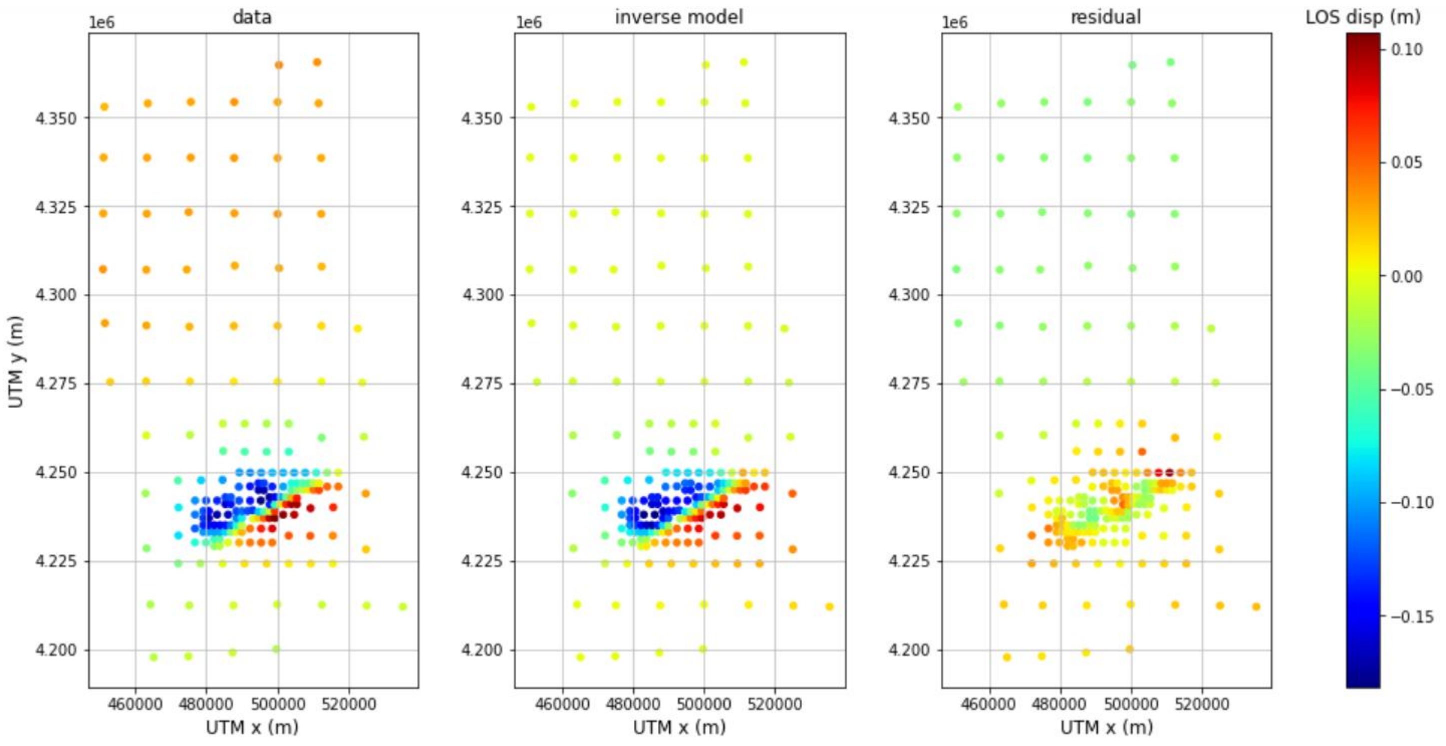
model parameters (location, model geometry, size, etc)



strike: 172°, dip: 38°, rake: –95°

Inverse modeling involves using observed data to estimate the most appropriate model parameters

