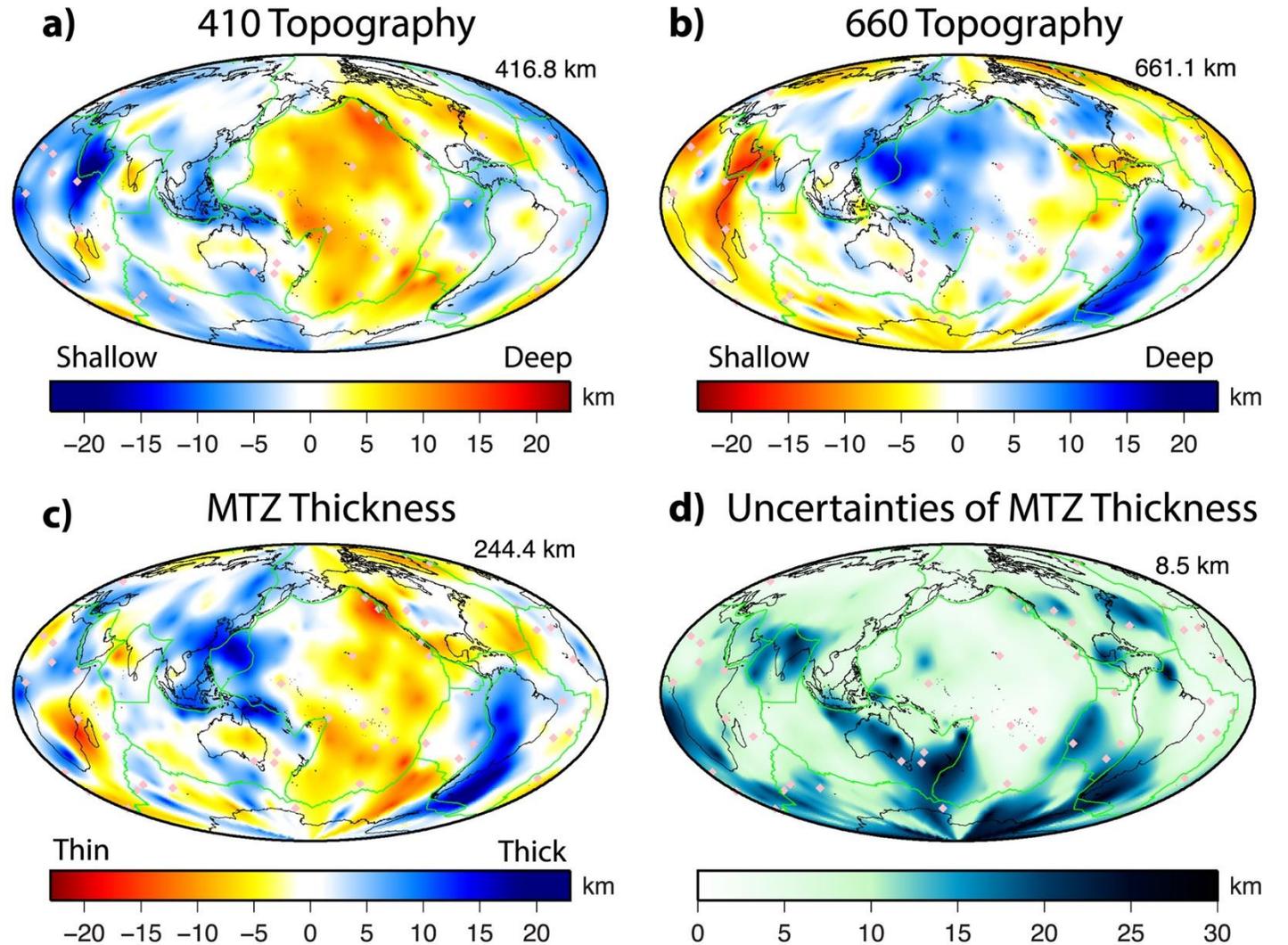


Displaced and Faded:

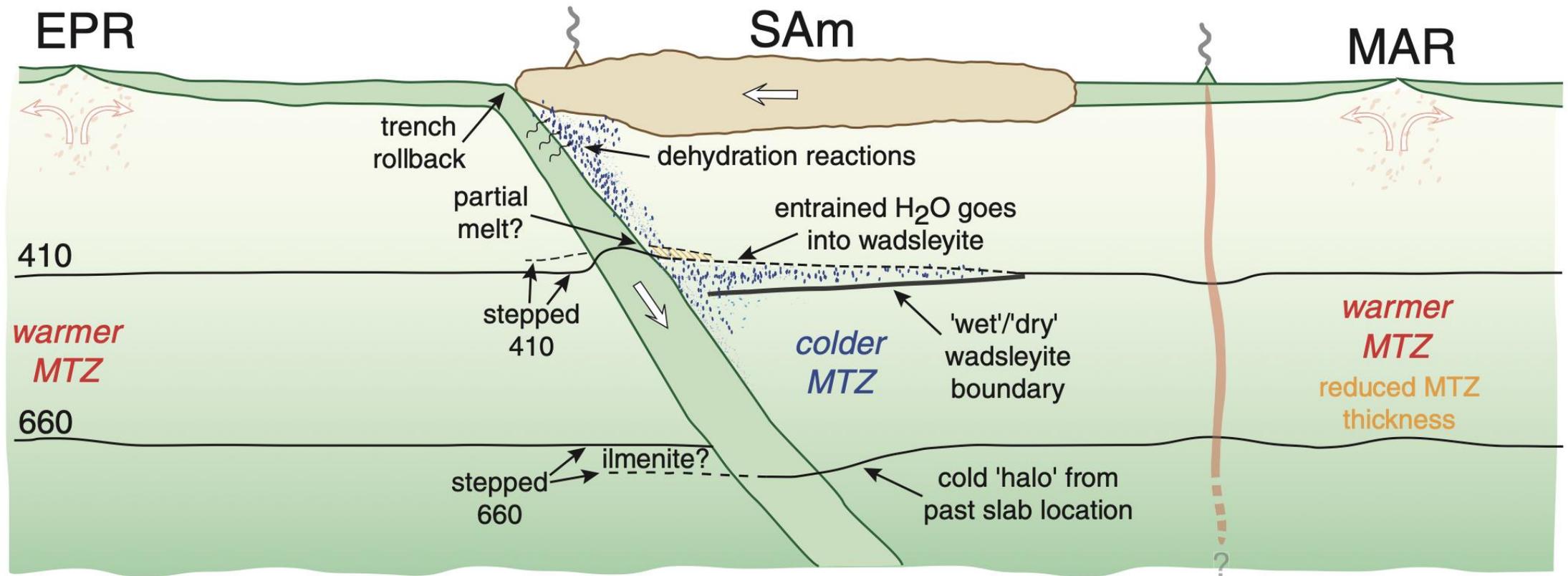
How thermodynamics and kinetics collude to complicate seismic structures in Earth's mantle

