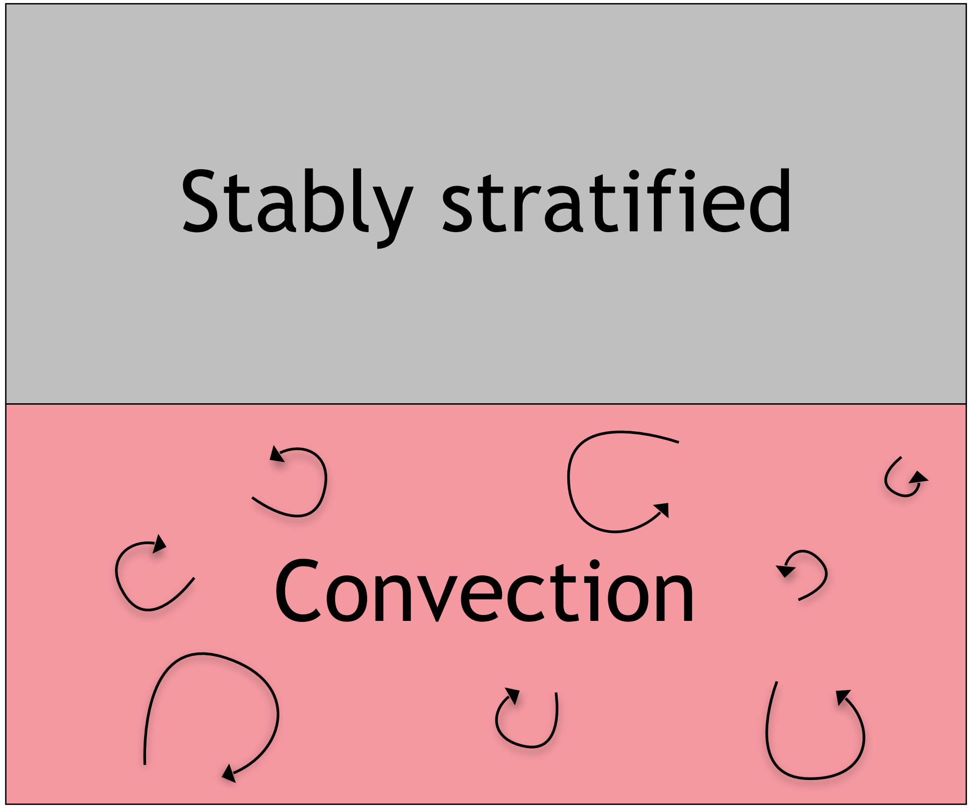
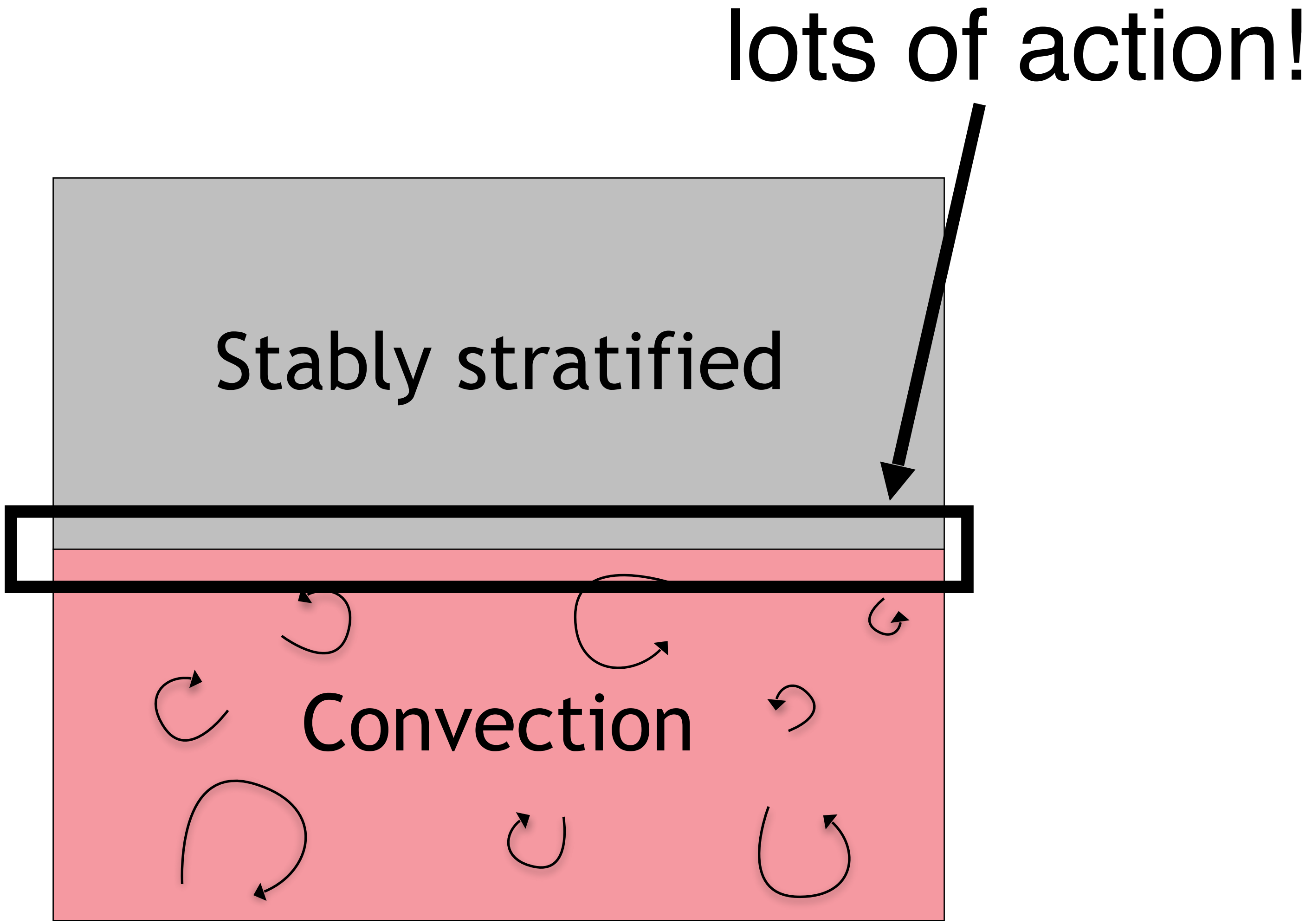
Testing Parameterizations of Convective Overshoot

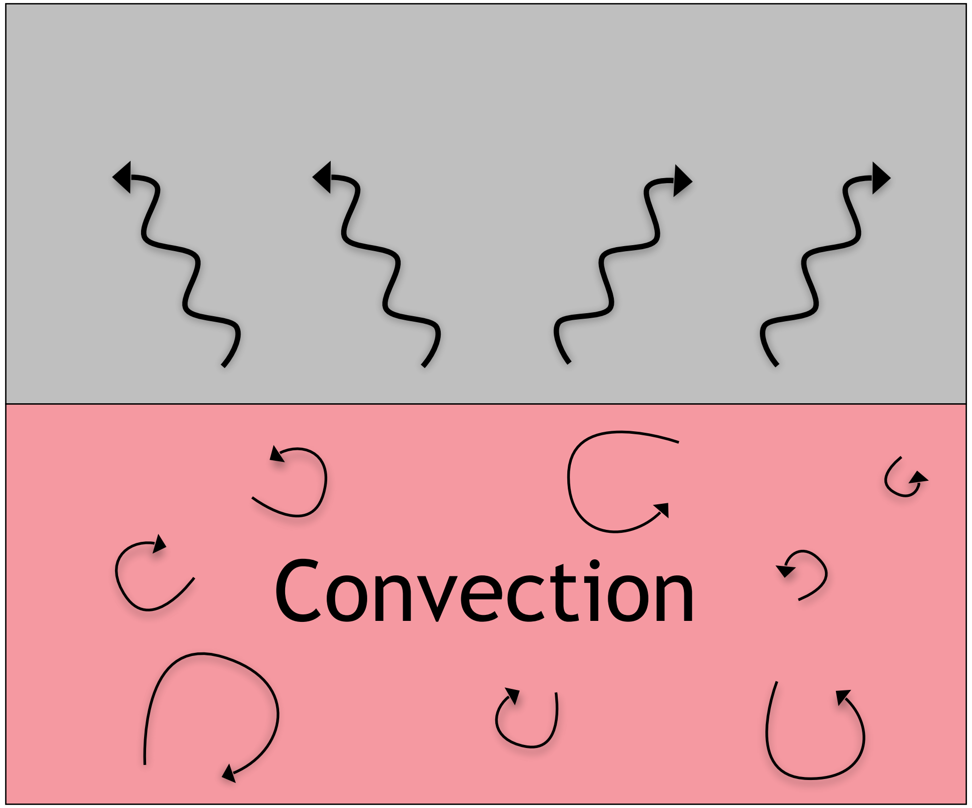
Daniel Lecoanet (Princeton)
Josiah Schwab (UCSC), Eliot Quataert (Berkeley)
Lars Bildsten, Frank Timmes
Geoff Vasil, Keaton Burns
Jeff Oishi, Ben Brown

Supported by: NSF, Hertz Foundation, NASA, ARC

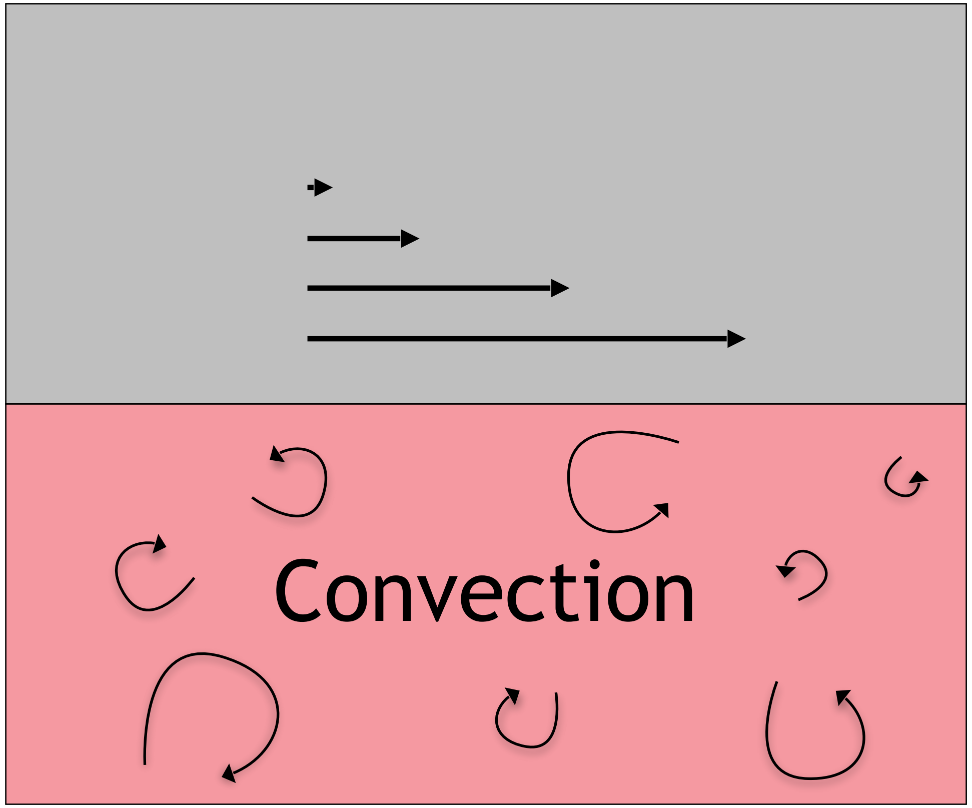

Stably stratified
Convection

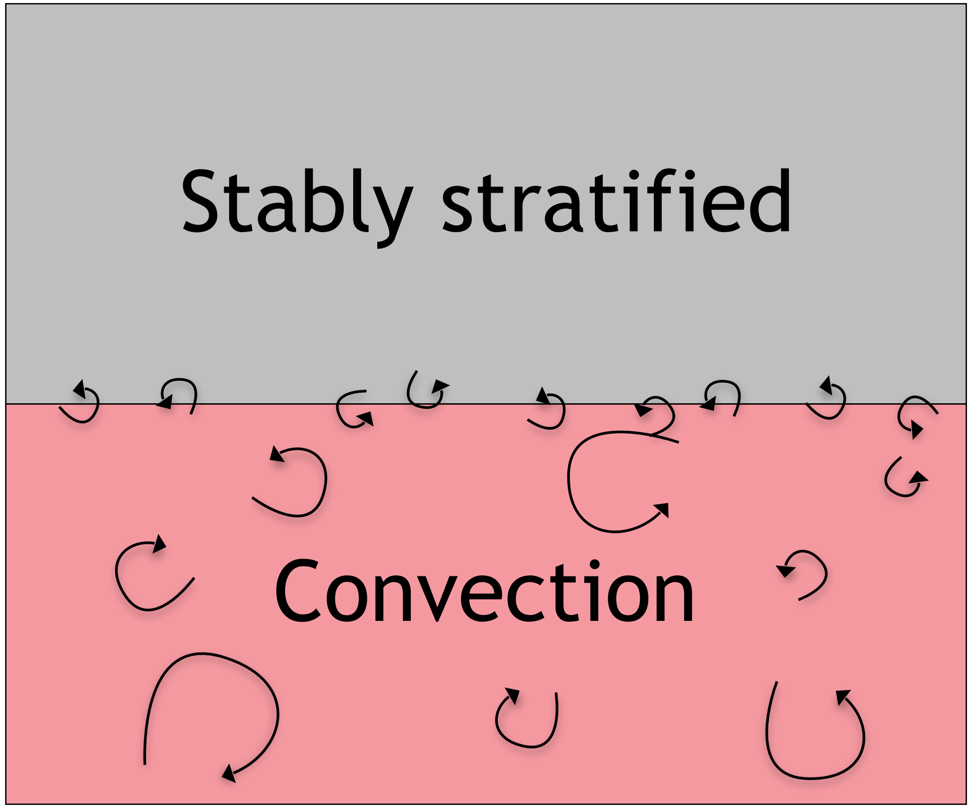

lots of action!
Stably stratified
Convection

Internal gravity waves — how efficient is excitation? what types of waves excited?