data model data-model



$$\mathbf{m'} = G^{-1}(\mathbf{d})$$

estimated model parameters

inverse of the theoretical model

observed data

Many crustal deformation processes are elastic

…such as deformation before and during the 1906 San Francisco earthquake

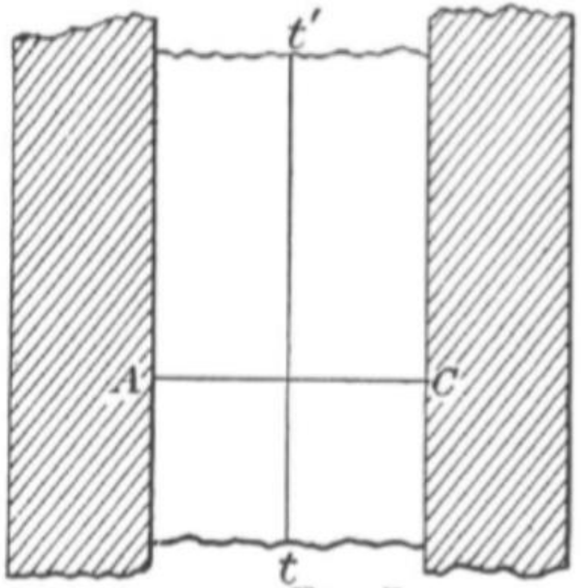


FIG. 7.

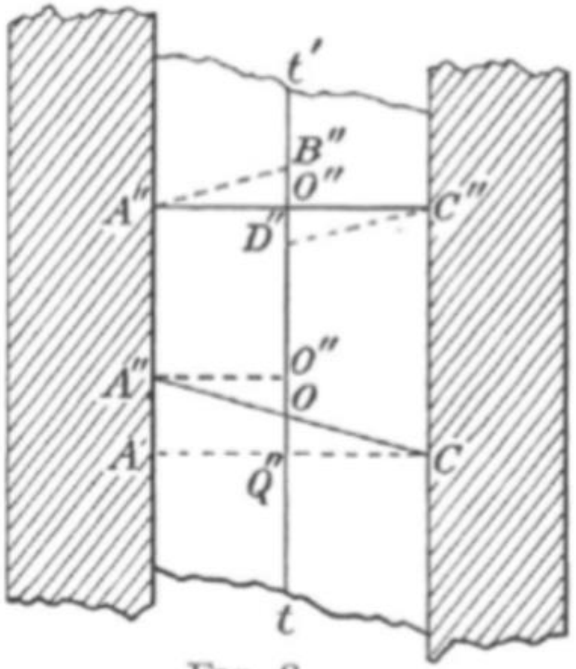


FIG. 8.