B. Kerswell, J. Wheeler, R. Gassmöller

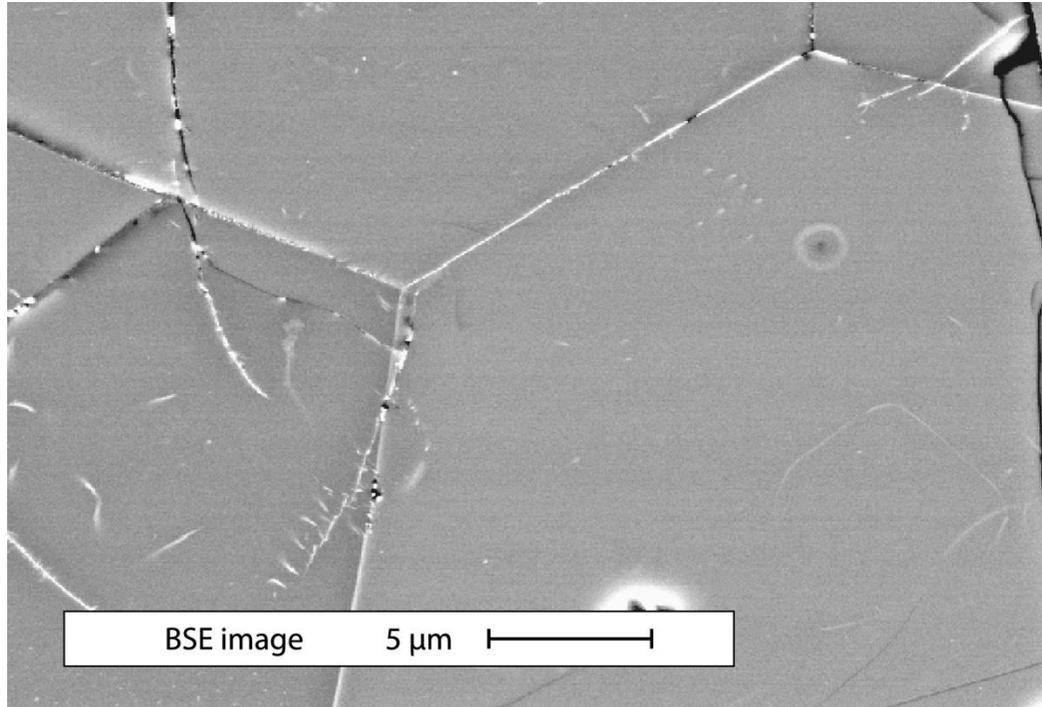
Seismic discontinuities show global variability



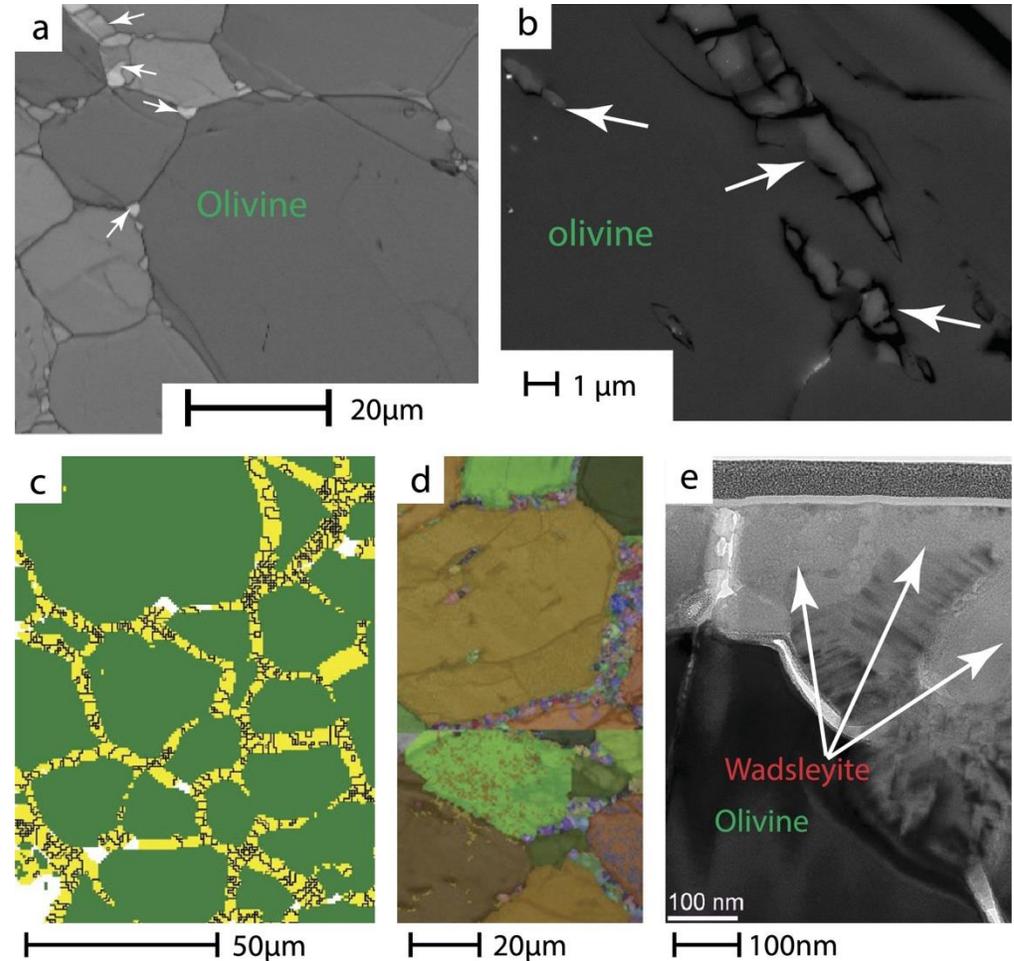
Thermodynamics cannot explain the seismic variability alone



Mantle phase transitions are rate-limited



Growth-limited after rapid nucleation saturation



Research question

What is the impact of stress, microstructures, and mineral-scale kinetics on mantle-scale seismic structures (and dynamics)?