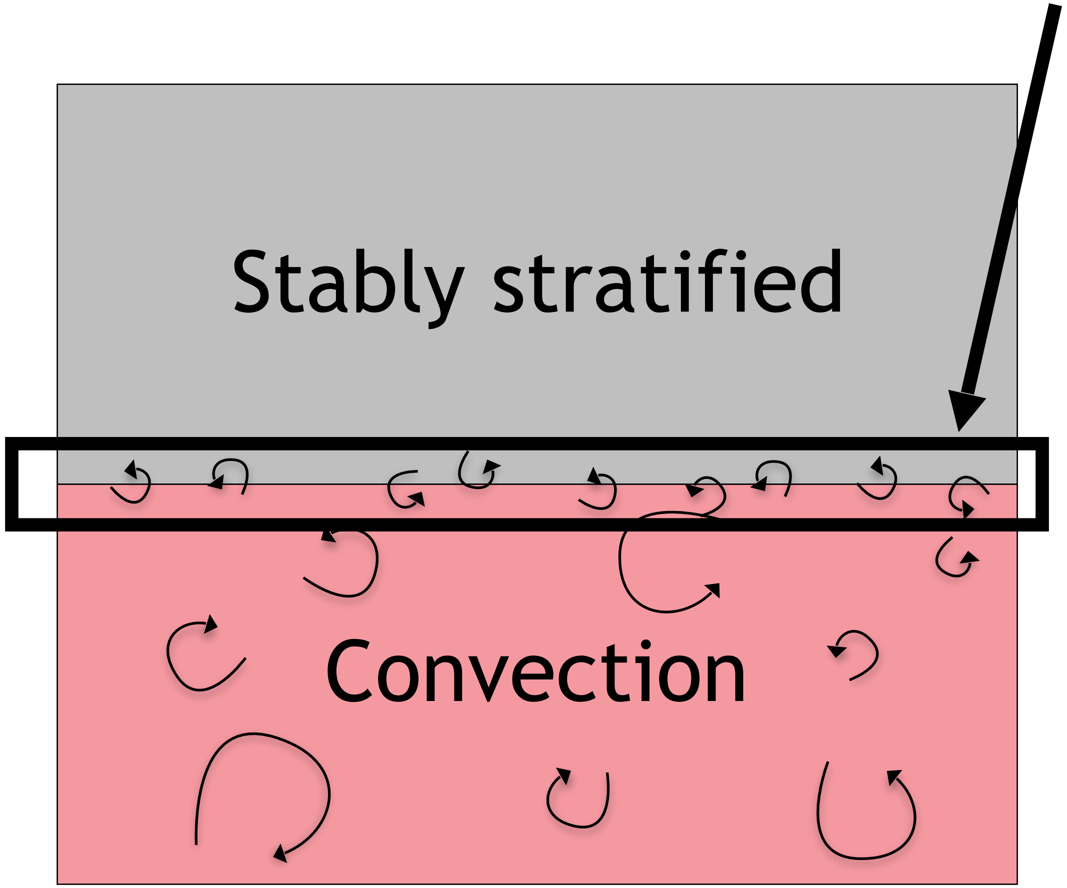


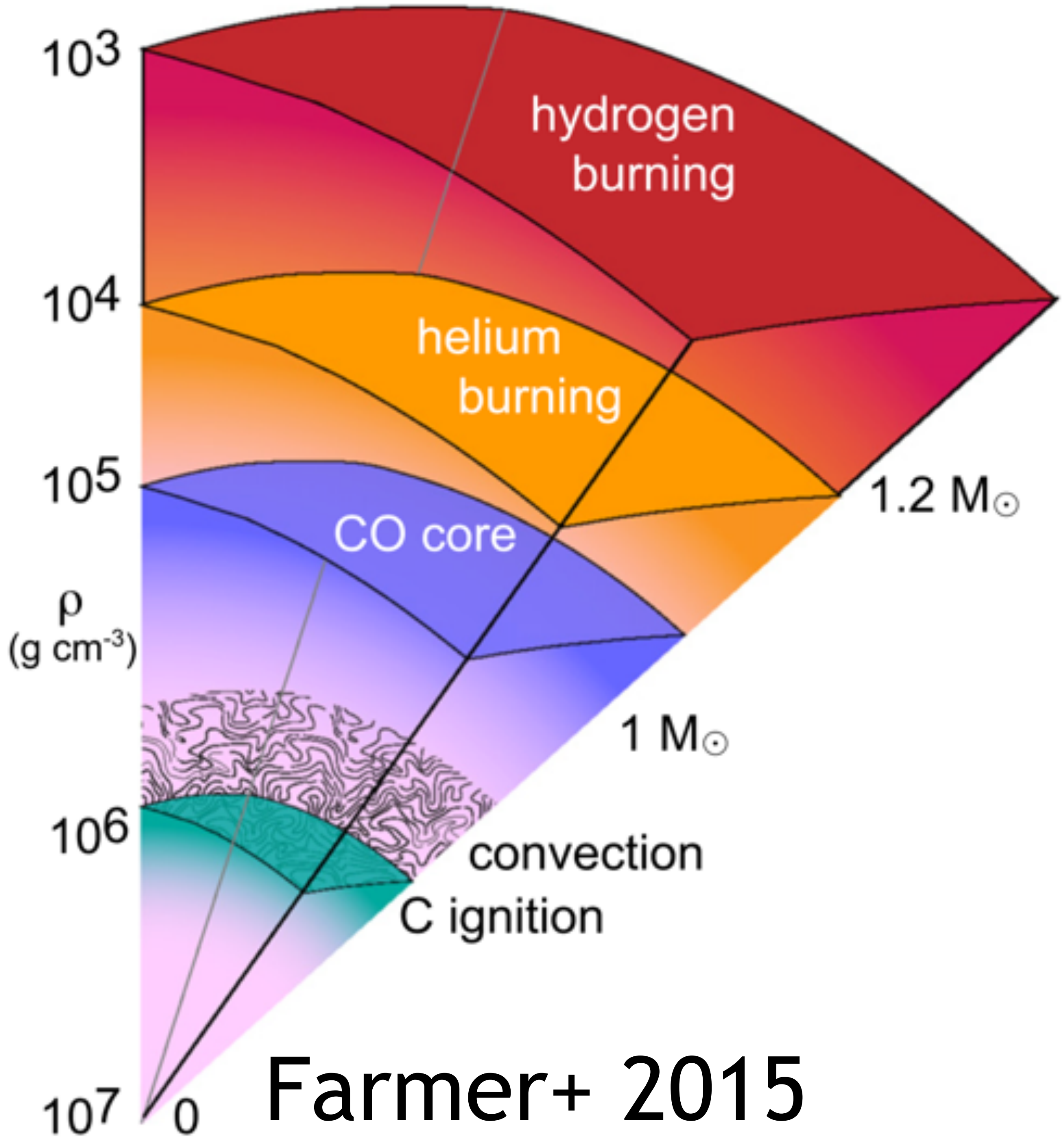
momentum transport: e.g., QBO




Stably stratified
Convection

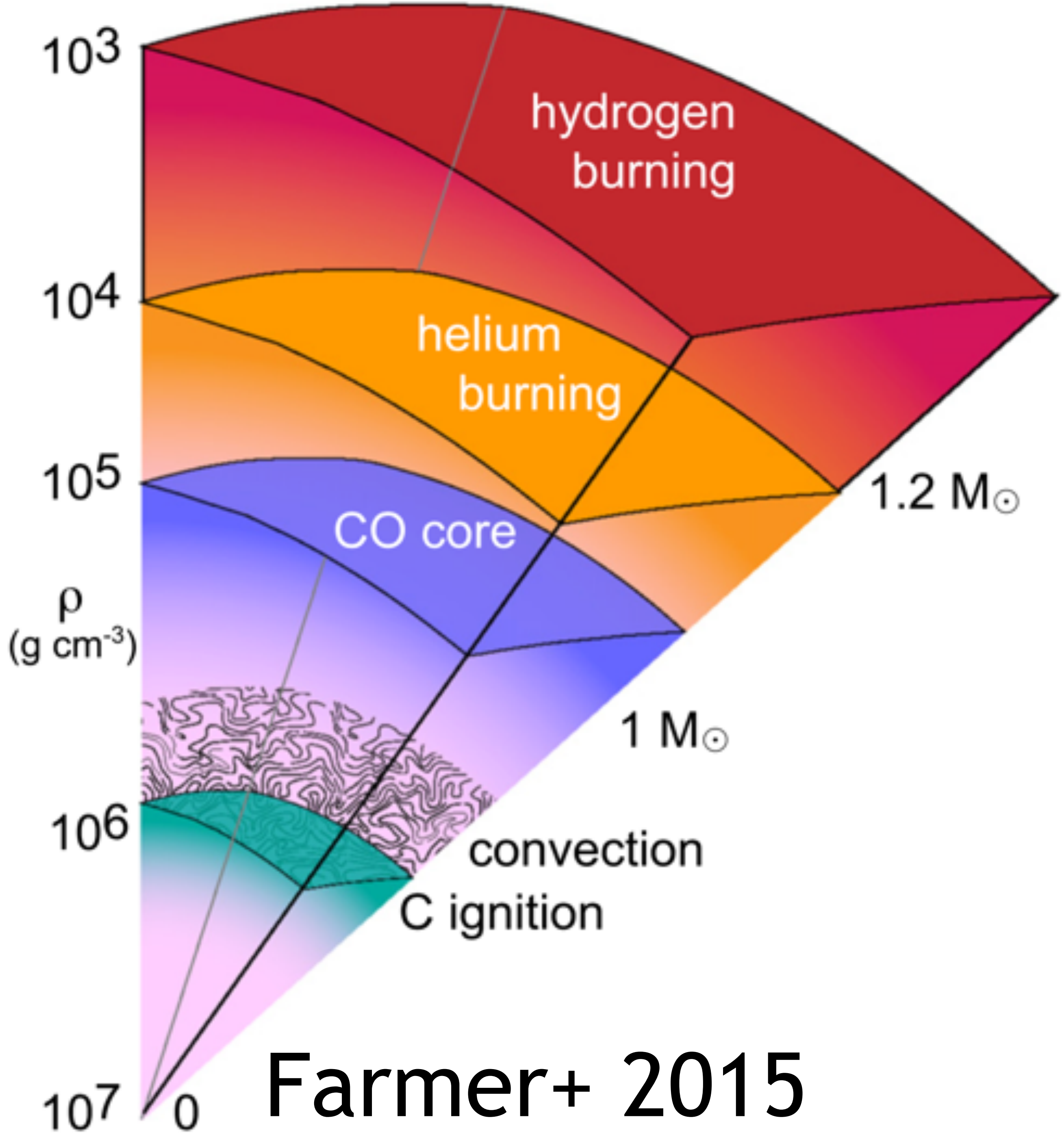
mixing — how to parameterize??




hydrogen burning
helium burning
CO core
convection
C ignition
1.2 M☉
1 M☉
0
10^3
10^4
10^5
10^6
10^7
ρ (g cm^{-3})