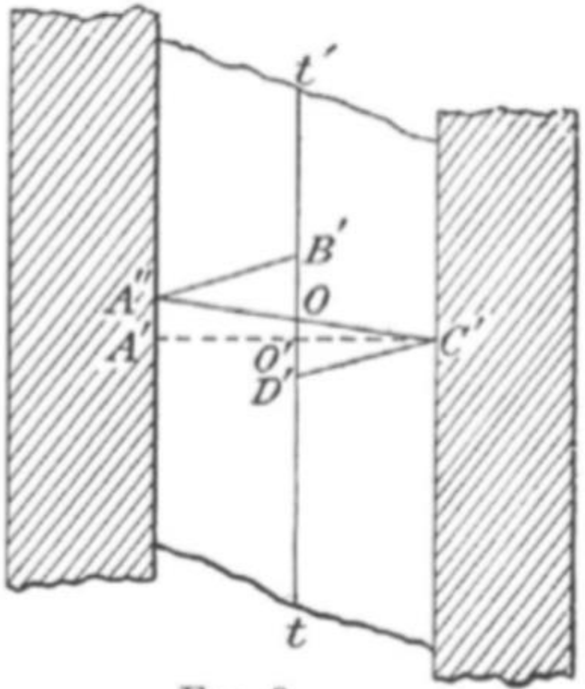
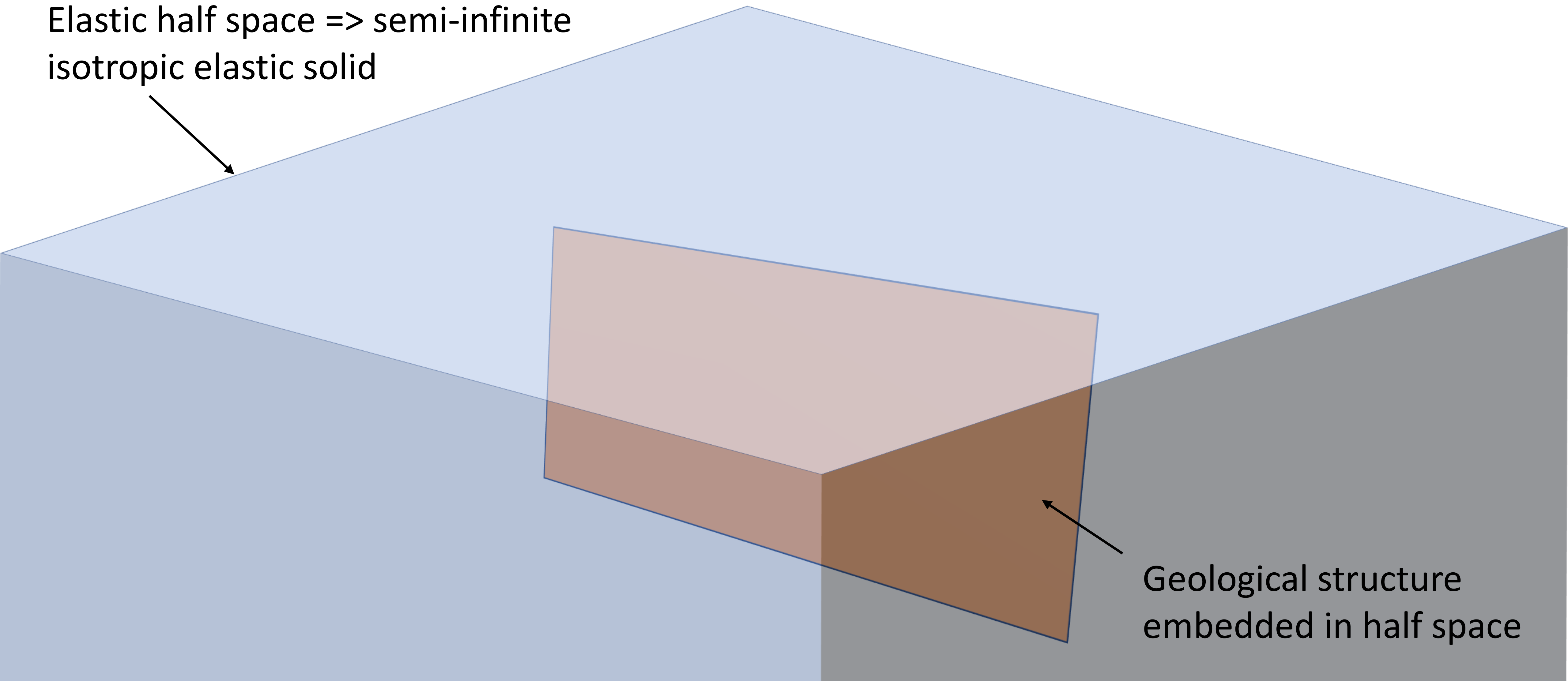


FIG. 9.

Reid, 1910

Elastic half space models

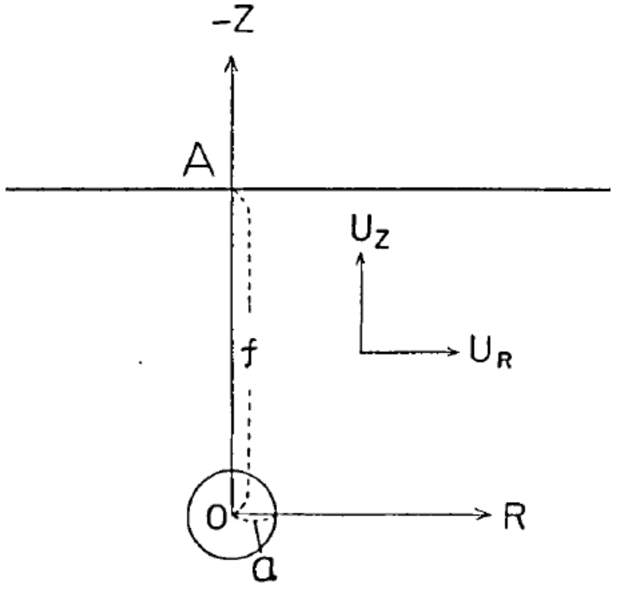


The Mogi model

BULLETIN OF THE EARTHQUAKE RESEARCH INSTITUTE
Vol. 36 (1958), pp. 99-134

6. *Relations between the Eruptions of Various Volcanoes and the Deformations of the Ground Surfaces around them.*

By Kiyoo MOGI,
Earthquake Research Institute.



