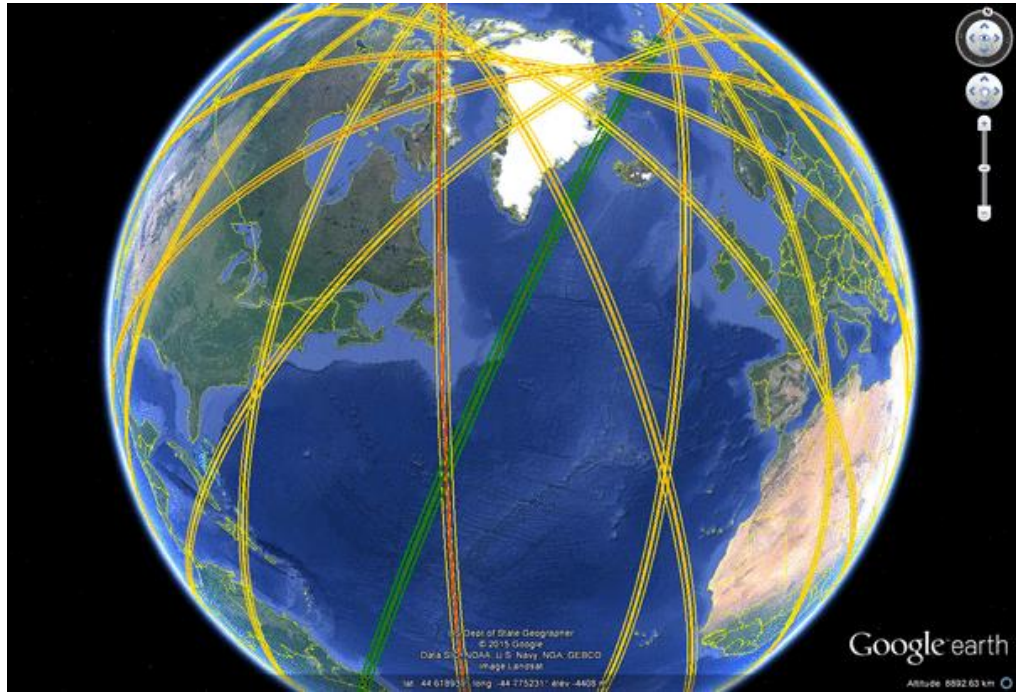
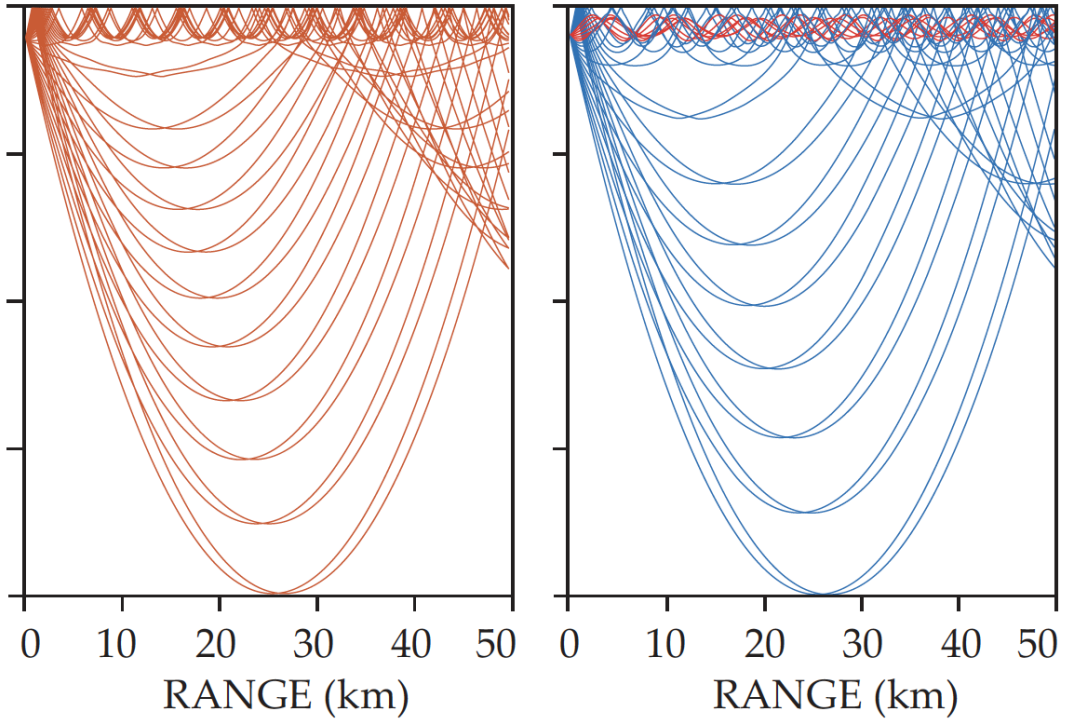
ObsFit : an MITgcm package for model-observation comparison