Hypothesis

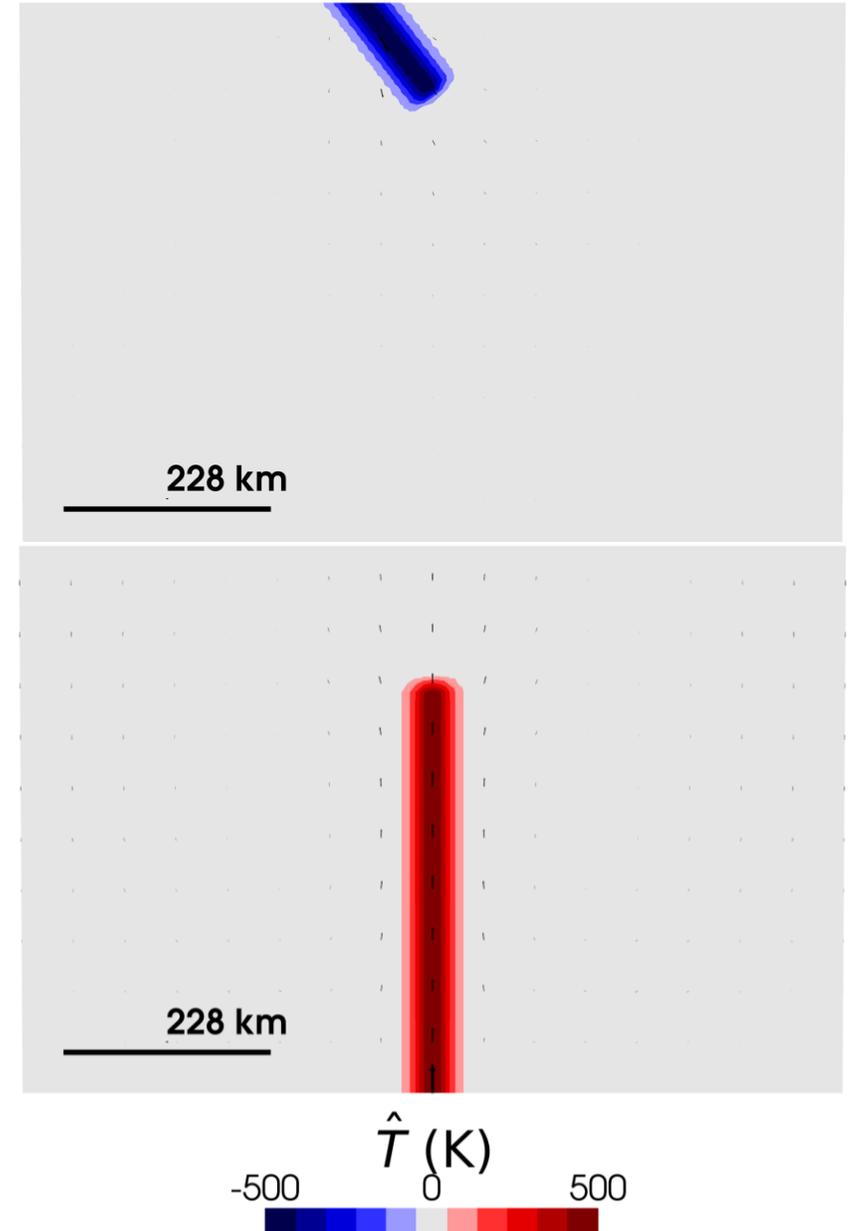
Variable olivine \leftrightarrow wadsleyite growth rates
in slabs and plumes are consistent with
observed seismic variability

We simulate plumes and slabs using ASPECT with a compressible mantle

Bangerth et al. (2025), and others

$$\nabla P - \nabla \cdot \sigma' = \rho g \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$$

$$\rho \bar{C}_p \left(\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T \right) - \nabla \cdot (\bar{k} \nabla T) = \dots \text{heating}$$



Growth rate depends on kinetic factors, dynamic PT, and saturation

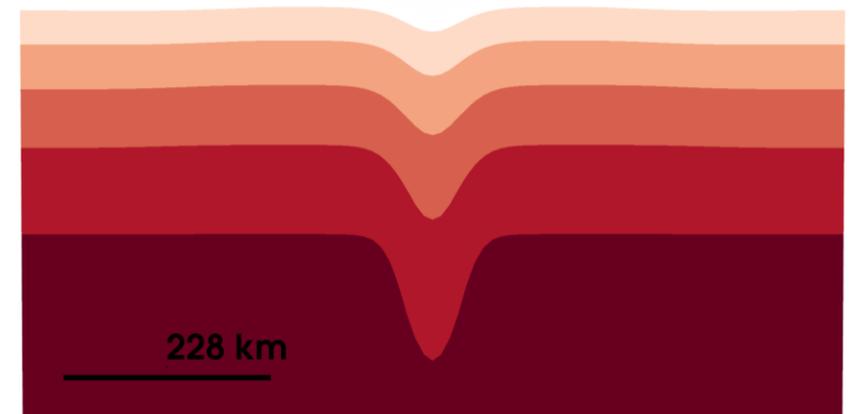
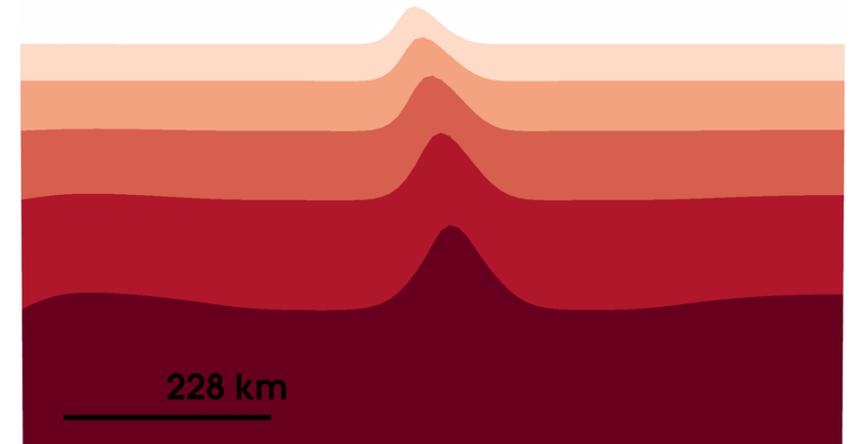
Hosoya et al. (2005), and others