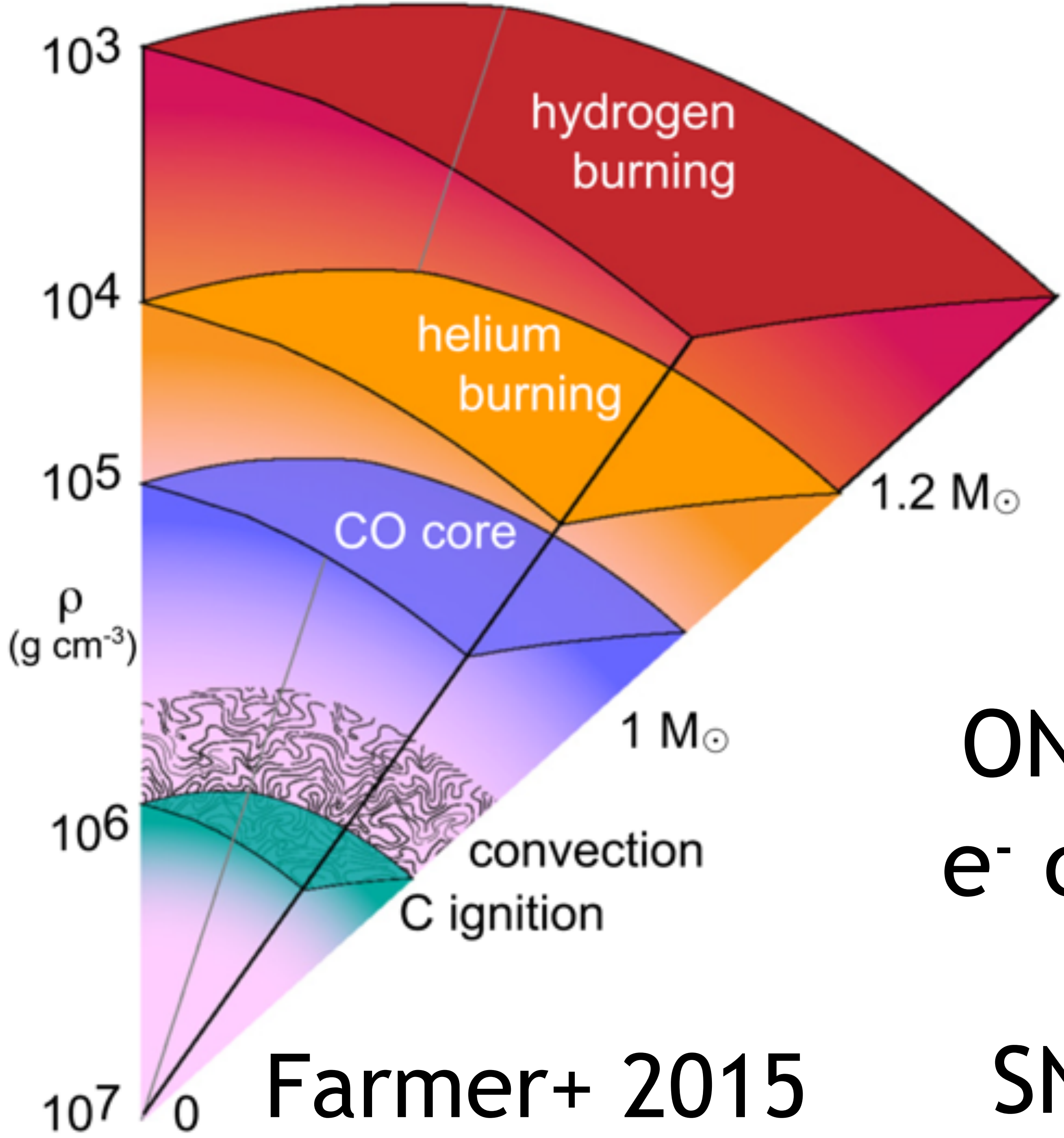
Farmer+ 2015

Does flame reach center???



Farmer+ 2015

Does flame reach center???



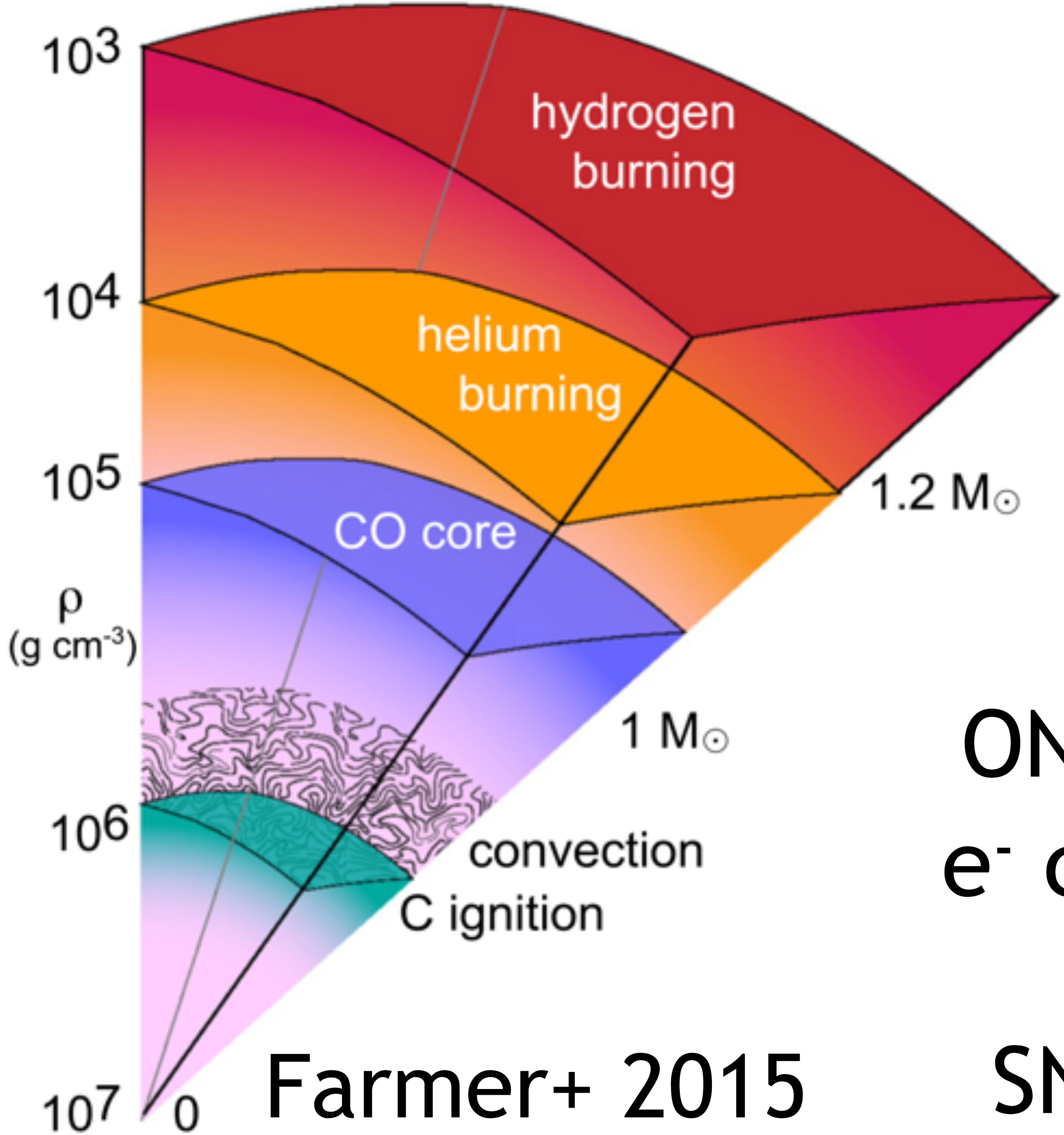
Yes:

ONe core

e$^-$ capture SNe

SNe II-P

Farmer+ 2015

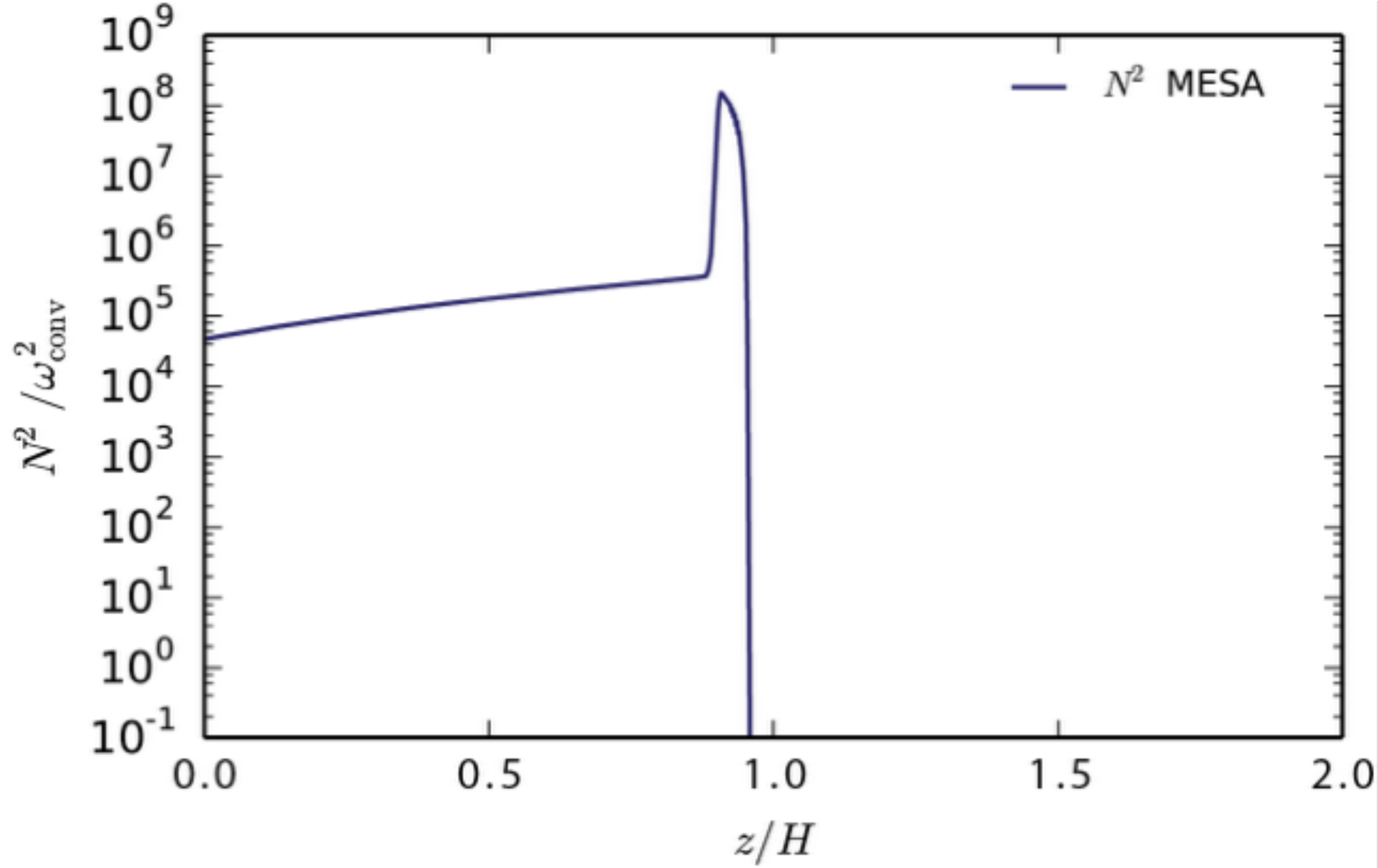


Farmer+ 2015

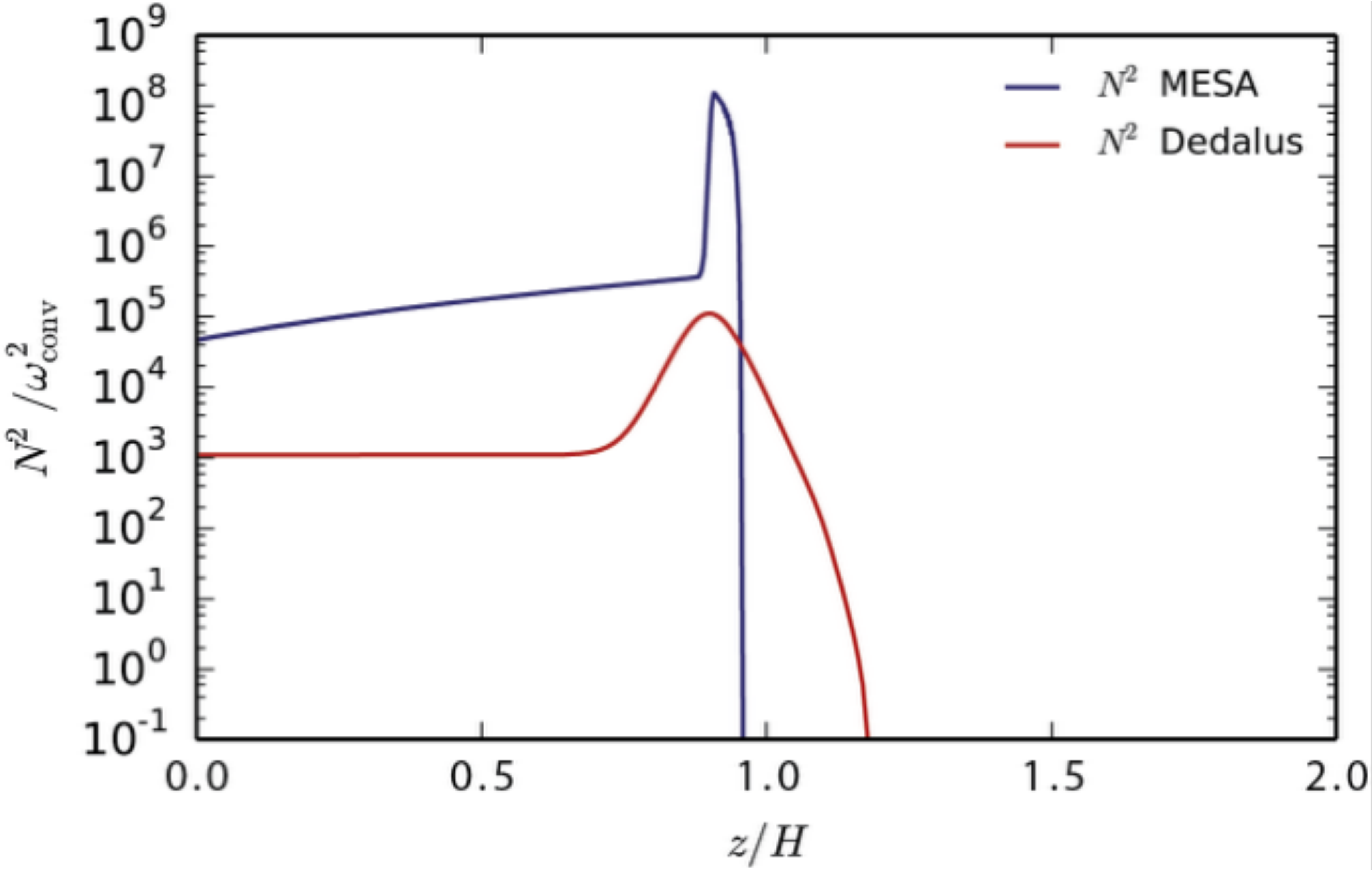
Does flame reach center???