Ariane Verdy, Matt Mazloff, Bruce Cornuelle, Patrick Heimbach

SWOT altimeter footprint

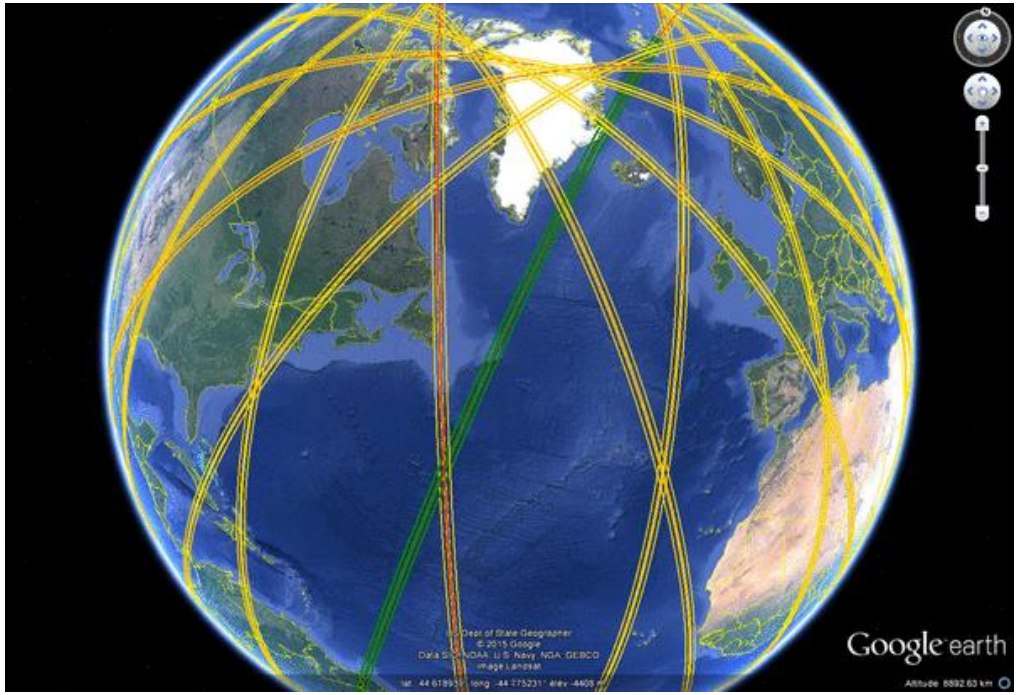


Tomography ray paths

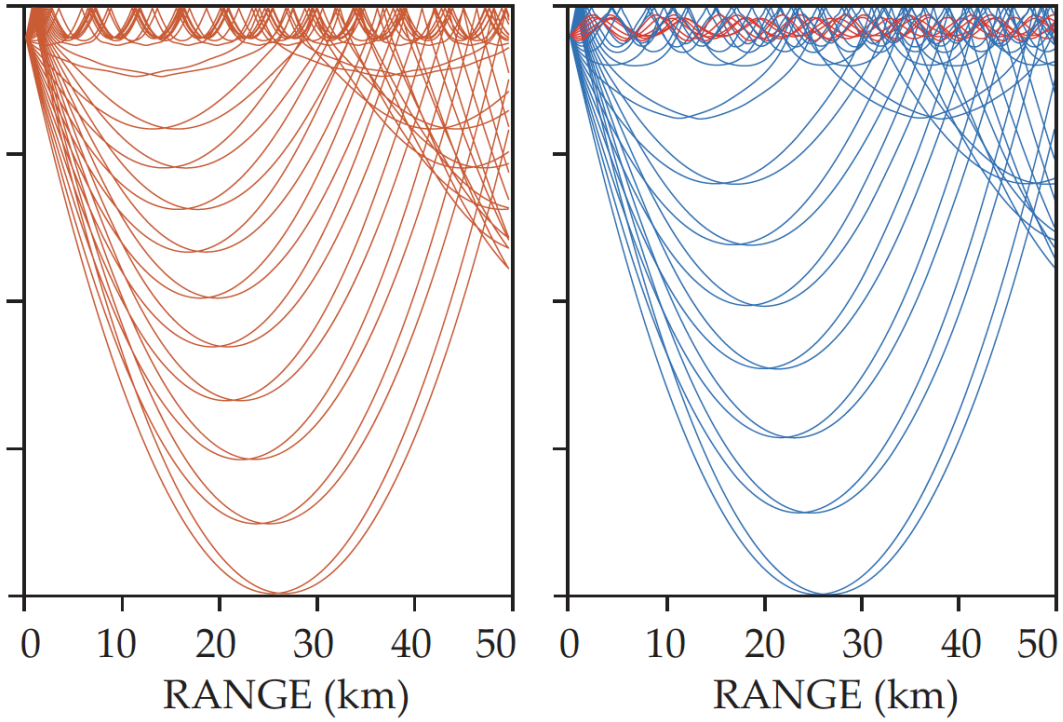


The “**ObsFit**” package for MITgcm consists of a set of routines that samples the model solution at the **time** and **location** of the observations and quantifies the differences. The package efficiently handles data that are spatially located at a single **point** or **averaged over arbitrary areas** (e.g. along tomographic rays or over a satellite track) and data that are **instantaneous** or **time-averaged**.

SWOT altimeter footprint



Tomography ray paths