$$\frac{dx}{dt} = Z T \exp\left(-\frac{H^* + PV^*}{RT}\right) \left(1 - \exp\left[-\frac{\Delta G}{RT}\right]\right) (1 - X)$$

$$Z = \frac{6.67}{d} A C_{OH}^n$$

$$\Delta G = \Delta \bar{G} + \hat{P} \Delta \bar{V} - \hat{T} \Delta \bar{S}$$

$$\hat{P} = P - \bar{P} \quad \hat{T} = T - \bar{T}$$



Thermodynamic term

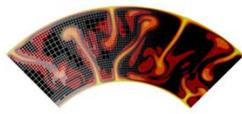


ASPECT Hackathon 2025

ASPECT is a community software project

```
44 // Get thermodynamic terms from profile
43 const double dS = profile.get_data_component(profile_pos, dS_idx);
42 const double dS = profile.get_data_component(profile_pos, dS_idx);
41 const double dV = profile.get_data_component(profile_pos, dV_idx);
40
39 if (!in.requests_property(MaterialProperties::viscosity))
38 {
37   // Compute viscosity factor
36   std::vector<double> viscosity_factors(transition_depths);
35   for (int i = 0; i < transition_depths.size(); i++)
34     calculate_viscosity_factor(transition_depths[i], viscosity_factors[i],
33     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
32     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
31     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
30     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
29     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
28     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
27     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
26     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
25     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
24     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
23     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
22     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
21     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
20     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
19     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
18     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
17     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
16     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
15     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
14     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
13     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
12     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
11     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
10     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
9     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
8     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
7     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
6     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
5     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
4     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
3     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
2     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i],
1     viscosity_factors[i], viscosity_factors[i], viscosity_factors[i];
}
}

// Clamp composition to valid range
13 const double X_clamped = std::max(0.0, std::min(1.0, X));
12
11 // Fill mass fractions array using clamped mass fraction
10 std::vector<double> mass_fractions(this->intrinsics().compositional_fields, 1.0);
9 mass_fractions[0] = 1 - X_clamped;
8 mass_fractions[1] = X_clamped;
7
6 // Compute volume fractions from mass fractions and material densities
5 const std::vector<double> volume_fractions = MaterialUtilities::compute_volumes_from_masses(mass_fractions, densities, transition_depths);
4
3 // Average material properties using computed mass fraction
2 const double cp = MaterialUtilities::average_value(volume_fractions, densities, MaterialUtilities::arithmetic);
1 const double beta = MaterialUtilities::average_value(volume_fractions, betas, MaterialUtilities::arithmetic);
0 const double alpha = MaterialUtilities::average_value(volume_fractions, alphas, MaterialUtilities::arithmetic);
-1 const double cp = MaterialUtilities::average_value(mass_fractions, cps, MaterialUtilities::arithmetic);
-2
-3 // Average material properties using computed mass fraction
-4 if (seismic_out)
-5 {
-6   if (Vp_a_idx != numbers::invalid_unsigned_int && Vp_b_idx != numbers::invalid_unsigned_int)
-7     const double Vp = MaterialUtilities::average_value(volume_fractions, Vps, MaterialUtilities::arithmetic);
-8     seismic_out->vp[q] = Vp;
-9
-10
-11
-12
-13
-14
-15
-16
-17
-18
-19
-20
-21
-22
-23
-24
-25
-26
-27
-28
-29
-30
-31
-32
-33
-34
-35
-36
-37
-38
-39
-40
-41
-42
-43
-44
-45
-46
-47
-48
-49
-50
-51
-52
-53
-54
-55
-56
-57
-58
-59
-60
-61
-62
-63
-64
-65
-66
-67
-68
-69
-70
-71
-72
-73
-74
-75
-76
-77
-78
-79
-80
-81
-82
-83
-84
-85
-86
-87
-88
-89
-90
-91
-92
-93
-94
-95
-96
-97
-98
-99
100
```



ASPECT 3.1.0-pre

Community Project

ASPECT is a community software project. A list of ASPECT developers and contributors is available [here](#). Contributions to software or documentation by every user are welcome and encouraged. See [here](#) for how to contribute.

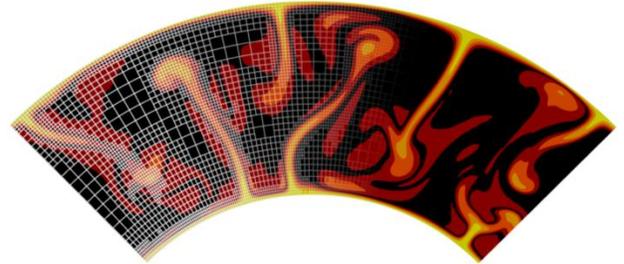
Search

- ASPECT
- User Guide
- Parameter Documentation
- Developer Documentation
- Authors
- References



Simplify infrastructure with MongoDB Atlas, the leading developer data platform

Ad by EthicalAds



About ASPECT

ASPECT is a code to simulate problems in thermal convection. Its primary focus is on the simulation of processes in the Earth's mantle, but its design is more general than that. The primary aims developing ASPECT are:

- Usability and extensibility:** Simulating mantle convection is a difficult problem characterized not only by complicated and nonlinear material models but, more generally, by a lack of understanding which parts of a much more complicated model are really necessary to simulate the defining features of the problem. This uncertainty requires a code that is easy to extend by users to support the community in determining what the essential features of convection in the Earth's mantle are.