Spherical pressure/volume change source (e.g. for magma chambers)

$$\Delta d = \frac{3a^3P}{4\mu}\frac{d}{(f^2+d^2)^{3/2}}$$

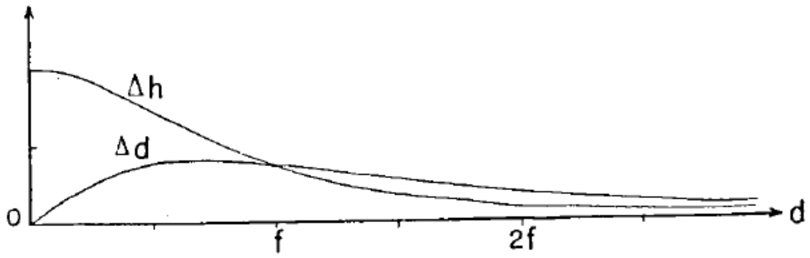
$$\Delta h = \frac{3a^3P}{4\mu}\frac{f}{(f^2+d^2)^{3/2}}$$

The good

- Simple model, very quick to compute

The bad

- Makes certain simplifying assumptions (isotropic half space, incompressible magma, source radius ≪ depth)



The Okada model

General solution for rectangular (1985) and point (1992) sources in an elastic half space

Pros

- Analytical solution, fast to compute
- Can model shear (fault slip) and opening (dike or sill intrusion/collapse)

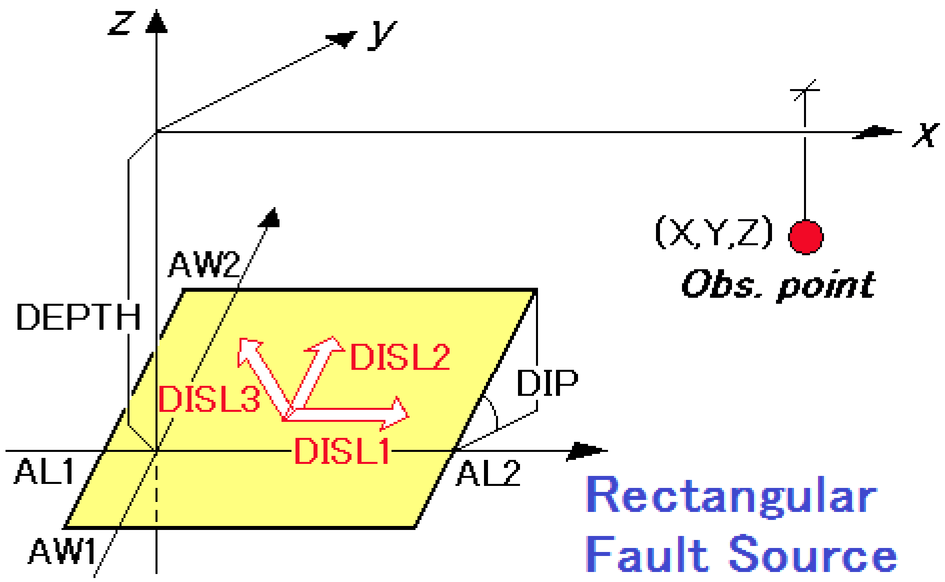
Cons

- Again, simplifying assumptions are not necessarily realistic
- Cannot tesselate into complex surfaces

Bulletin of the Seismological Society of America, Vol. 75, No. 4, pp. 1135–1154, August 1985

SURFACE DEFORMATION DUE TO SHEAR AND TENSILE FAULTS IN A HALF-SPACE

BY YOSHIMITSU OKADA*



Finite element models (FEMs)