Why?

- **Sparse data (memory allocation)**
  Obsfit reads in vector of observations instead of 2D or 3D matrices
- **Respect the data sampling**
  No need to interpolate observations to model grid
- **Flexible averaging operators**
  ObsFit handles spatial and temporal averaging and integration
- **Multi-grid optimization**
  Model-equivalent values are stored in data space, independent of model grid
- **Universal model validation**
  Input and output files are independent of model setup

The **observation** of some **property** comes out of a measurement system

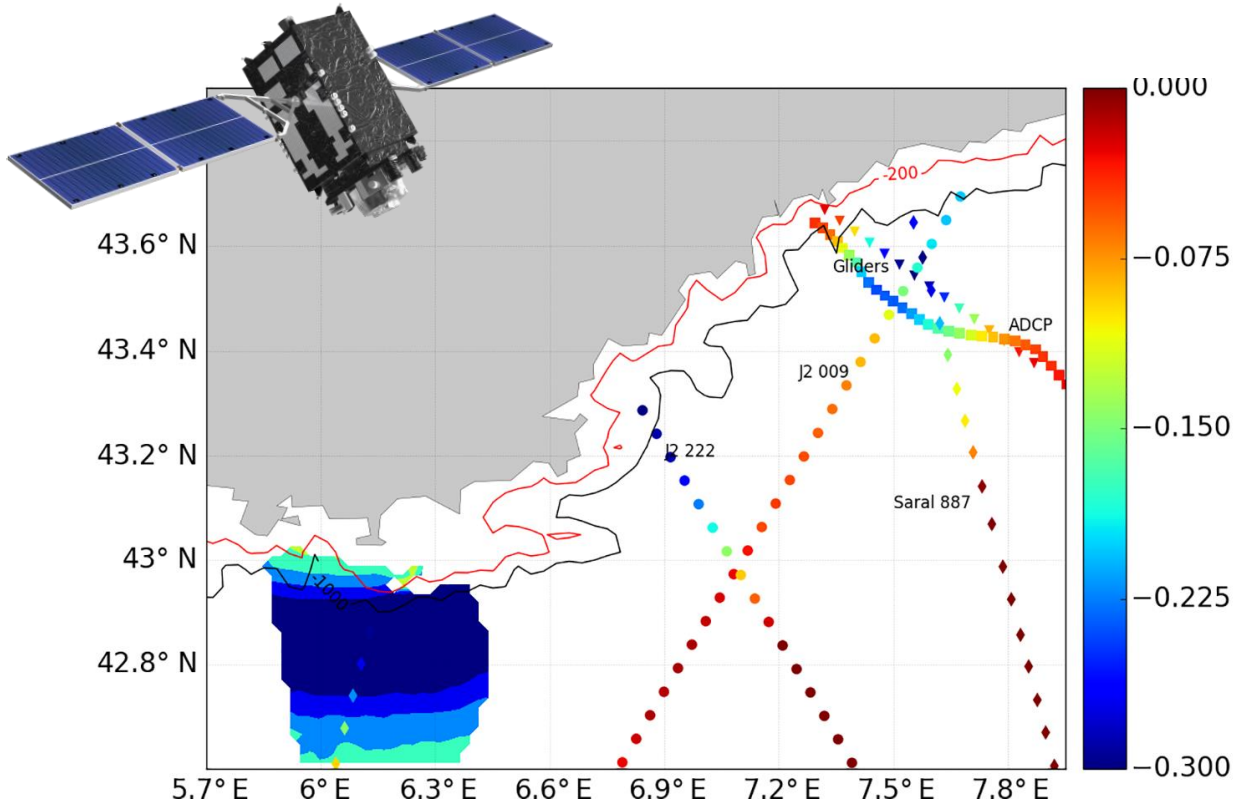
- *start time*
- *duration*
- *obs_weight (*uncertainty)