ASPECT Hackathon 2025

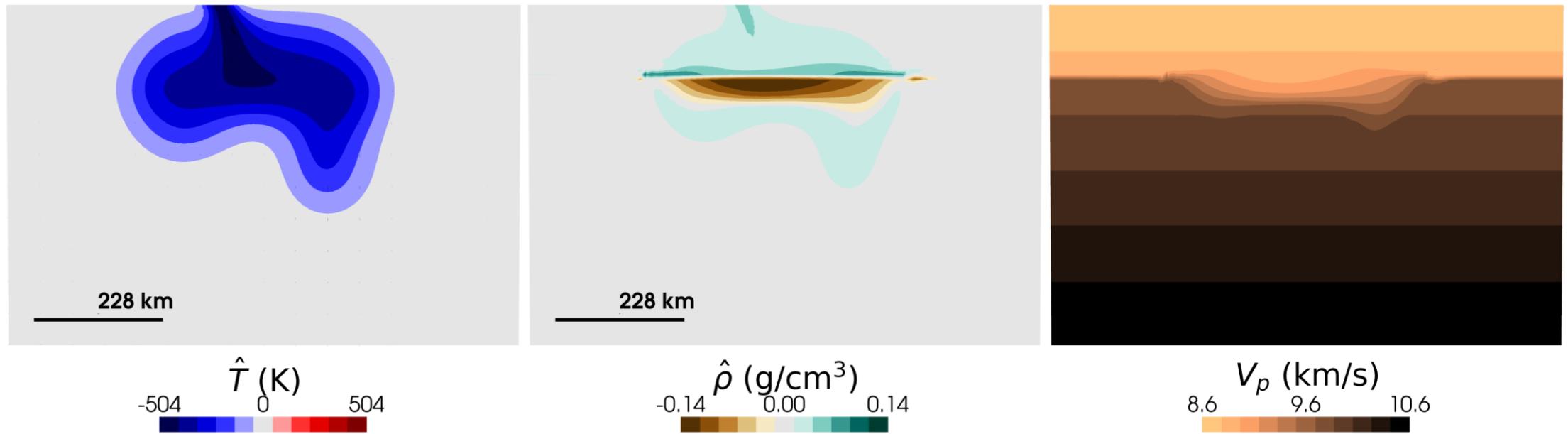
Bear Lake, Utah



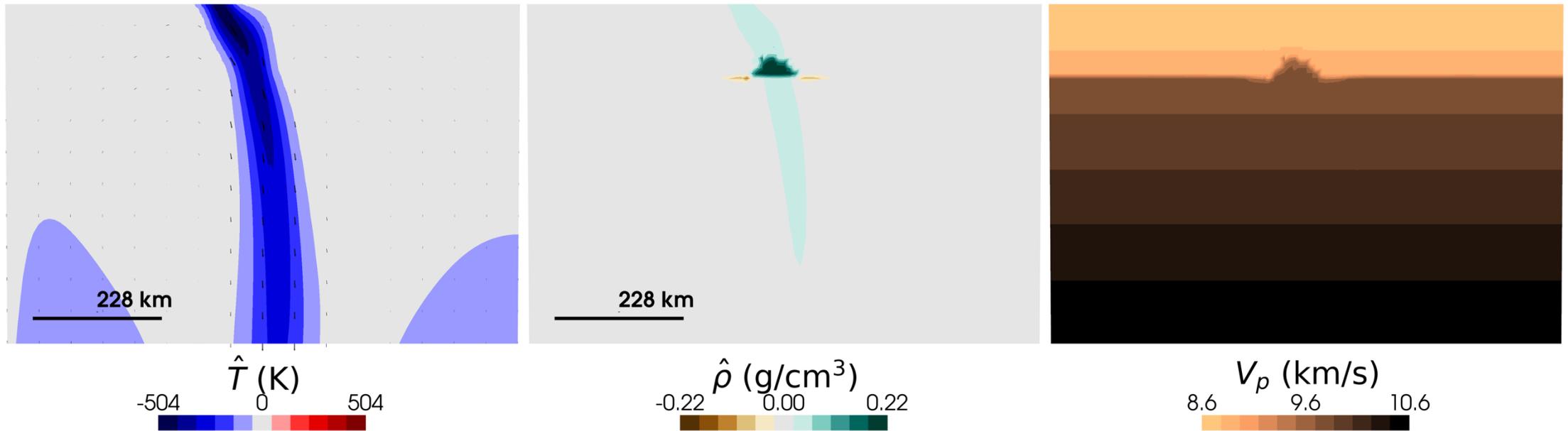
ASPECT Hackathon 2025



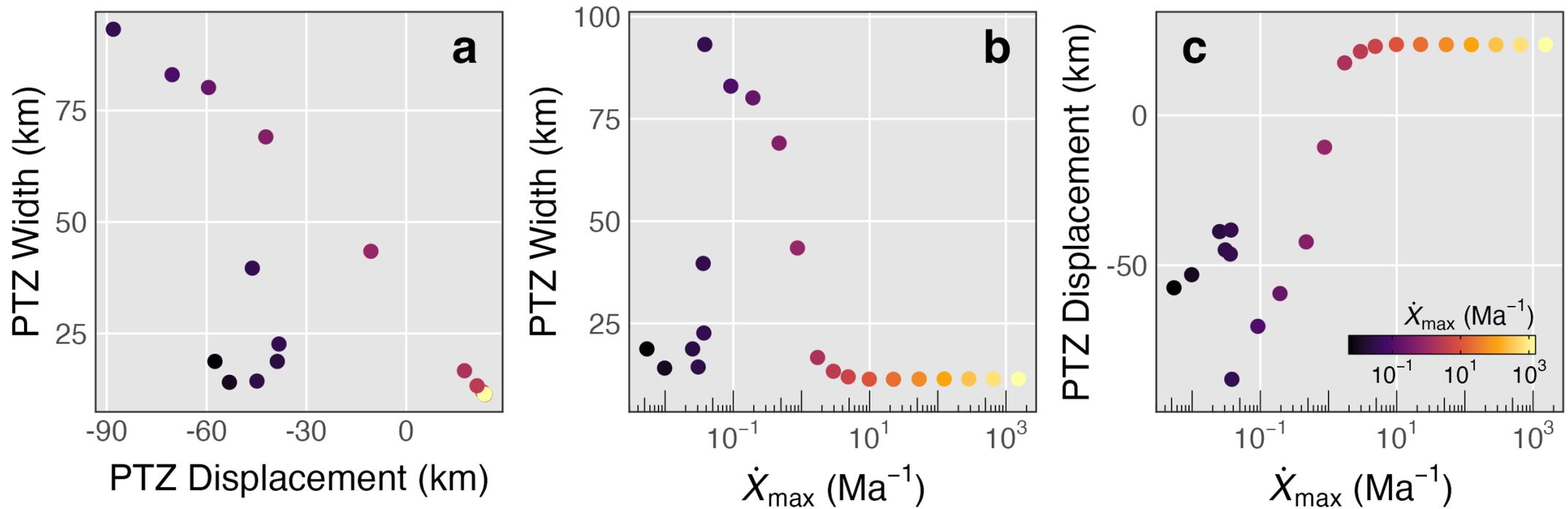
Sluggish kinetics in slabs promote ponding and large metastable olivine regions