Yes:	No:
ONe core	``Hybrid'' WD
e^- capture SNe	SNe Ia?
SNe II-P	???

Background Properties



Background Properties



Model

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} p = -\boldsymbol{g} \alpha \frac{T}{T_0} + \nu \nabla^2 \boldsymbol{u}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$$

$$\partial_t T + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T$$

Model

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} p = -\boldsymbol{g} \alpha \frac{T}{T_0} + \nu \nabla^2 \boldsymbol{u}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$$

$$\partial_t T + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T$$

$$N^2 = \frac{\alpha g}{T_0} \frac{dT}{dz}$$

negative in conv
VERY positive in stable

Carbon Flames Model

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} p = -\boldsymbol{g}\alpha \frac{T}{T_0} + \nu \nabla^2 \boldsymbol{u}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$$

$$\partial_t T + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T + \bar{H}$$

$$\bar{H} = -\kappa \nabla^2 T_{\mathrm{background}}$$

Carbon Flames Model

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} p = -\boldsymbol{g} \alpha \frac{T}{T_0} + \nu \nabla^2 \boldsymbol{u}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$$

$$\partial_t T + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T + \bar{H}$$

$$\bar{H} = -\kappa \nabla^2 T_{\mathrm{background}}$$

BC’s: no slip, no temperature perturbation

3D domain, 4H x 4H x 2H

Carbon Flames Model

$$\partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla} p = -\boldsymbol{g}\alpha\frac{T}{T_0} + \nu\nabla^2\boldsymbol{u}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$$

$$\partial_t T + \boldsymbol{u}\cdot\boldsymbol{\nabla}T = \kappa\nabla^2 T + \bar{H}$$

$$\bar{H} = -\kappa\nabla^2 T_{\mathrm{background}}$$

$$\mathrm{Ra} = \frac{g\alpha\Delta T H^3}{T_0\nu\kappa}$$

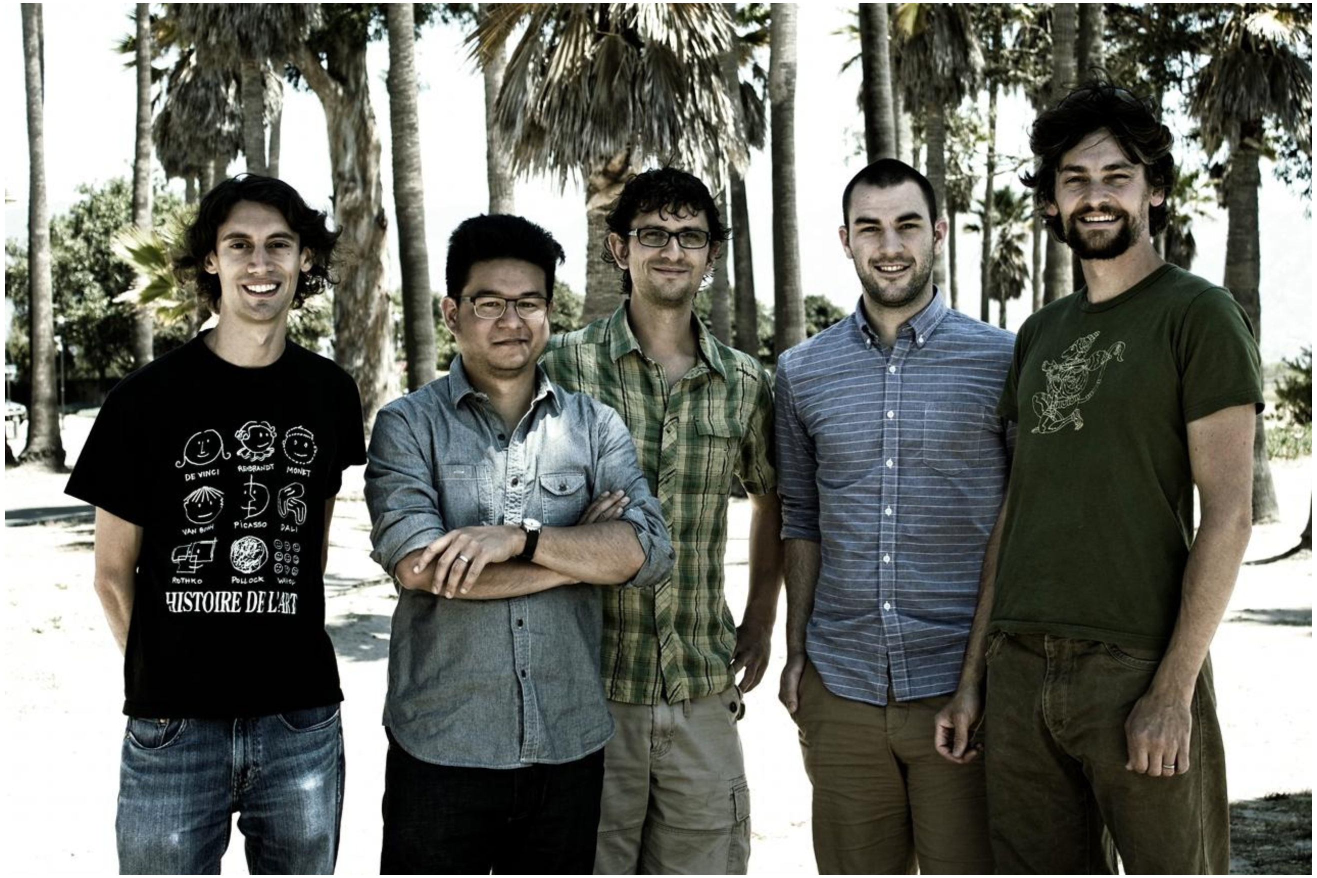
$$\mathrm{Pr} = \frac{\nu}{\kappa}$$

BC’s: no slip, no temperature perturbation

3D domain, 4H x 4H x 2H


DEDALUS
A FLEXIBLE FRAMEWORK FOR SPECTRALLY SOLVING DIFFERENTIAL EQUATIONS
LEARN MORE
dedalus-project.org

The team so far



Daniel Lecoanet (Princeton)   Keaton Burns (MIT)

Jeff Oishi (Bates)   Ben Brown (Colorado)

Geoff Vasil (Sydney)

Australian Government

Australian Research Council

NSF

$$\partial_t u - D\nabla^2 u = Ru(1-u)$$

```
import numpy as np
import dedalus.public as de

# Bases: names, modes, intervals, dealiasing
x_basis = de.Fourier('x', 128,
    interval=(0, 2*np.pi), dealias=3/2)
y_basis = de.Chebyshev('y', 64,
    interval=(0, 1), dealias=3/2)
# Domain: bases, datatype
domain = de.Domain([x_basis, y_basis], float)
```

```
# Problem: domain and variables
prob = de.IVP(domain, variables=['u','uy'])
# Fixed parameters
prob.parameters['D'] = 0.1
prob.parameters['R'] = 1
# Parsing substitutions
prob.substitutions['Lu'] = "dx(dx(u)) + dy(uy)"
# First-order reduction
prob.add_equation("uy - dy(u) = 0")
# Fisher-KPP equation
prob.add_equation("dt(u) - D*Lu = R*u*(1-u)")
# Neumann boundary conditions
prob.add_bc("left(uy) = 0")
prob.add_bc("right(uy) = 0")
```

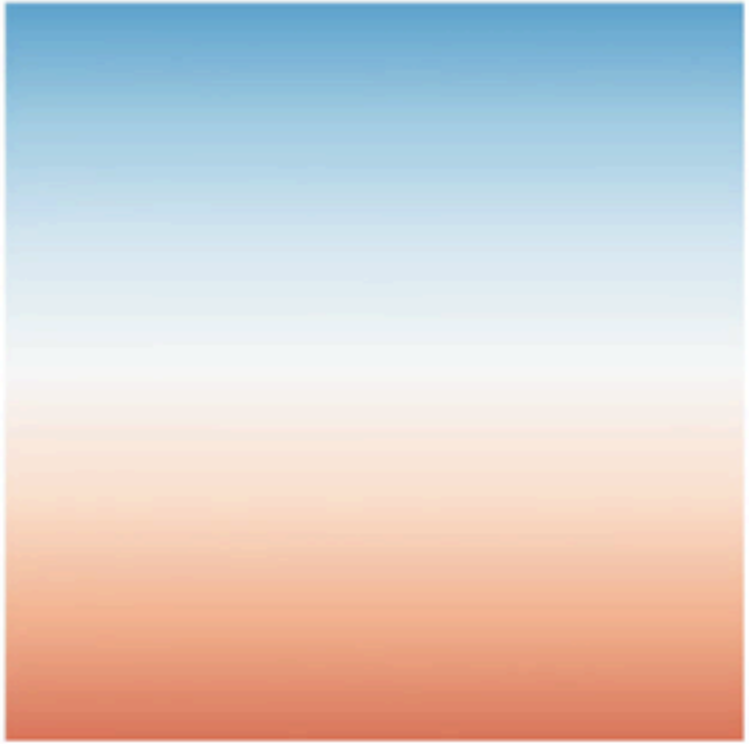
$$\partial_t u - D\nabla^2 u = Ru(1-u)$$

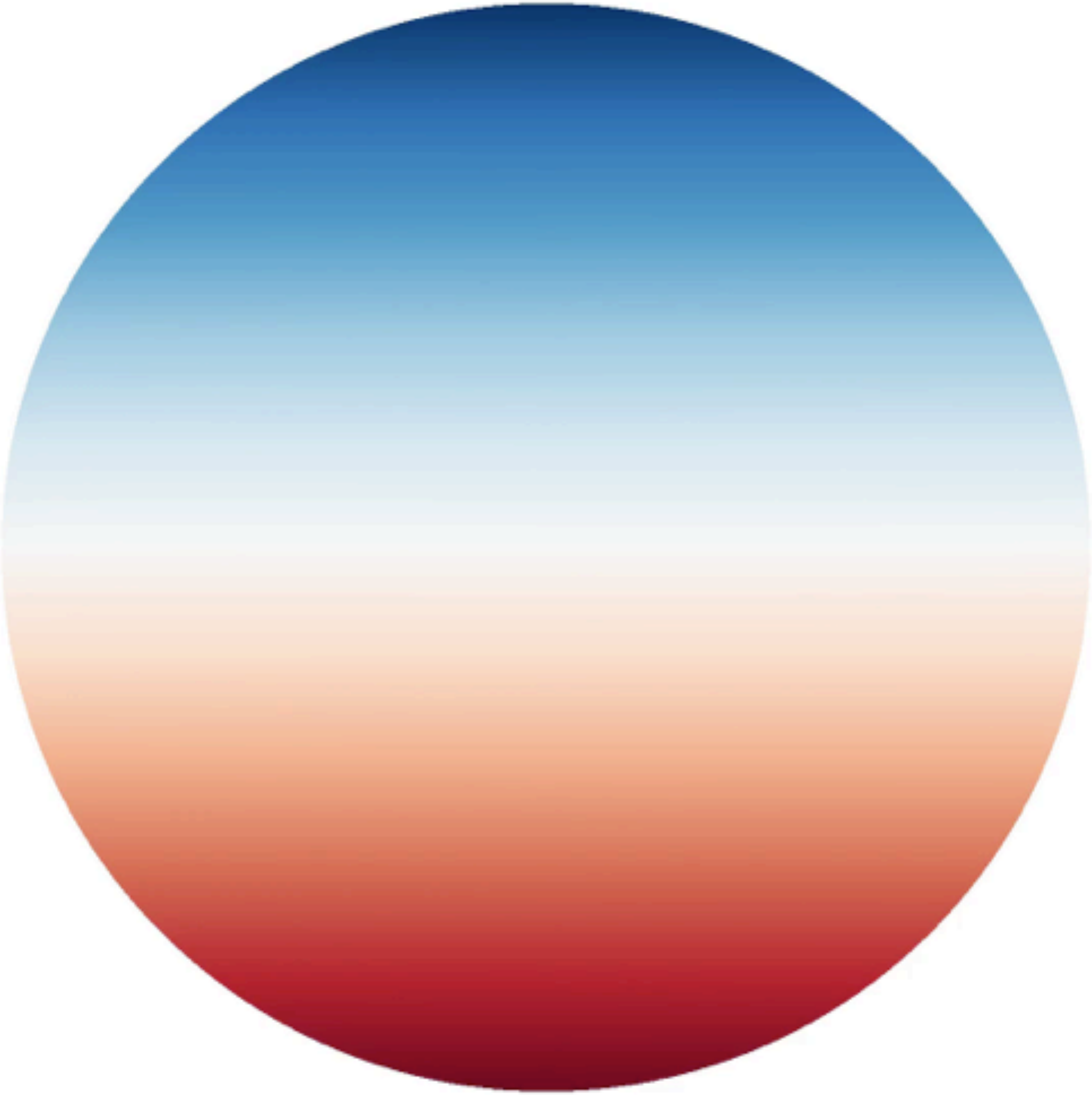
```
# Pick a timestepper
ts = de.timesteppers.RK222
# Build solver
solver = prob.build_solver(ts)
# Set integration limits
solver.stop_sim_time = 10
# Set initial conditions
u = solver.state['u']
u['g'] = np.random.rand(*u['g'].shape)
```