An observation is made of *NP* **samples**

- *spatial coordinates* (x, y, z)
- *sample_weight* (w) for averaging

ObsFit will return model-equivalent values for each **observation** given

Observation:
SSH
October 1, 2012



The **observation** of some **property** comes out of a measurement system

- *start time*
- *duration*
- *obs_weight (*uncertainty)

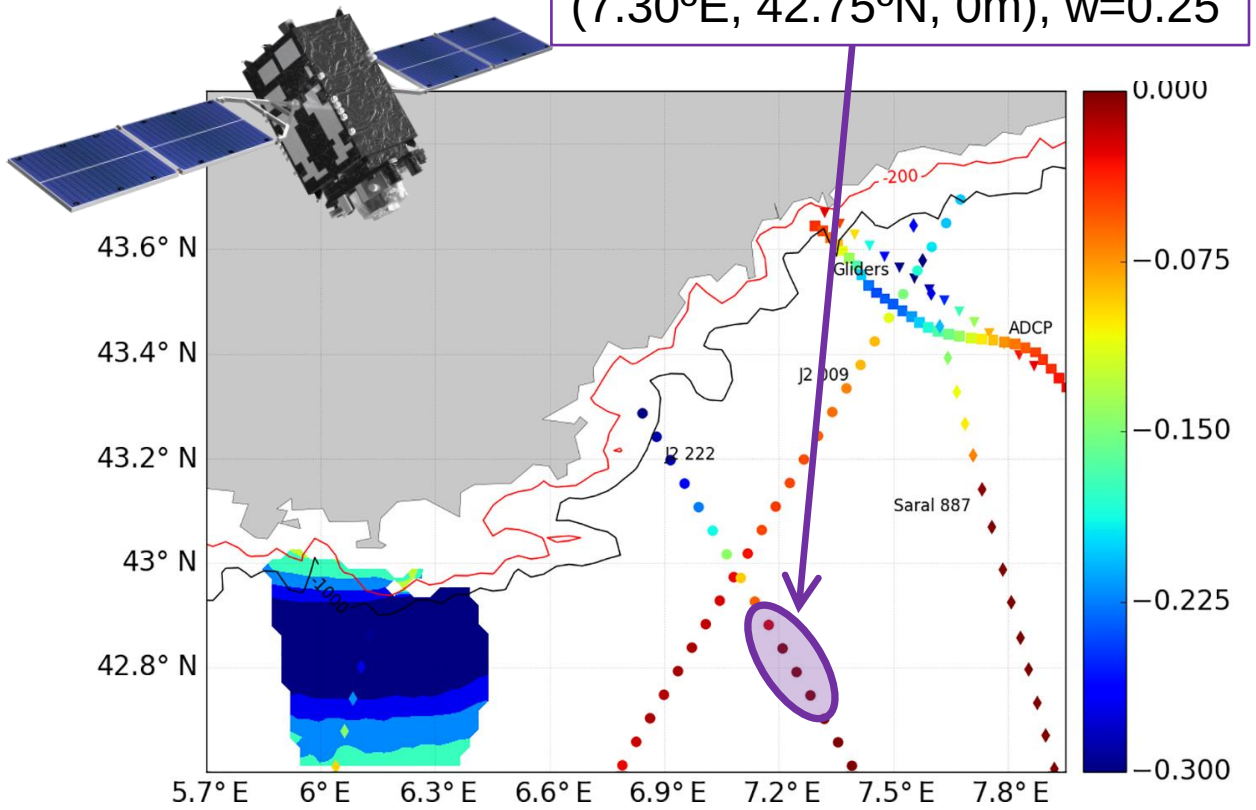
An observation is made of *NP* **samples**

- *spatial coordinates* (x, y, z)
- *sample_weight* (w) for averaging

ObsFit will return model-equivalent values for each **observation** given

Observation:
SSH
October 1, 2012
NP = 4

Samples:
(7.18ºE, 42.90ºN, 0m), w=0.25
(7.22ºE, 42.85ºN, 0m), w=0.25
(7.26ºE, 42.80ºS, 0m), w=0.25
(7.30ºE, 42.75ºN, 0m), w=0.25



Pre-processing:

- Read input files (# observations, # samples, start time)
- Identify observations that fall within the simulation; corresponding samples in each model tile
- For **each sample**, save [*start time, end time, time operation, property, interpolation information*]

Sampling:

- If [*start time* < model time < *end time*], then interpolate property to (*x, y, z*)
- Model-equivalent values are saved in binary tiled files
- If time operation = averaging or integrating, read previous value and add

Post-processing:

- Read tiled files and combine
- Average sampled values to get model-equivalent of each observation
- Write in a **global netcdf file**, which is later read during cost calculation

https://github.com/averdy/obsfit_mitgcm