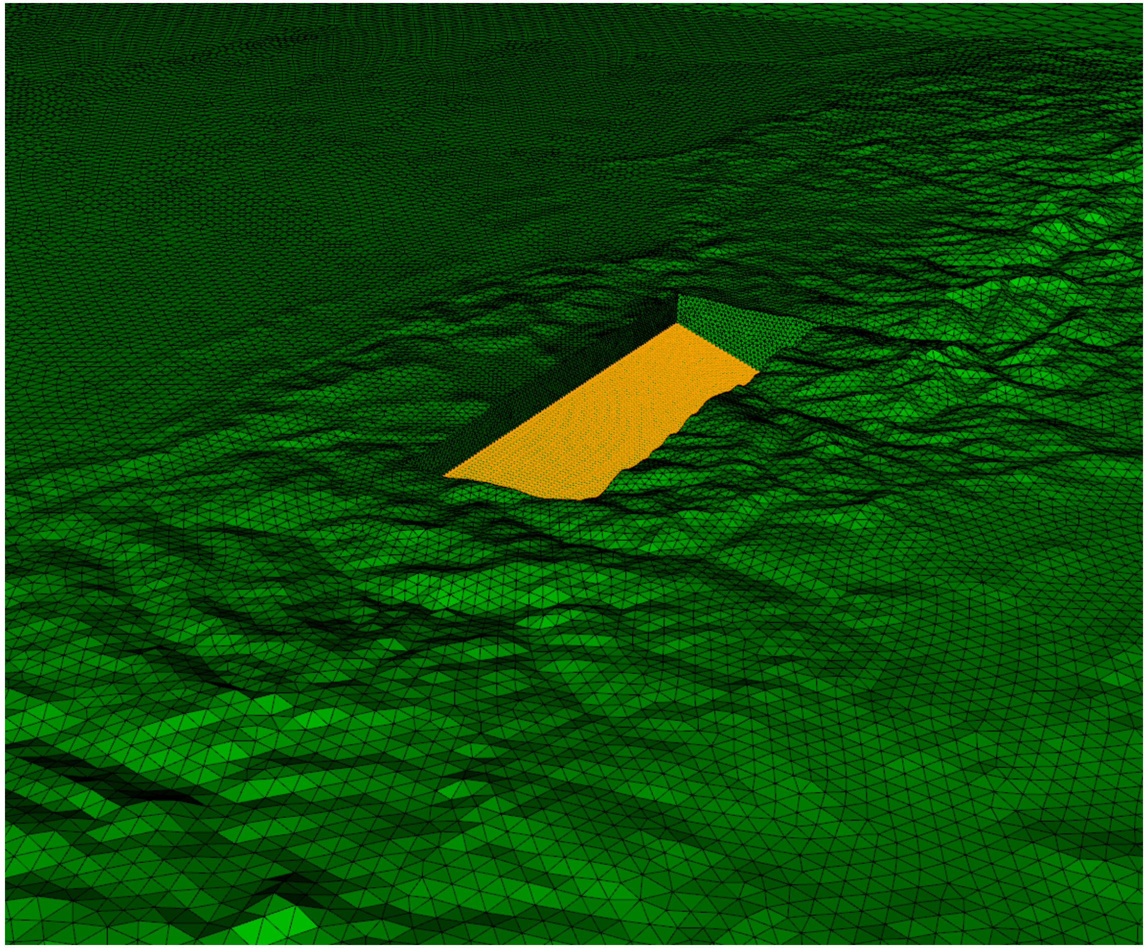
Can compute displacements and stresses for generalized solids and sources

Good

- Can incorporate heterogeneous material properties, nonplanar geometries, realistic topography
- Can incorporate more complex rheologies (e.g. viscoelasticity)

Less good

- Making meshes is complicated and slow
- Computing displacements is expensive (minutes to hours)



Gorkha, Nepal earthquake source region (7 million tetrahedral elements!)

Finite element models (FEMs)