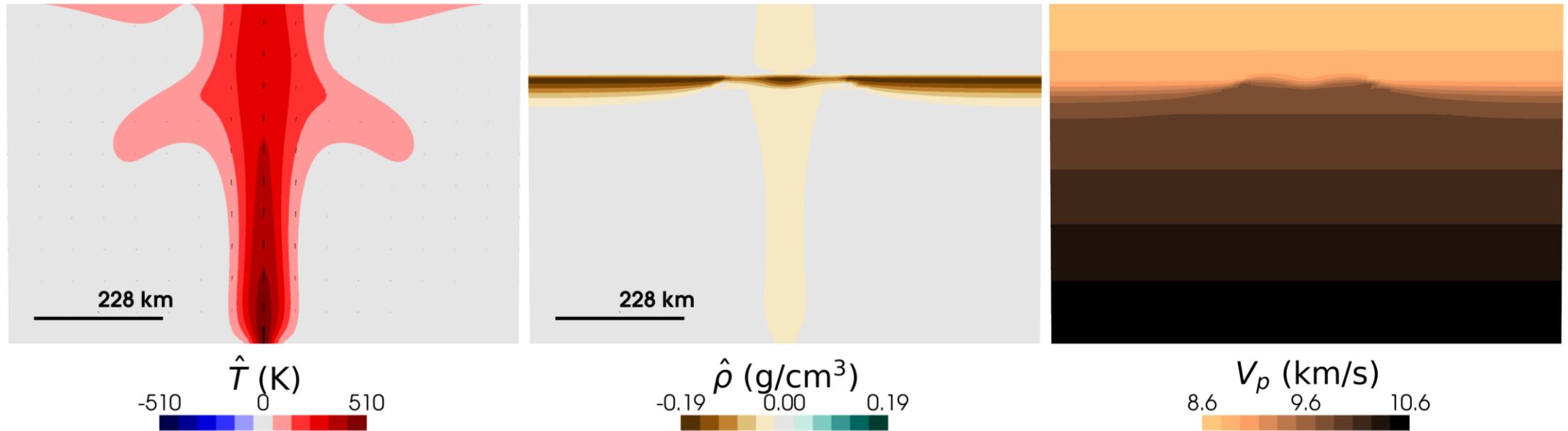
Faster kinetics sharpen the slab phase transition and enable penetration



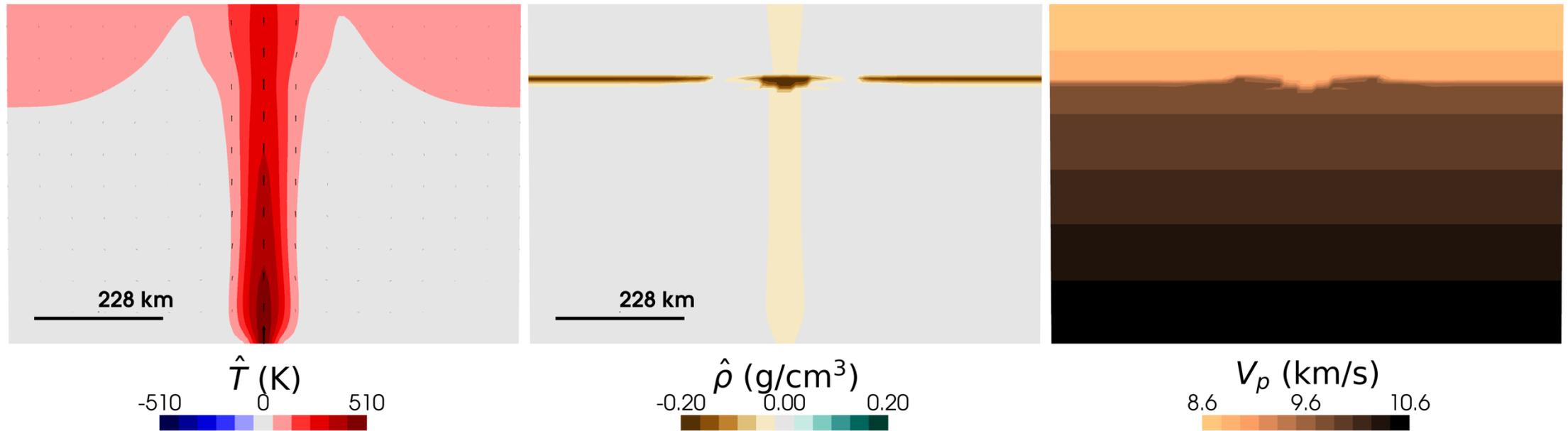
Slabs exhibit log-linear relationships between kinetics and PTZ structure (with thresholds)