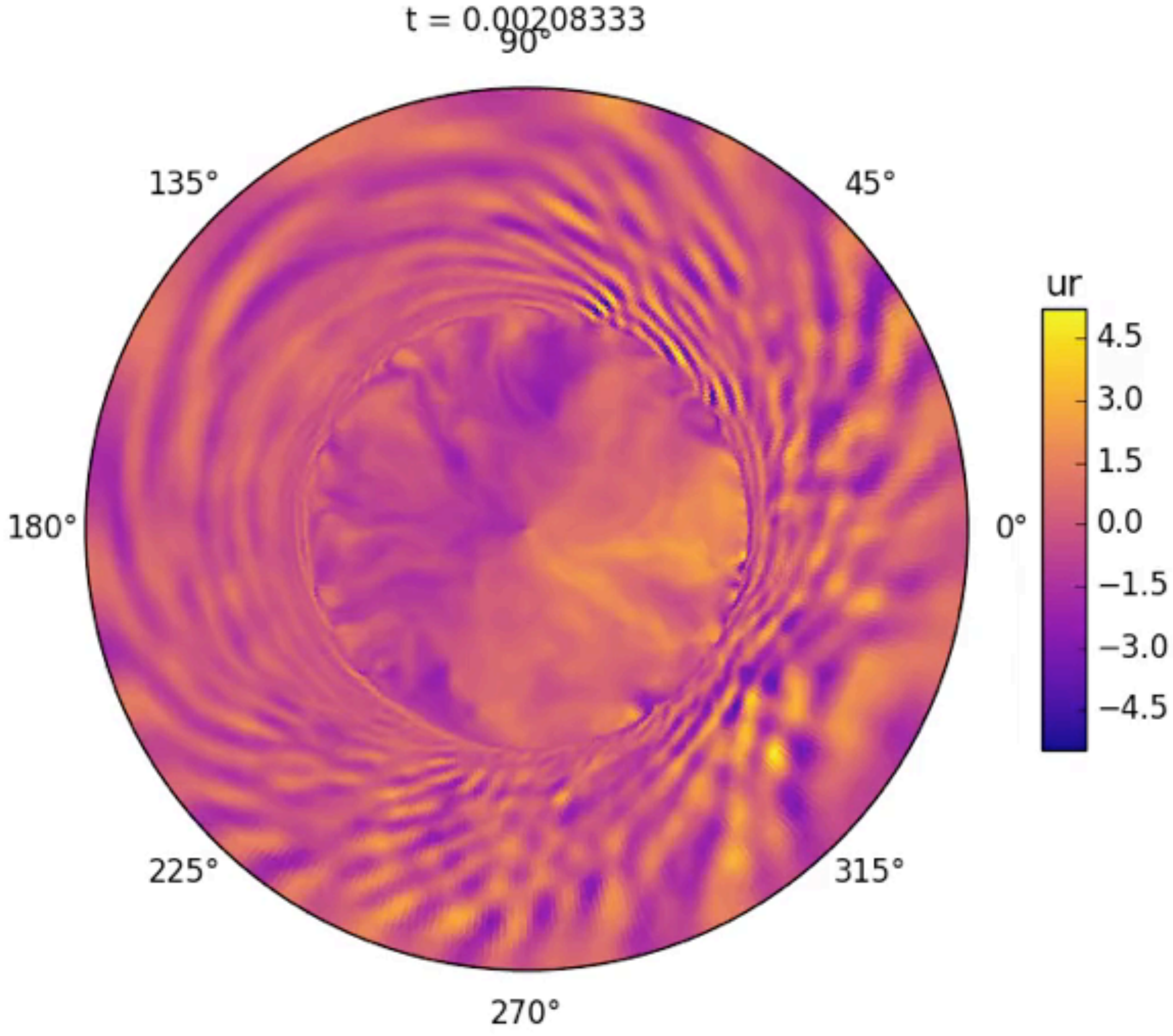
```
dt = 0.01
# Main loop chceking stopping criteria
while solver.ok:
    # Step forward
    solver.step(dt)
    # Perform some analysis
    print(np.mean(u['g']), np.std(u['g']))
```

Dedalus code

```
problem.add_equation("dx(u) + dy(v) + dz(w) = 0")

problem.add_equation("dt(u) − ν*(dx(dx(u)) + dy(dy(u)) + dz(uz)) + dx(p)
                                            = −(u*dx(u) + v*dy(u) + w*uz)")

problem.add_equation("dt(v) − ν*(dx(dx(v)) + dy(dy(v)) + dz(vz)) + dy(p)
                                            = −(u*dx(v) + v*dy(v) + w*vz)")

problem.add_equation("dt(w) − ν*(dx(dx(w)) + dy(dy(w)) + dz(wz)) + dz(p) − b
                                            = −(u*dx(w) + v*dy(w) + w*wz)")

problem.add_equation("dt(b) − κ*(dx(dx(b)) + dy(dy(b)) + dz(bz))
                                         = −(u*dx(b) + v*dy(b) + w*bz) − H")

problem.add_equation("bz − dz(b) = 0")
problem.add_equation("uz − dz(u) = 0")
problem.add_equation("vz − dz(v) = 0")
problem.add_equation("wz − dz(w) = 0")
```

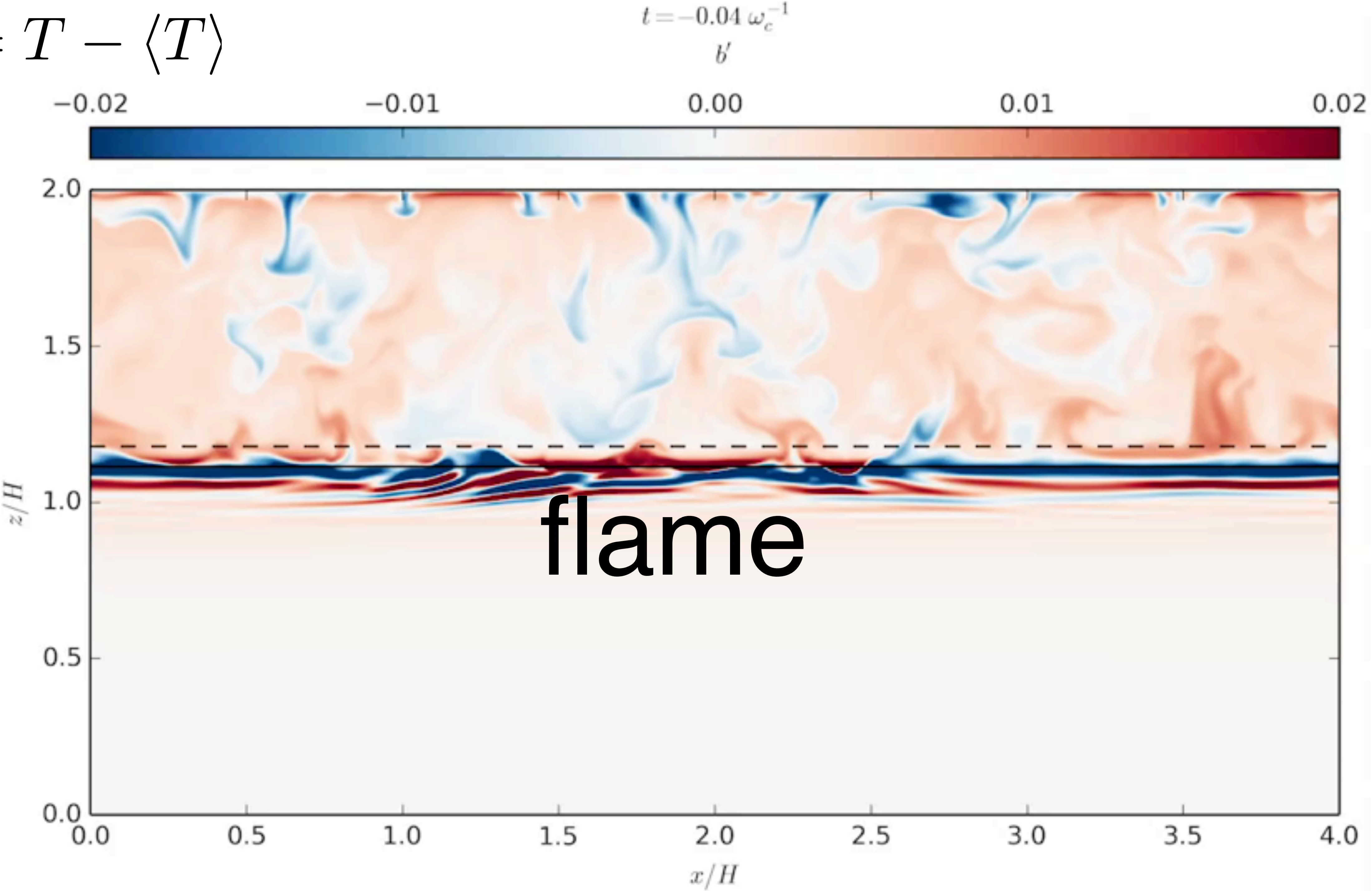




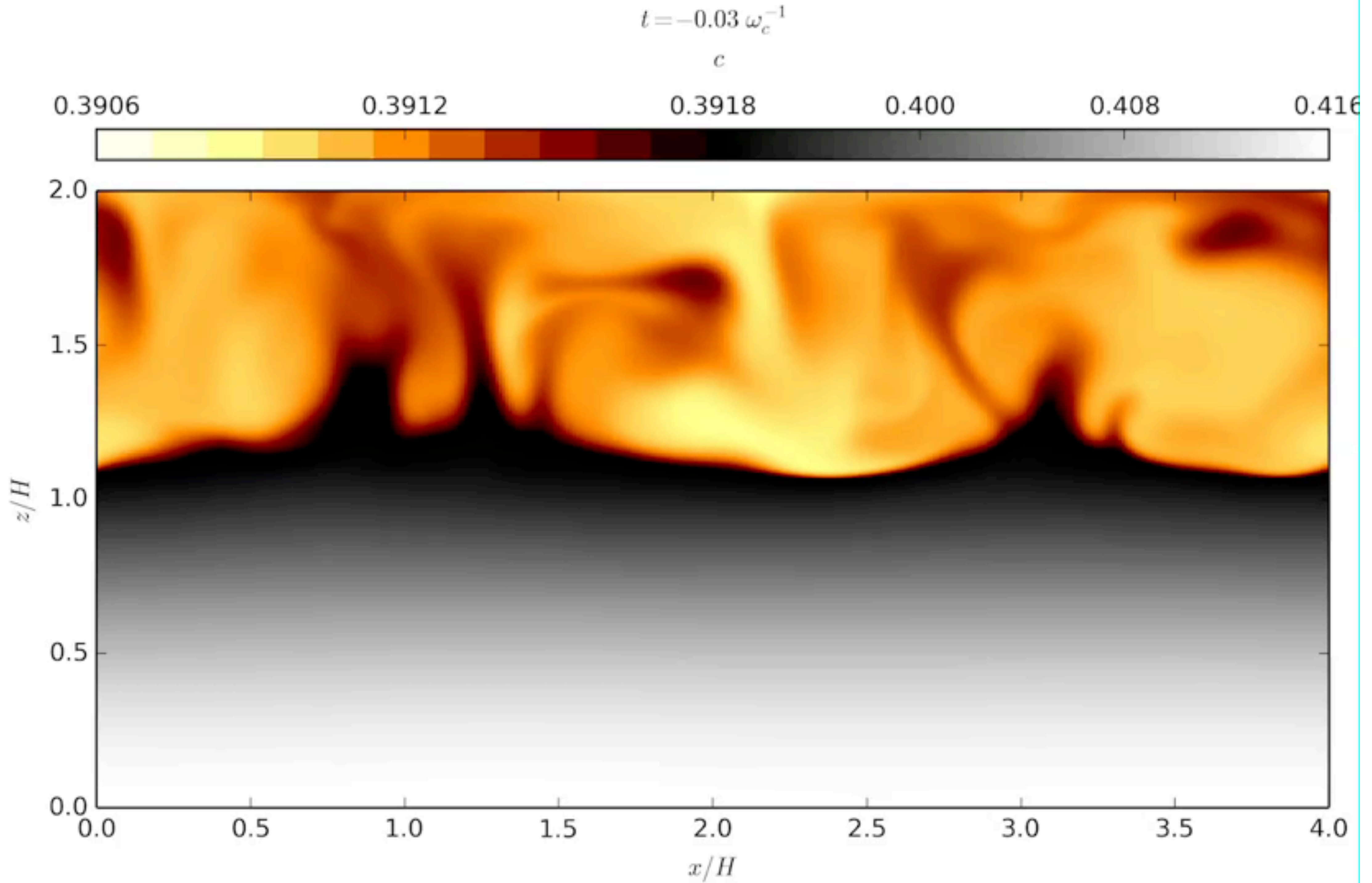


t = 0.00208333
90°
45°
0°
315°
270°
225°
180°
135°
ur
4.5
3.0
1.5
0.0
−1.5
−3.0
−4.5