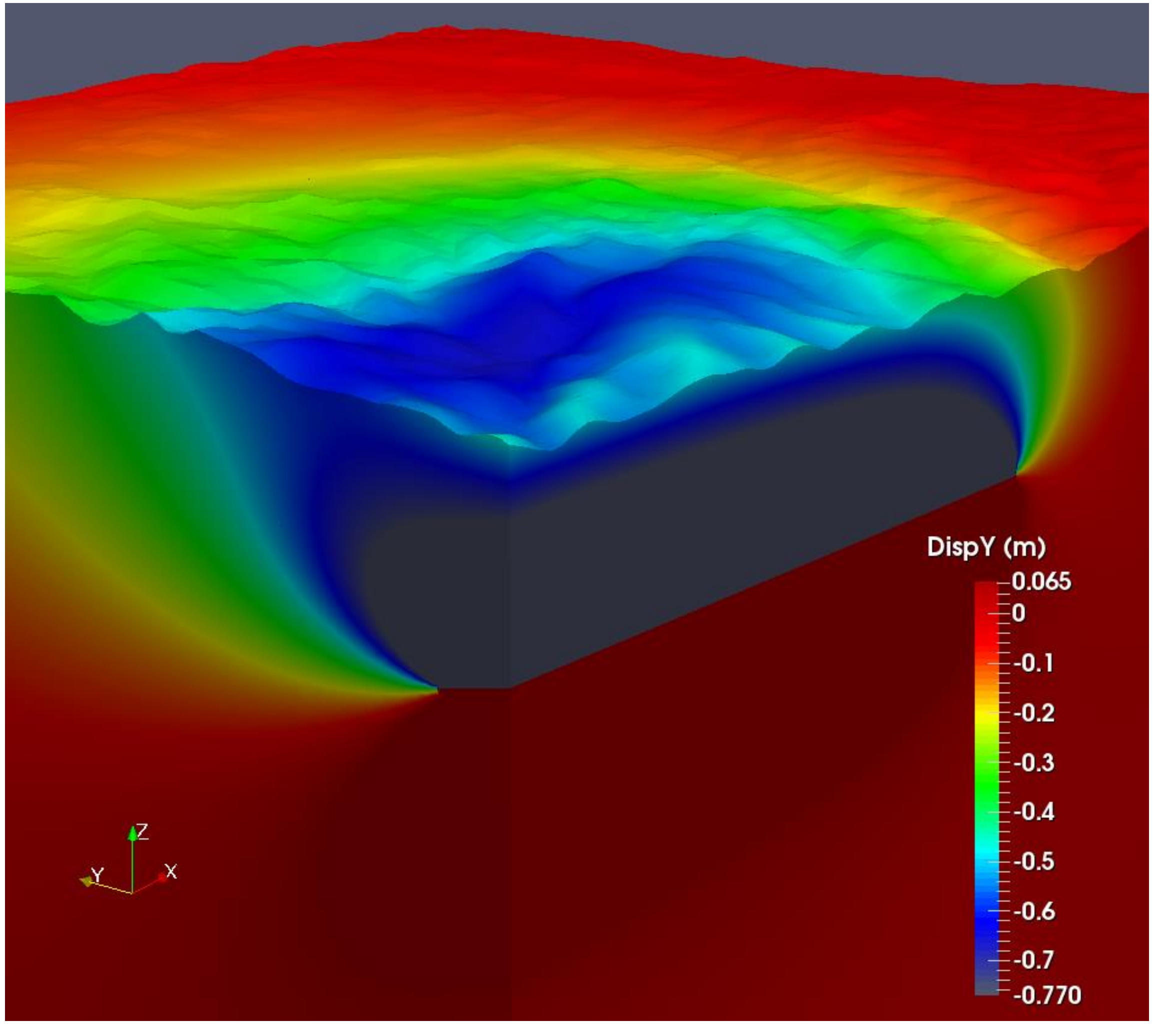
Can compute displacements and stresses for generalized solids and sources

Good

- Can incorporate heterogeneous material properties, nonplanar geometries, realistic topography
- Can incorporate more complex rheologies (e.g. viscoelasticity)

Less good

- Making meshes is complicated and slow
- Computing displacements is expensive (minutes to hours)



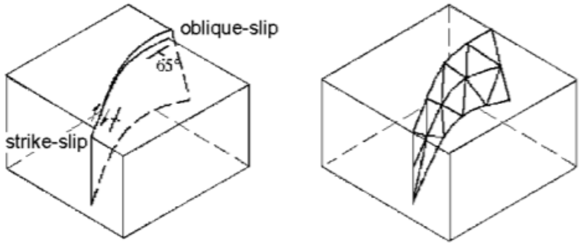
Boundary element models

Numerical method in which quantities are computed on surfaces rather than in volumes

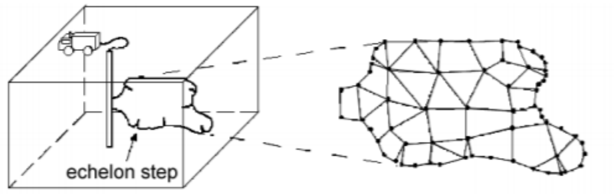
Yay
- Faster than FEMs
- Polygonal elements can allow complex source geometries
- Can compute stresses, use driving stresses

Nay
- Does not allow heterogeneous material properties
- Slower than analytical codes

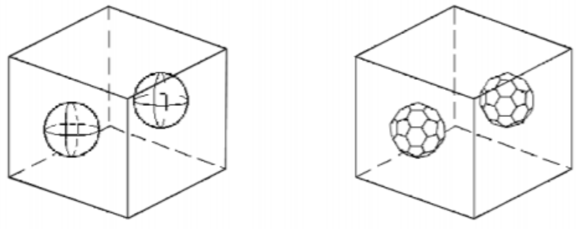
Fault surfaces which change in both strike and dip can be meshed without creating gaps.