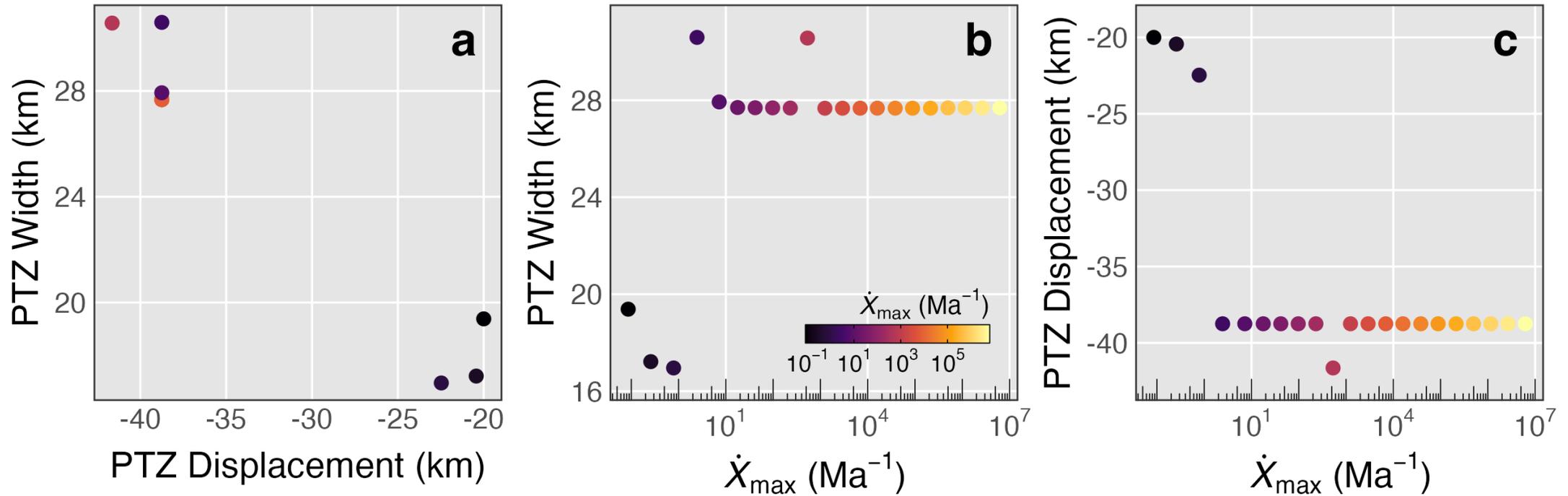
Sluggish kinetics in plumes promote *negligible* stagnation and 410 uplift



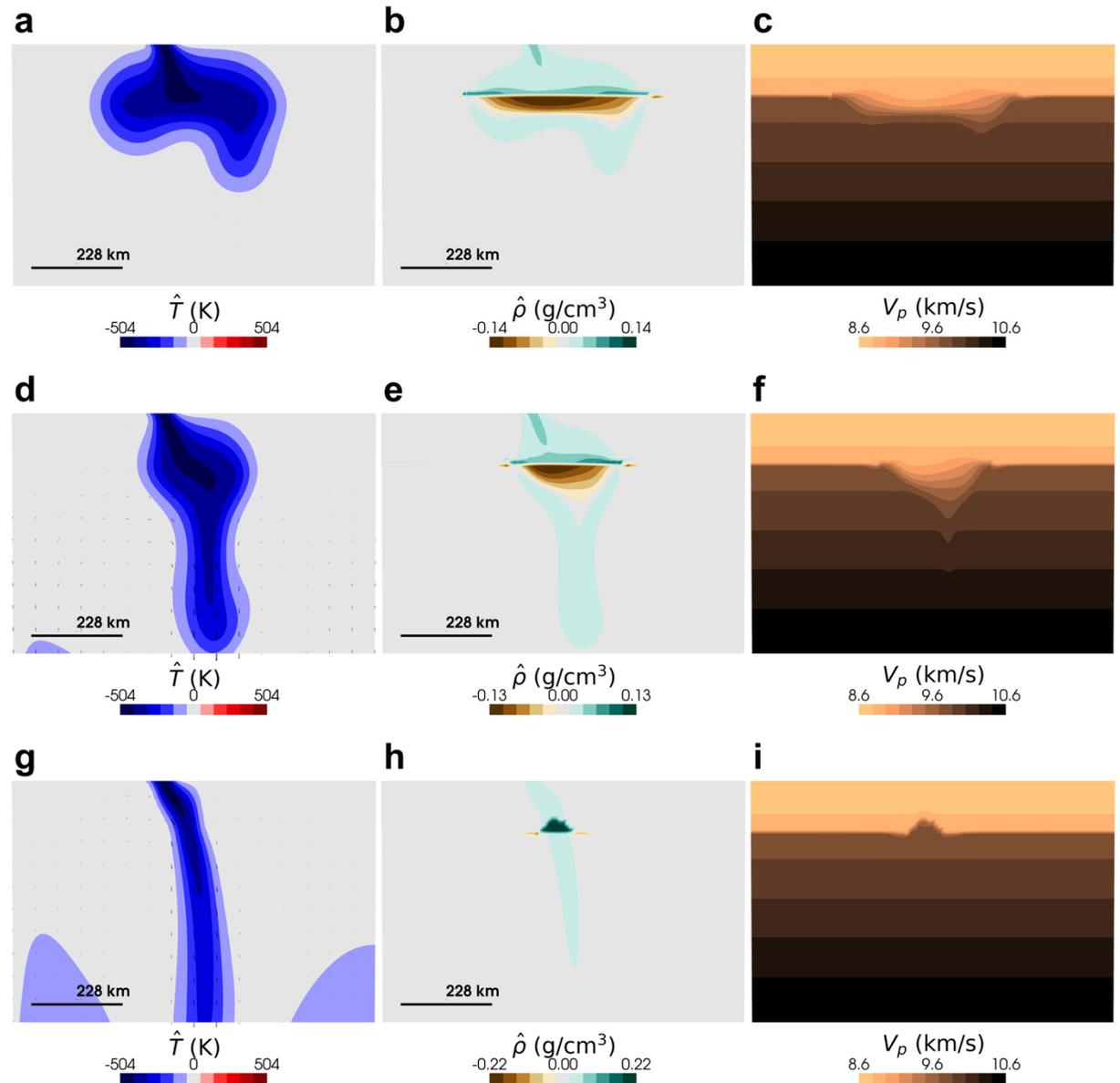
Kinetic rates are always high in plumes: the 410 remains at its predicted equil. depth