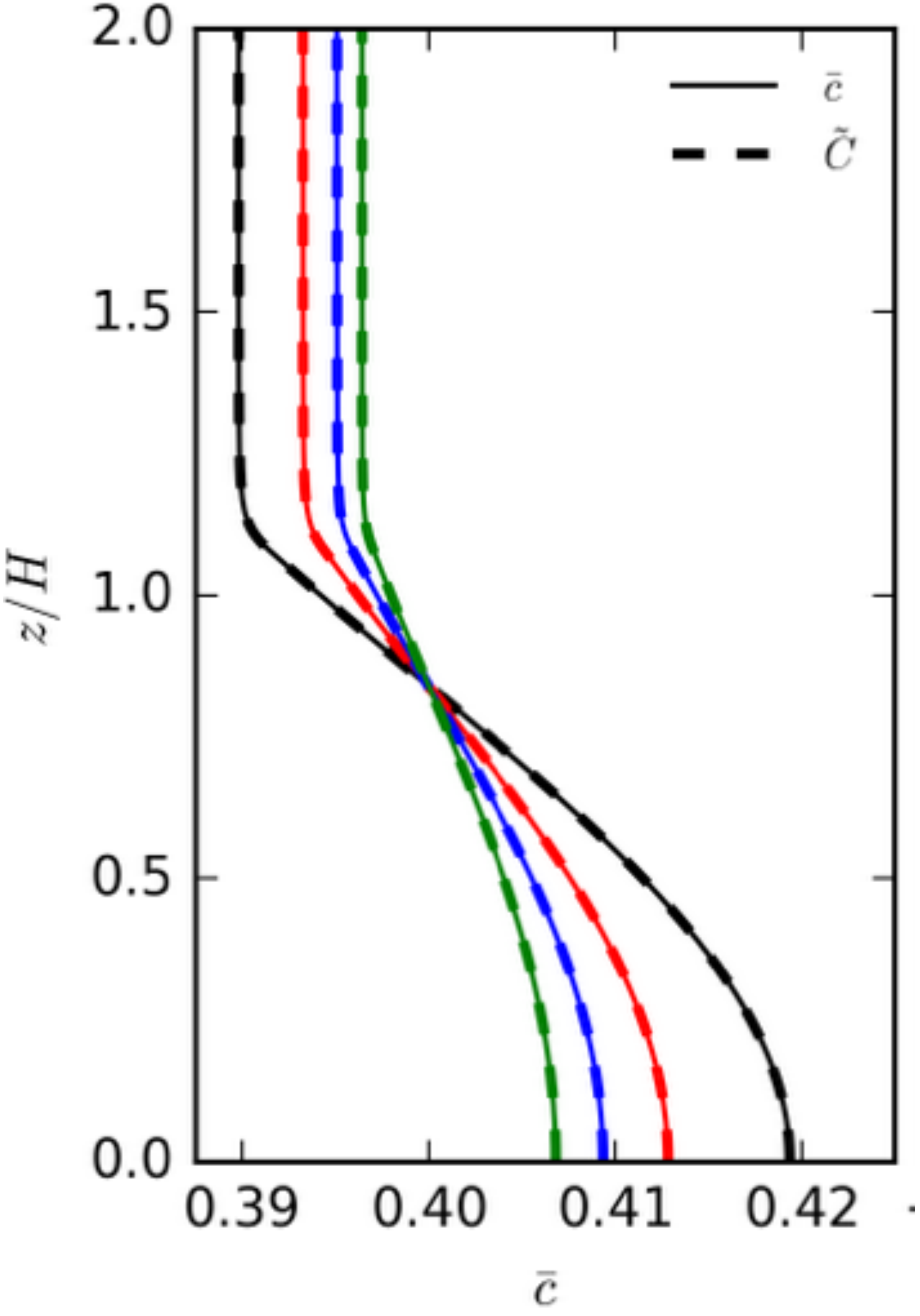
$$b' = T - \langle T \rangle$$



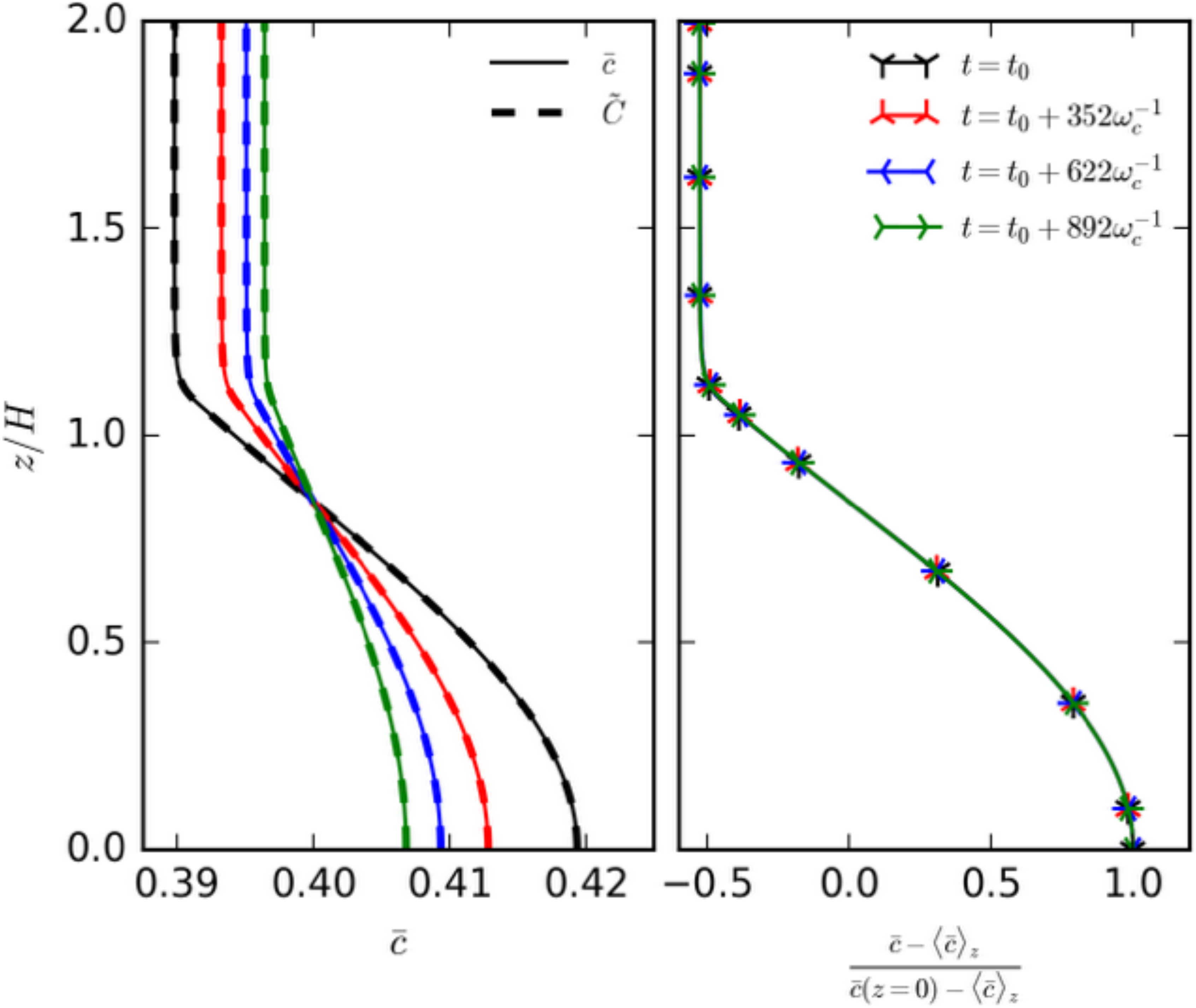
$$\partial_t c + \boldsymbol{u} \cdot \boldsymbol{\nabla} c = D \nabla^2 c$$



$t = -0.03\ \omega_c^{-1}$
c
0.3906
0.3912
0.3918
0.400
0.408
0.416
2.0
1.5
1.0
0.5
0.0
z/H
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x/H



2.0
1.5
1.0
0.5
0.0
z/H
0.39
0.40
0.41
0.42
$\bar{c}$
$\bar{c}$
$\tilde{C}$



2.0
1.5
1.0
0.5
0.0
z/H
0.39
0.40
0.41
0.42
$\bar{c}$
$\tilde{C}$
$\bar{c}$
−0.5
0.0
0.5
1.0
$t = t_0$
$t = t_0 + 352\omega_c^{-1}$
$t = t_0 + 622\omega_c^{-1}$
$t = t_0 + 892\omega_c^{-1}$
$\frac{\bar{c} - \langle \bar{c} \rangle_z}{\bar{c}(z=0) - \langle \bar{c} \rangle_z}$

Passive Scalar Field

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u} \cdot \boldsymbol{\nabla} c$$

Passive Scalar Field

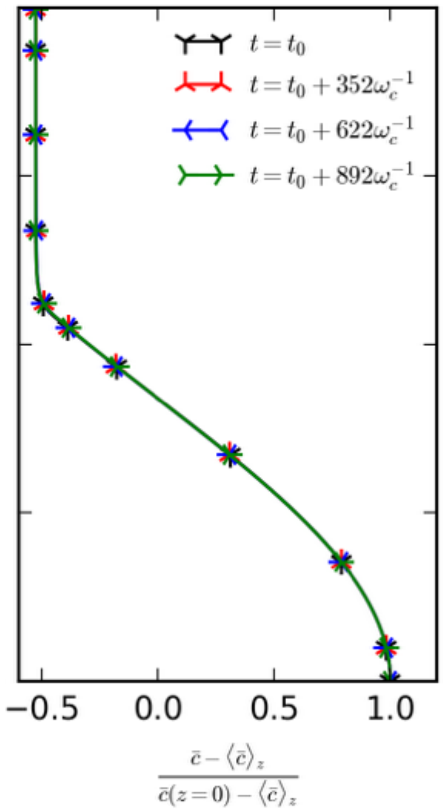
$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$\bar{c}(z,t) - \langle\bar{c}\rangle_z \to c_{\mathrm{s}}\,(z,t) = A(t)C(z)$$

Passive Scalar Field

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$\bar{c}(z,t) - \langle \bar{c} \rangle_z \to c_{\mathrm{s}}\ (z,t) = A(t)C(z)$$



Passive Scalar Field

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$\bar{c}(z,t) - \langle\bar{c}\rangle_z \to c_{\mathrm{s}}\ (z,t) = A(t)C(z)$$

$$-\lambda C - D\partial_z^2 C = -\left\langle \boldsymbol{u}\cdot\boldsymbol{\nabla}\frac{c}{A}\right\rangle_{x,y,t}$$

Passive Scalar Field

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$\bar{c}(z,t) - \langle \bar{c} \rangle_z \to c_{\mathrm{s}}\ (z,t) = A(t)C(z)$$

$$-\lambda C - D\partial_z^2 C = -\left\langle \boldsymbol{u}\cdot\boldsymbol{\nabla}\frac{c}{A}\right\rangle_{x,y,t}$$

Ansatz:

$$-\left\langle \boldsymbol{u}\cdot\boldsymbol{\nabla}\frac{c}{A}\right\rangle_{x,y,t} = \partial_z(D_{\mathrm{t}}\partial_z C)$$