Polygonal elements easily replicate the irregular boundary of a hydraulic fracture.



A spherical void can be modeled by assembling hexagonal and pentagonal elements in the manner of a soccer ball.



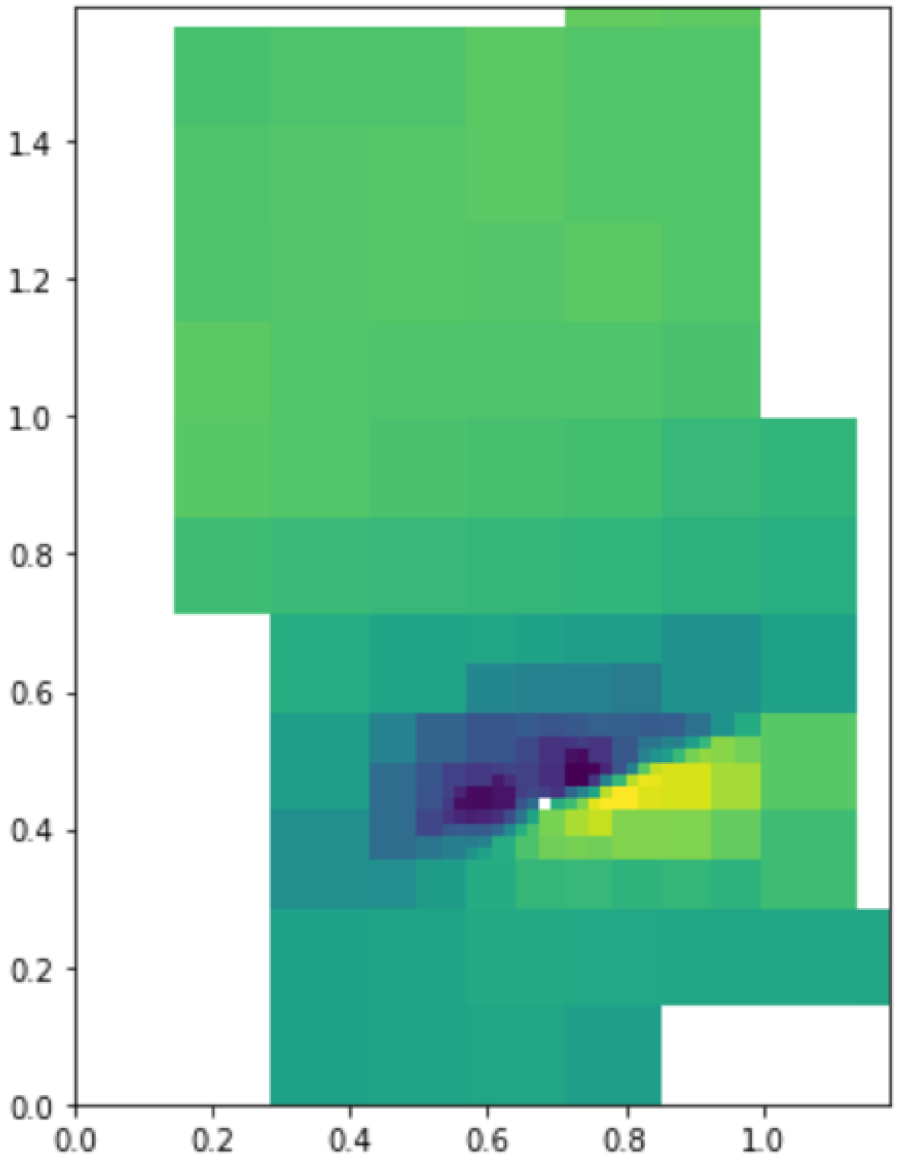
poly3d manual

Data downsampling

InSAR data are highly spatially correlated – you do not need every pixel to capture information on a process

Typically more datapoints = longer computation time => downsampling can make modeling more efficient

For example, quadtree decomposition (right) can divide an unwrapped interferogram into a set of regions with similar variance