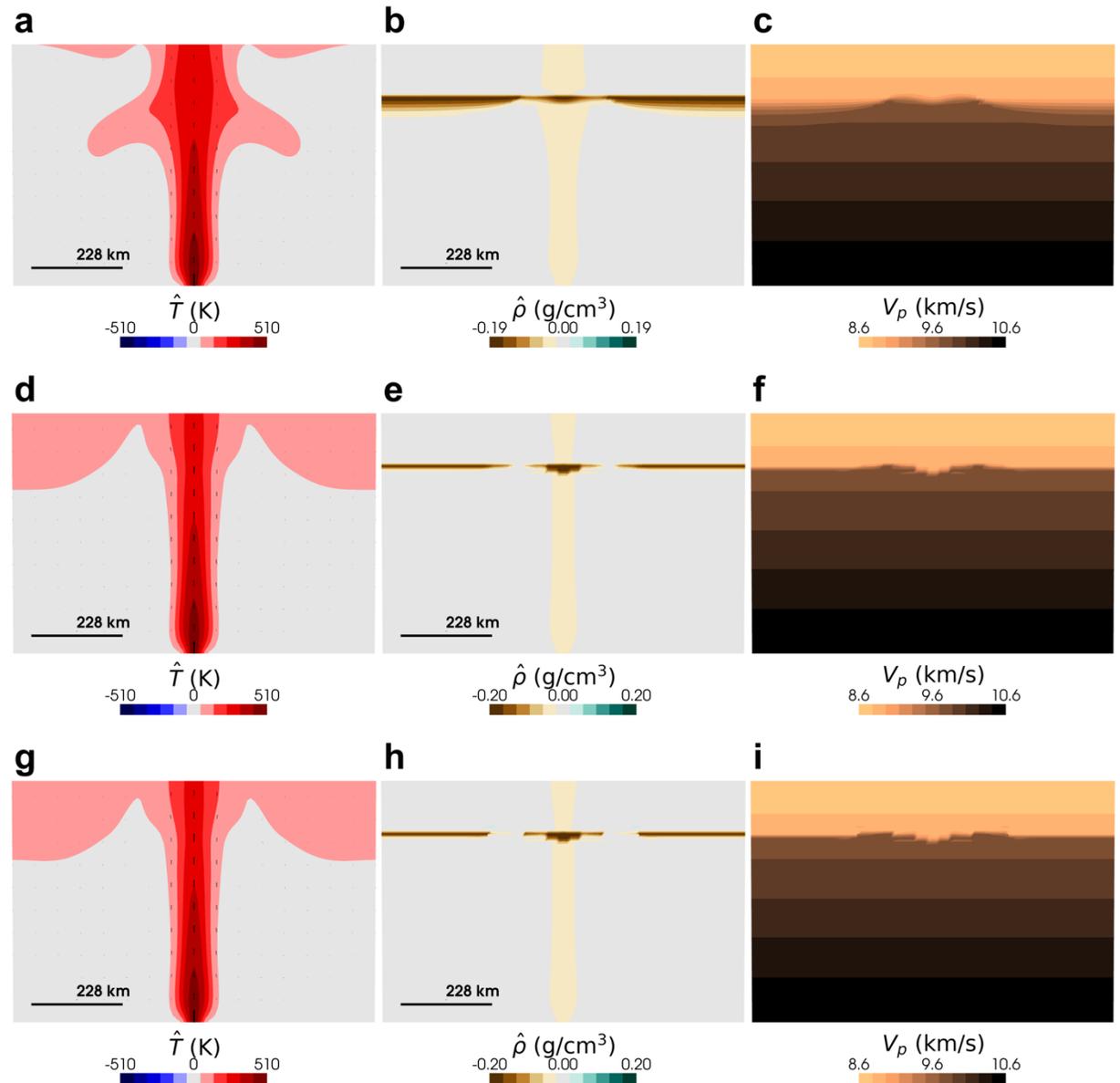
Plumes show no scaling between kinetics and PTZ structure (with a few outliers)



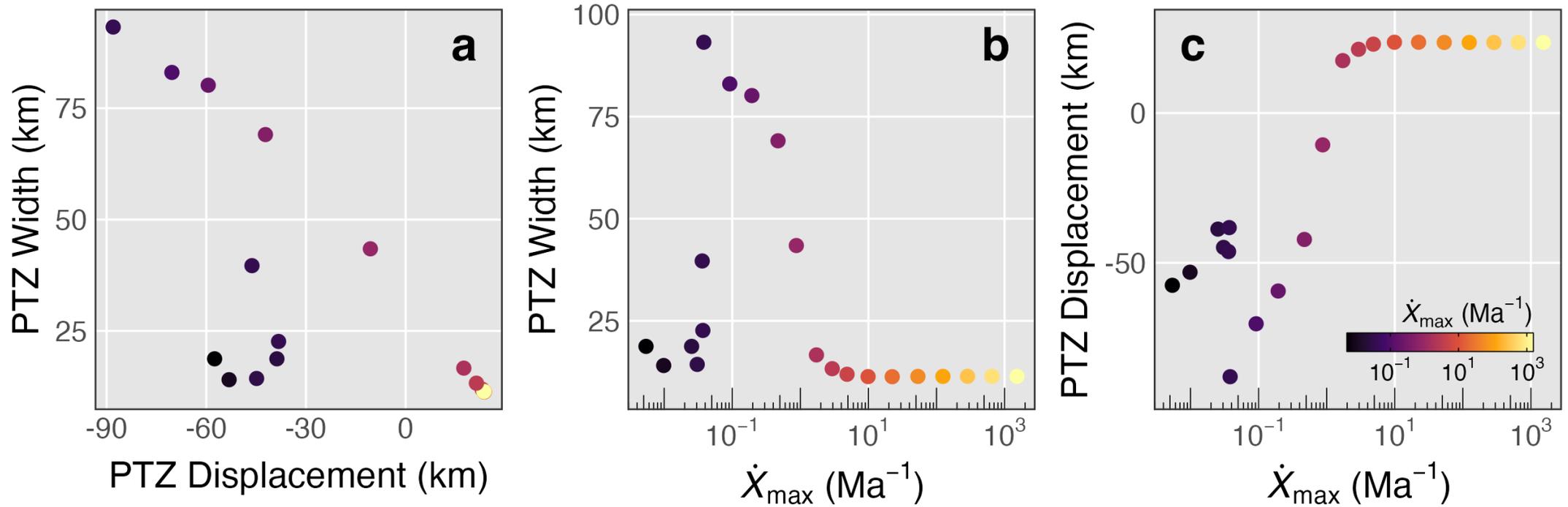
Kinetics explain slab stagnation, deep EQs, and variable seismic signals



Kinetics cannot explain uplifted and/or weak 410 signals near hotspots



Seismic observables can constrain phase transition rates in slabs (?)



Takeaways from Kerswell et al. (in prep)

- 410 is uniform in plumes, but slabs show displacement and fading
- Kinetics can explain slab dynamics and unusual seismic observations
- Kinetics cannot explain unusual seismic signals near plumes
- Seismic discontinuities are probes of phase transition rates