Passive Scalar Field

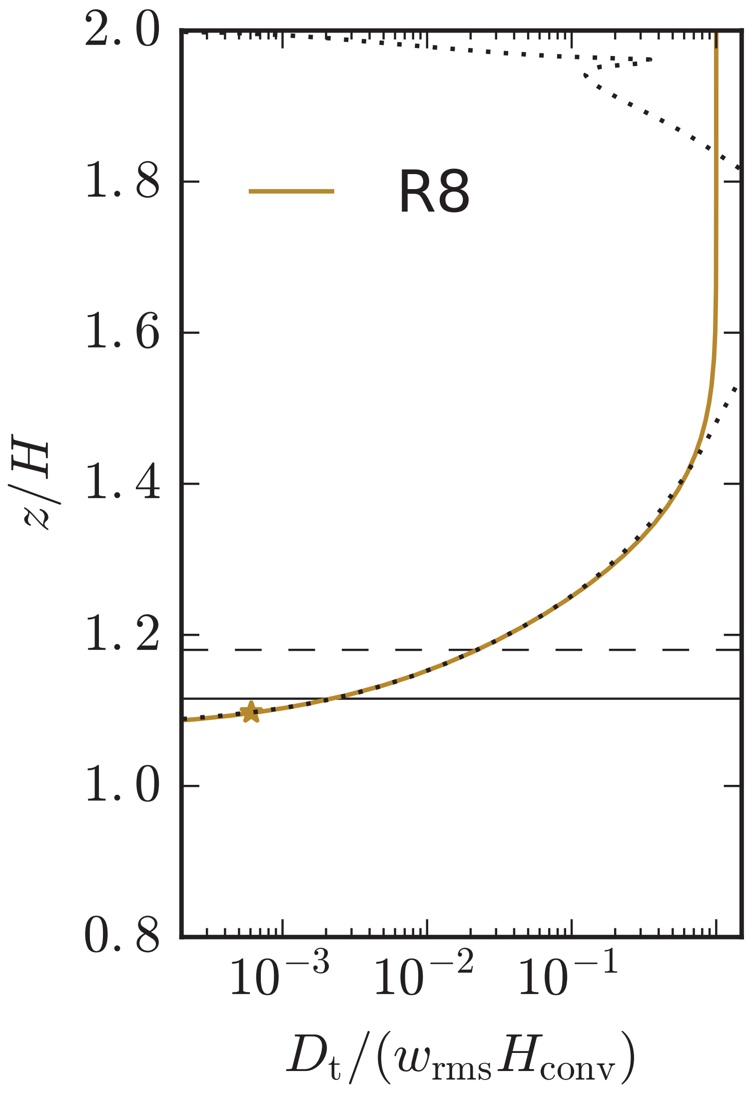
$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$\bar{c}(z,t) - \langle\bar{c}\rangle_z \to c_{\mathrm{s}}\,(z,t) = A(t)C(z)$$

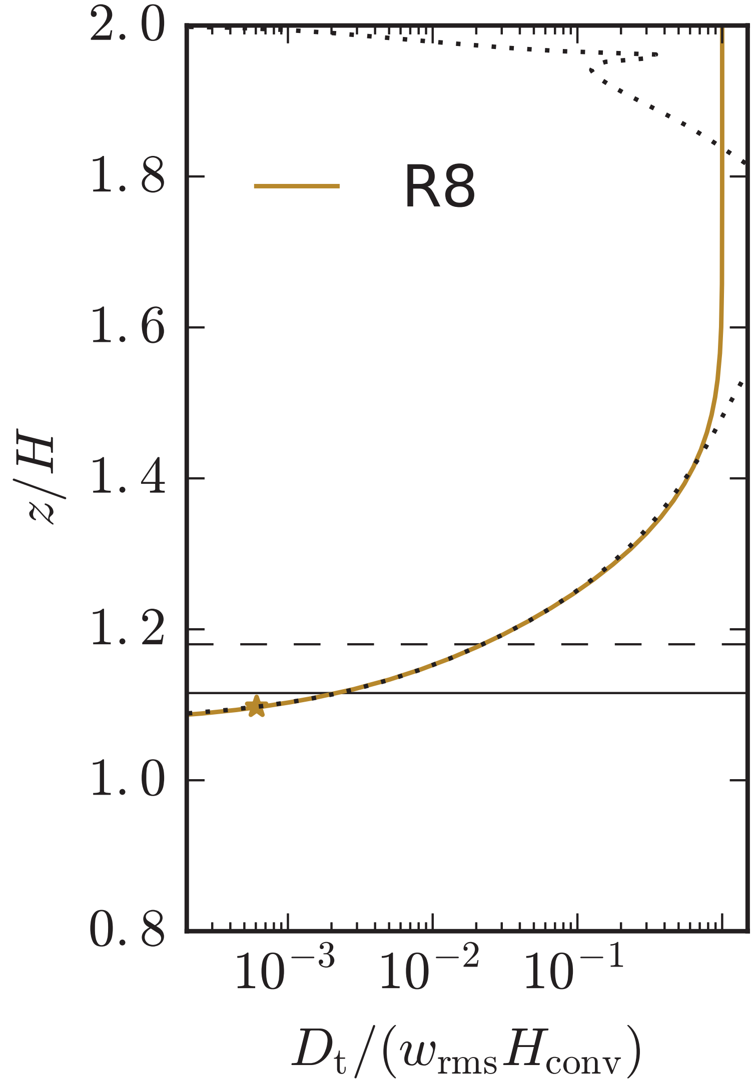
$$-\lambda C - D\partial_z^2 C = -\left\langle \boldsymbol{u}\cdot\boldsymbol{\nabla}\frac{c}{A}\right\rangle_{x,y,t}$$

$$-\lambda C = \partial_z\left[(D + D_{\mathrm{t}})\partial_z C\right]$$

Turbulent Diffusivity



Turbulent Diffusivity

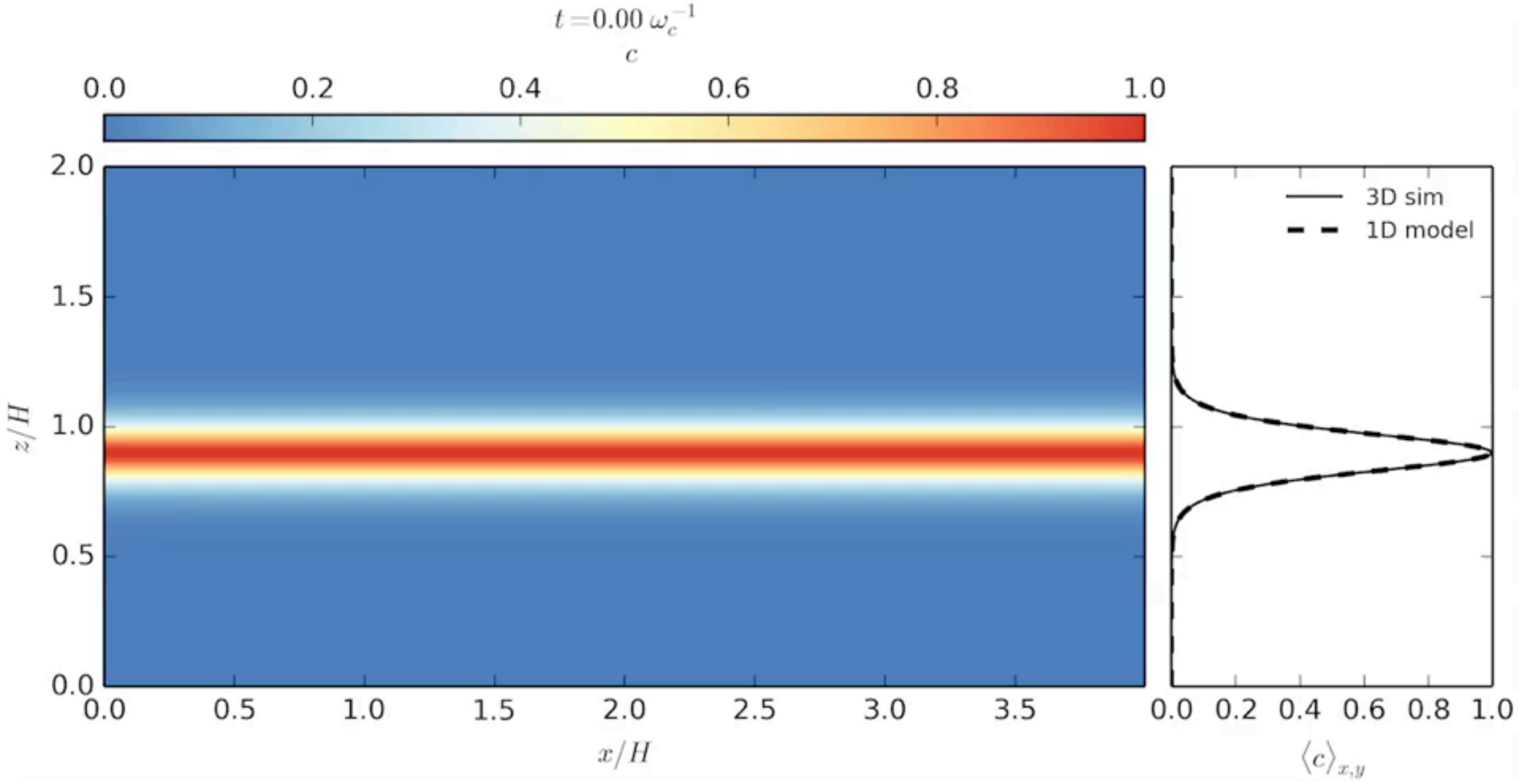


Solve:

$$\partial_t C = \partial_z[(D + D_{\rm t})\partial_z C]$$

fit

Does it work?



$t = 0.00\ \omega_c^{-1}$
c
0.0
0.2
0.4
0.6
0.8
1.0
2.0
1.5
1.0
0.5
0.0
z/H
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
x/H
3D sim
1D model
0.0 0.2 0.4 0.6 0.8 1.0
$\langle c \rangle_{x,y}$

Turbulent Diffusivity

Solve:

$$\partial_t C = \partial_z[(D + D_\mathrm{t})\partial_z C]$$

Turbulent diffusivity is a good approximation!

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u} \cdot \boldsymbol{\nabla} c$$

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u} \cdot \boldsymbol{\nabla} c$$

$$c(x, y, z, t) = \overline{c}(z, t) + \tilde{c}(x, y, z, t)$$

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$c(x,y,z,t) = \overline{c}(z,t) + \tilde{c}(x,y,z,t)$$

$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla}\cdot(\boldsymbol{u}\tilde{c})\right\rangle = -\partial_z \left\langle w\tilde{c}\right\rangle$$

$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z \overline{c} - \partial_z \left(w\tilde{c} - \left\langle w\tilde{c}\right\rangle\right)$$

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

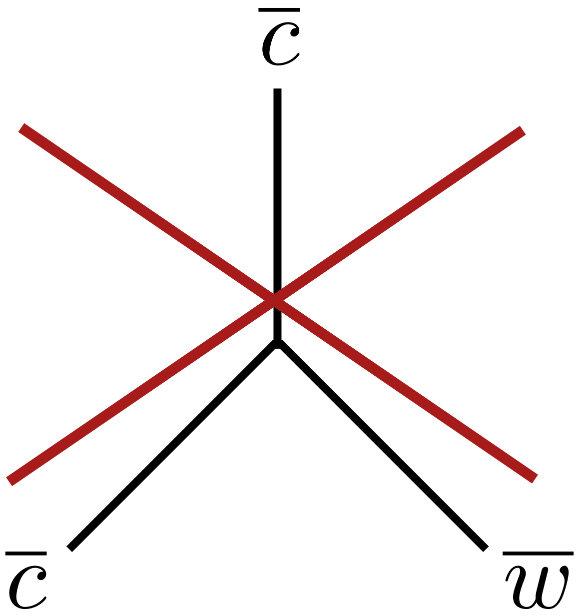
$$c(x,y,z,t) = \overline{c}(z,t) + \tilde{c}(x,y,z,t)$$

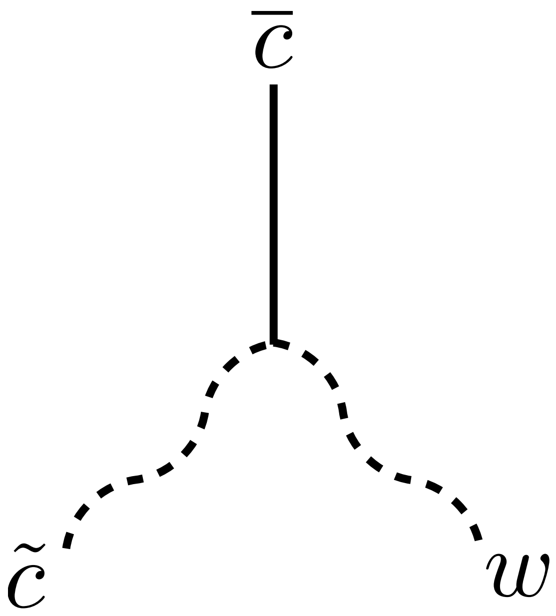
$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla}\cdot(\boldsymbol{u}\tilde{c})\right\rangle = -\partial_z \left\langle w\tilde{c}\right\rangle$$

$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z \overline{c} - \partial_z \left(w\tilde{c} - \left\langle w\tilde{c}\right\rangle\right)$$

“pain in the neck”

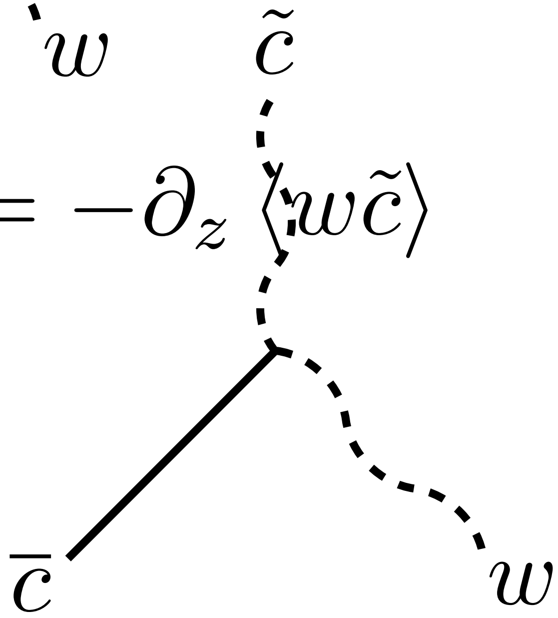
Why Turbulent Diffusivity?