Linear inverse modeling

If you have

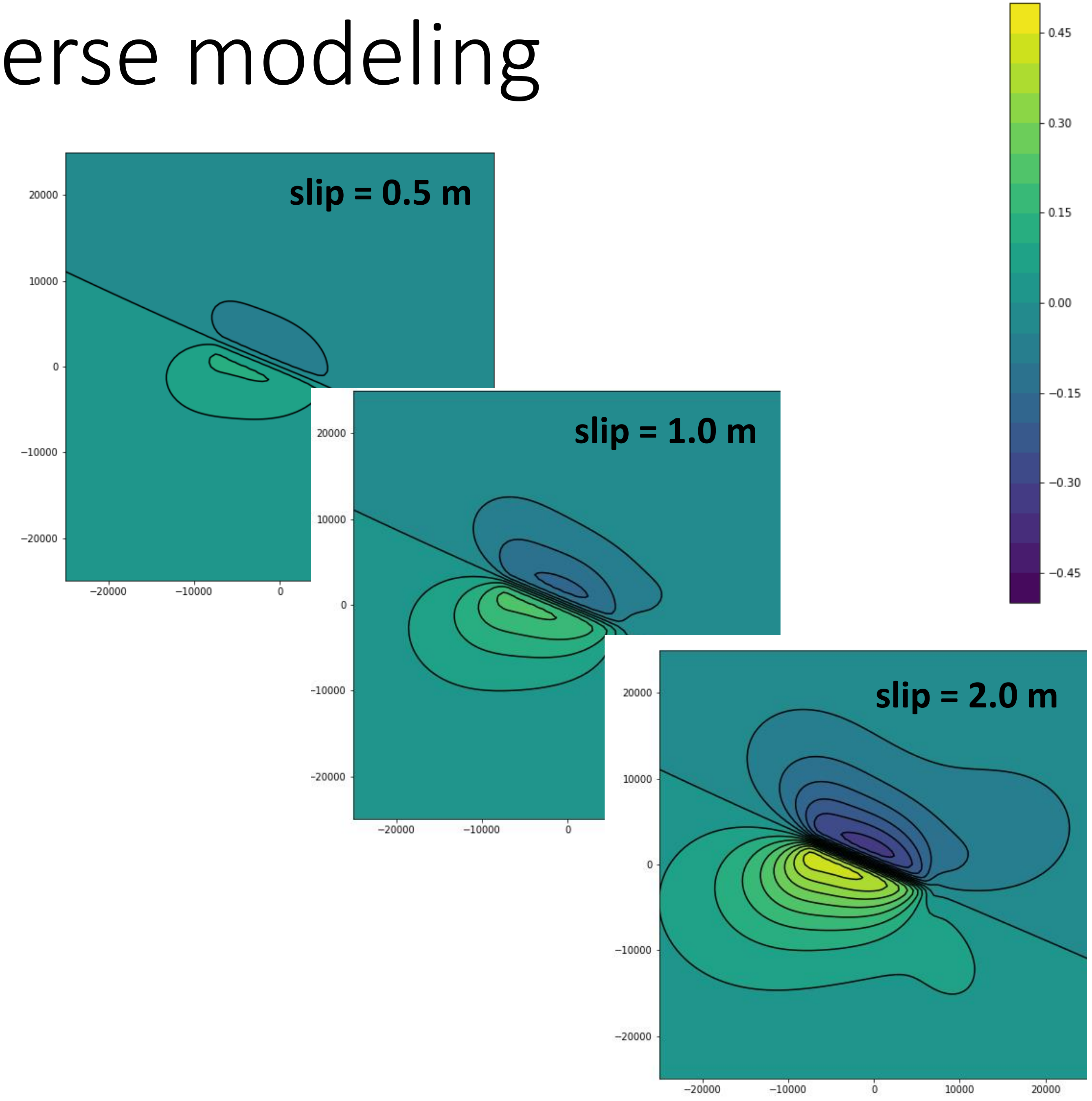
- Fixed model geometry
- Linear relationship between model parameters and surface displacements

Things are fairly straightforward!

- Model simplifies to a matrix inversion problem:

$$\mathbf{d} = \mathbf{G}\,\mathbf{m}$$
$$\mathbf{m} = (\mathbf{G}^T\mathbf{G})^{-1}\,\mathbf{G}^T\,\mathbf{d}$$

(**G** is a matrix of Green's functions)



Nonlinear inverse modeling

Unfortunately, not everything has a linear relationship with displacement!

- changes in position/depth
- changes in dimensions
- changes in orientation

In these cases, we may use an optimization approach:

- forward model displacements using guessed model parameters
- calculate the fit of the forward model to the data
- vary the model parameters until a good fit is obtained (e.g. by using an algorithm!)