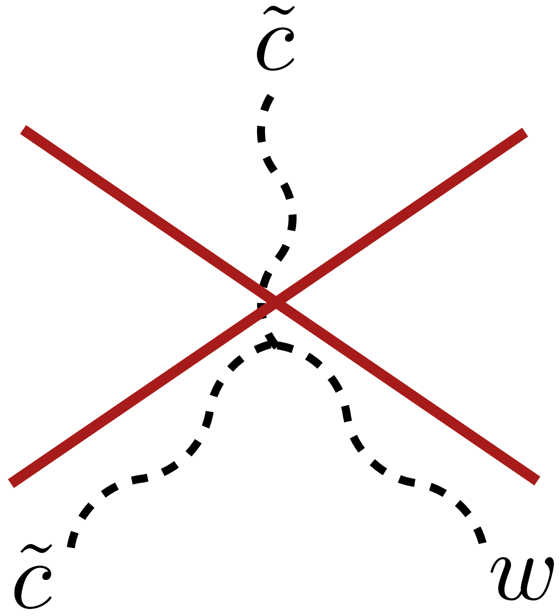




$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z \overline{c} - \partial_z \left(w\tilde{c} - \langle w\tilde{c} \rangle \right)$$

$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla} \cdot (\boldsymbol{u}\tilde{c}) \right\rangle = -\partial_z \left\langle w\tilde{c} \right\rangle$$





Quasilinear Approximation

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$c(x,y,z,t) = \overline{c}(z,t) + \tilde{c}(x,y,z,t)$$

$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla}\cdot(\boldsymbol{u}\tilde{c})\right\rangle = -\partial_z \left\langle w\tilde{c}\right\rangle$$

$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z \overline{c} - \xcancel{\partial_z\left(w\tilde{c} - \left\langle w\tilde{c}\right\rangle\right)}$$

"pain in the neck"

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$c(x,y,z,t) = \overline{c}(z,t) + \tilde{c}(x,y,z,t)$$

$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla}\cdot(\boldsymbol{u}\tilde{c})\right\rangle = -\partial_z \left\langle w\tilde{c}\right\rangle$$

$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z \overline{c} - \xcancel{\partial_z \left(w\tilde{c} - \left\langle w\tilde{c}\right\rangle\right)}$$

$$\tilde{c} = -\partial_z \overline{c}\ \mathcal{R}(i\omega; -D\nabla^2) w$$

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$c(x,y,z,t) = \overline{c}(z,t) + \tilde{c}(x,y,z,t)$$

$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla}\cdot(\boldsymbol{u}\tilde{c})\right\rangle = -\partial_z \left\langle w\tilde{c}\right\rangle$$

$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z\overline{c} - \xcancel{\partial_z\left(w\tilde{c} - \left\langle w\tilde{c}\right\rangle\right)}$$

$$\tilde{c} = -\partial_z\overline{c}\;\mathcal{R}(i\omega; -D\nabla^2)w$$

$$-\partial_z\left\langle w\tilde{c}\right\rangle = \partial_z\left[\left\langle w\mathcal{R}(i\omega; -D\nabla^2)w\right\rangle \partial_z\overline{c}\right]$$

Why Turbulent Diffusivity?

$$\partial_t c - D\nabla^2 c = -\boldsymbol{u}\cdot\boldsymbol{\nabla} c$$

$$c(x,y,z,t) = \overline{c}(z,t) + \tilde{c}(x,y,z,t)$$

$$\partial_t \overline{c} - D\partial_z^2 \overline{c} = -\left\langle \boldsymbol{\nabla}\cdot(\boldsymbol{u}\tilde{c})\right\rangle = -\partial_z \left\langle w\tilde{c}\right\rangle$$

$$\partial_t \tilde{c} - D\nabla^2 \tilde{c} = -w\partial_z \overline{c} - \xcancel{\partial_z \left(w\tilde{c} - \left\langle w\tilde{c}\right\rangle\right)}$$

$$\tilde{c} = -\partial_z \overline{c}\;\mathcal{R}(i\omega; -D\nabla^2)w$$

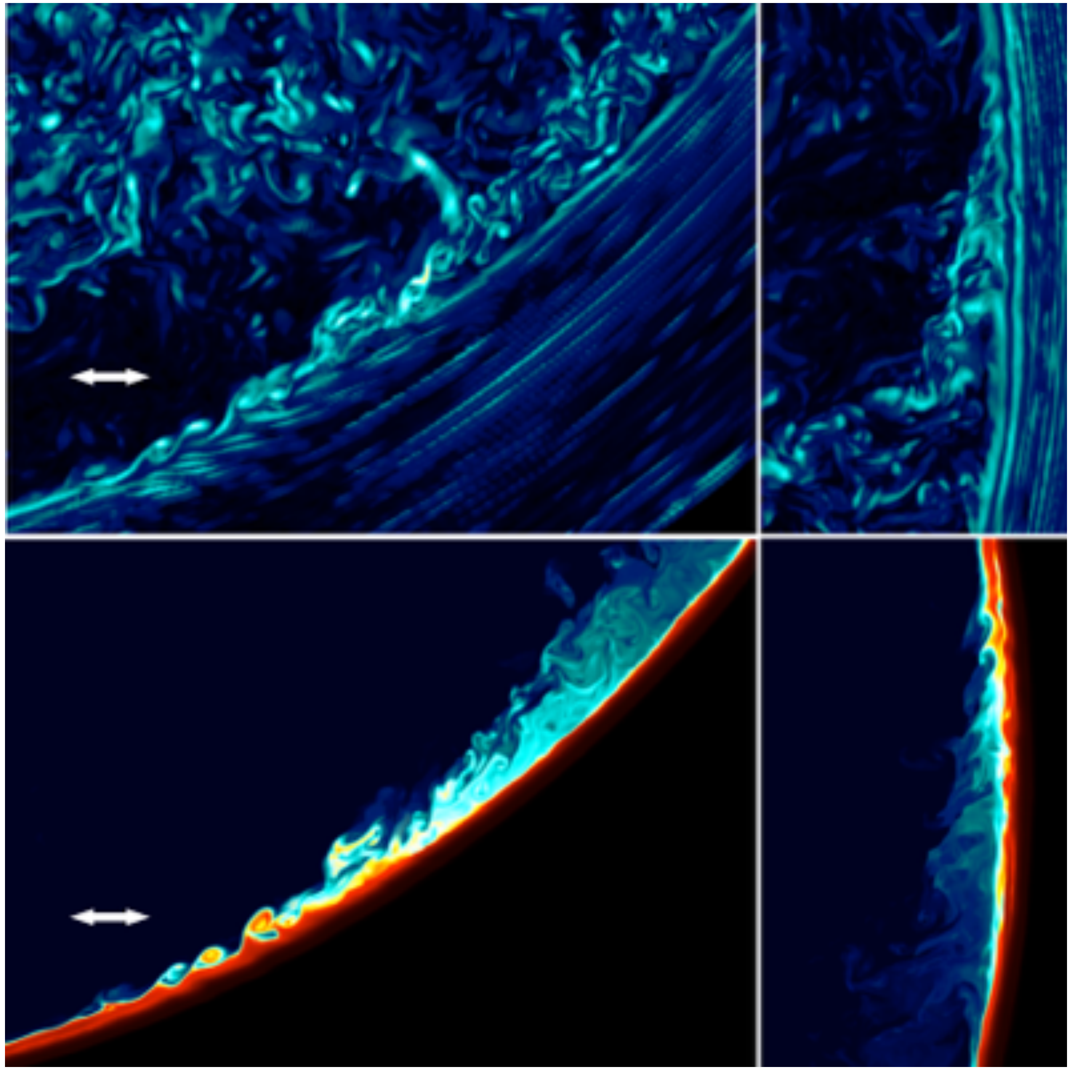
$$-\partial_z \left\langle w\tilde{c}\right\rangle = \partial_z\left[\left\langle w\mathcal{R}(i\omega;-D\nabla^2)w\right\rangle \partial_z\overline{c}\right]$$

D_{t}

Summary

1. Penetrative convection mixes fluid like a turbulent diffusivity (as least in these sims)
2. Very little mixing in regions which are very stable (D_t ~ gaussian)
3. Passive scalar field good for diagnosing mixing

Physical Model of Overshoot



Kelvin-Helmholtz instability?

$$\mathrm{Ri} = \frac{N^2}{(dU/dz)^2}$$

$$\mathrm{Ri} \sim 10^3$$

Woodward et al 2014

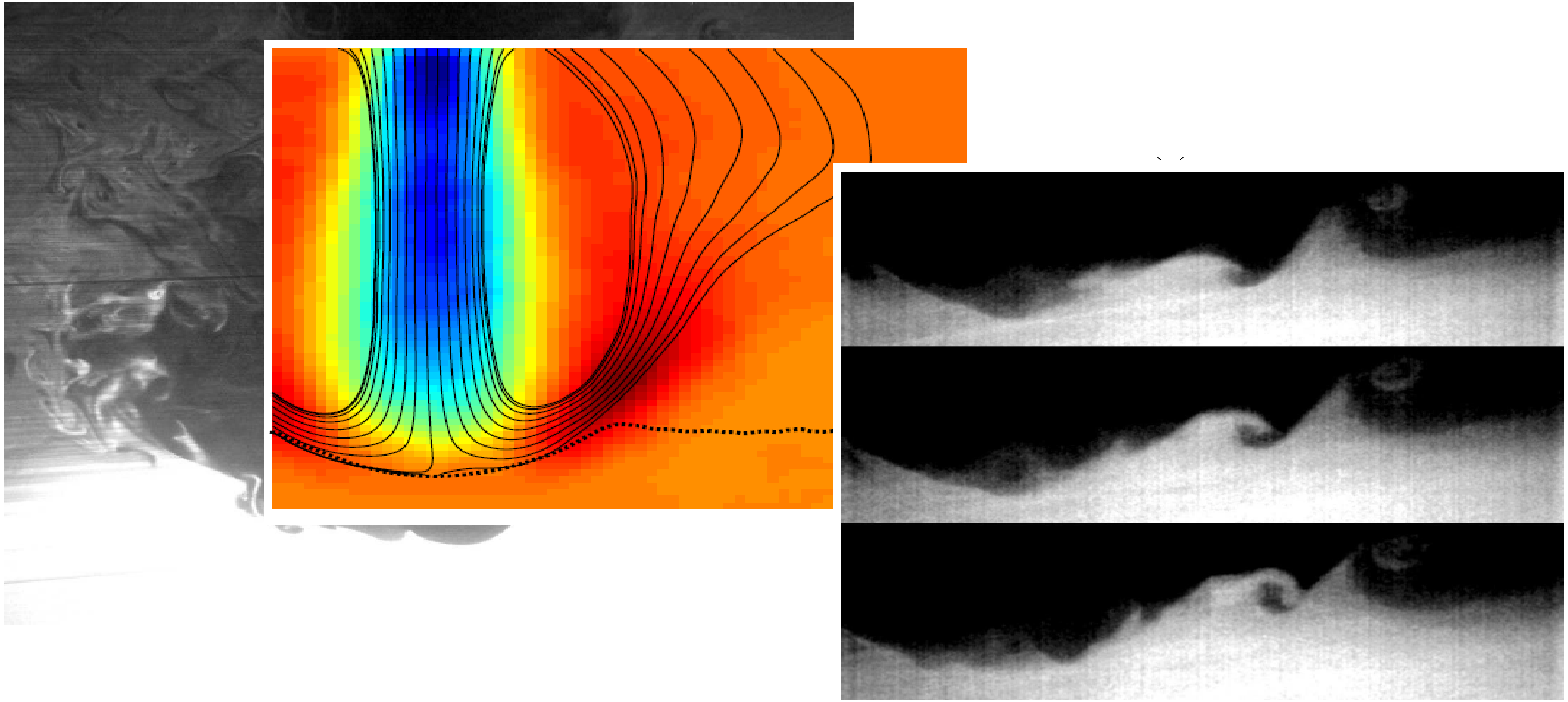
Physical Model of Overshoot

Miles instability

$$U(z_0) = c_p$$

resonance

Physical Model of Overshoot



Herault et al 2018