Los Alamos NATIONAL LABORATORY

3D Hydrodynamics for Stellar Evolution

Philipp Edelmann

PACE Meeting, Los Alamos, NM, USA

May 30, 2022

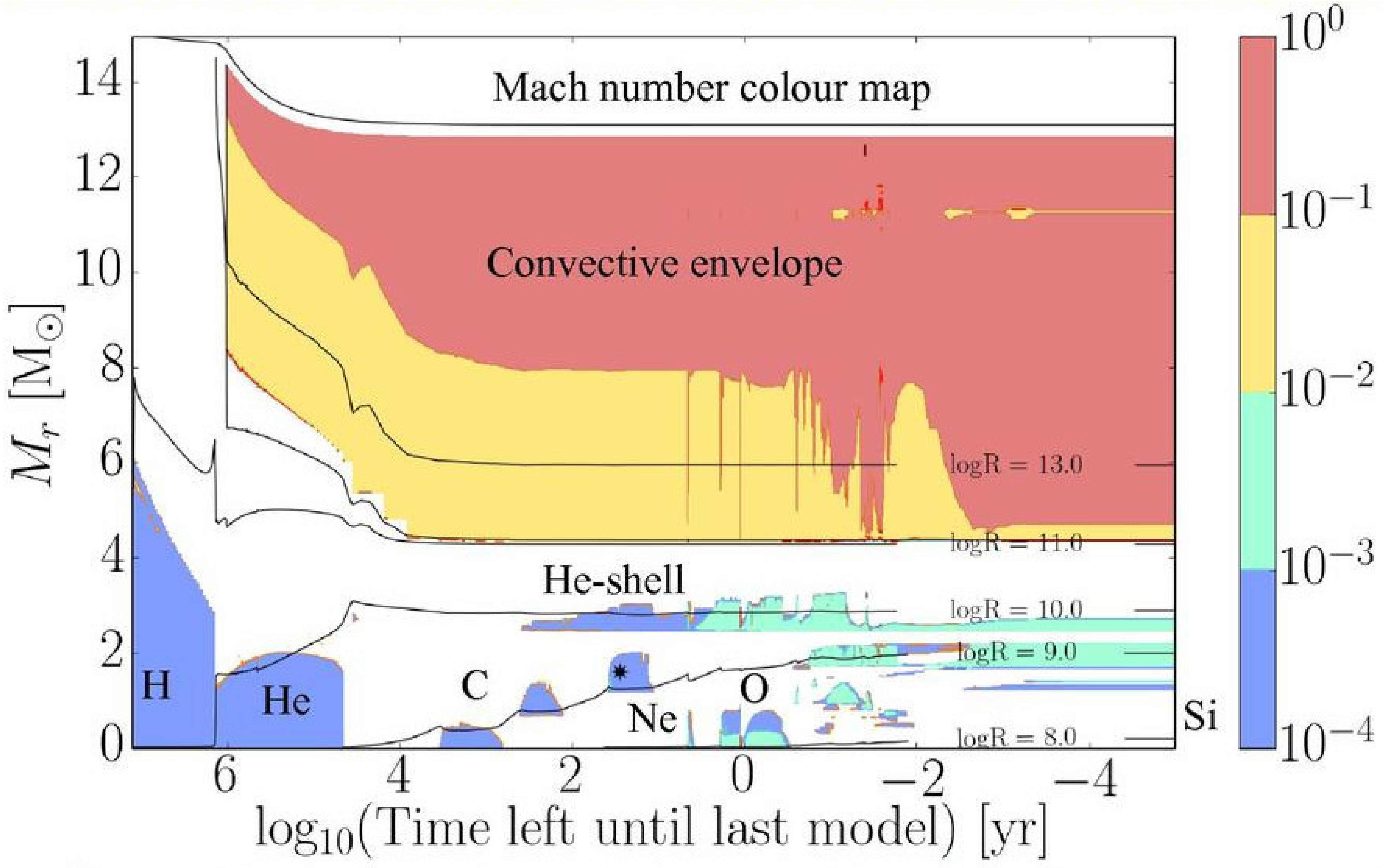
Hydrodynamics and stellar models

- one-dimensional, hydrostatic models still the default in stellar modeling
  time scales (sun as example):
  evolution at $\sim 10^9$ years, dynamics at ~ 30 min
- multidimensional fluid dynamics treated using “recipes”, e.g. mixing-length theory

Multidimensional hydrodynamics simulations cannot cover significant parts of stellar lifespan, but…

- they can cover the last evolutionary stages.
- they can validate the “recipes”.
- they can capture phenomena such as internal waves and make observational predictions.

Mach Numbers in Stellar Evolution



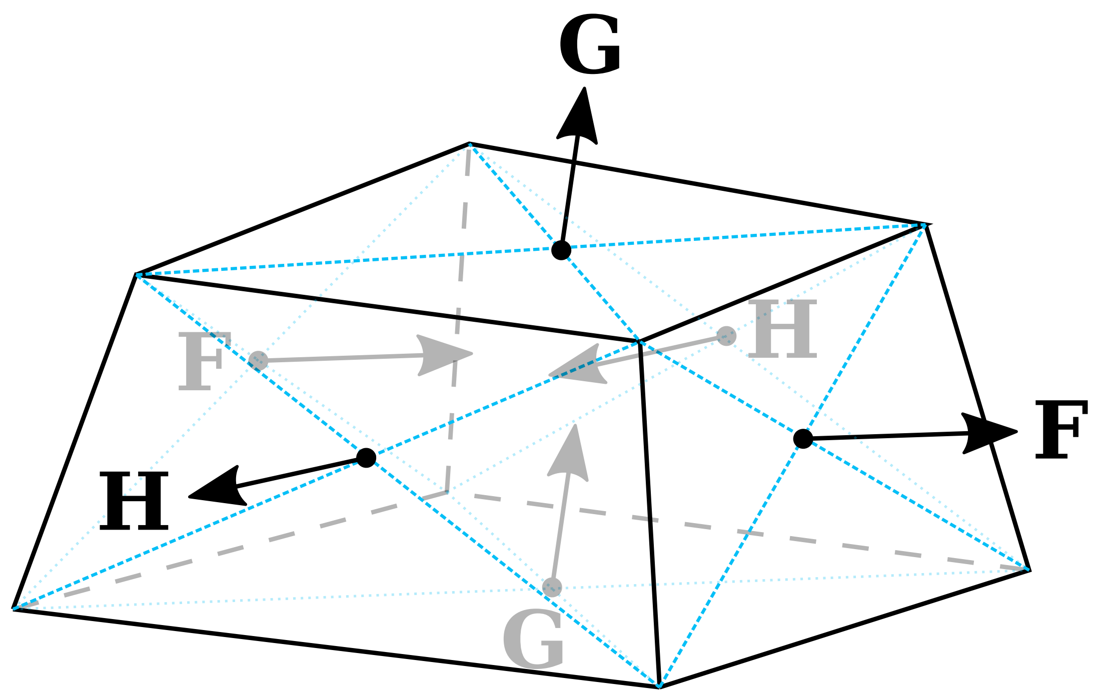
credit to Raphael Hirschi

Finite-Volume Schemes and Low Mach Numbers

- averages over control volumes
- fluxes between volumes given by Riemann solver or other numerical flux

Finite-Volume Schemes and Low Mach Numbers

- averages over control volumes
- fluxes between volumes given by Riemann solver or other numerical flux



Finite-Volume Schemes and Low Mach Numbers

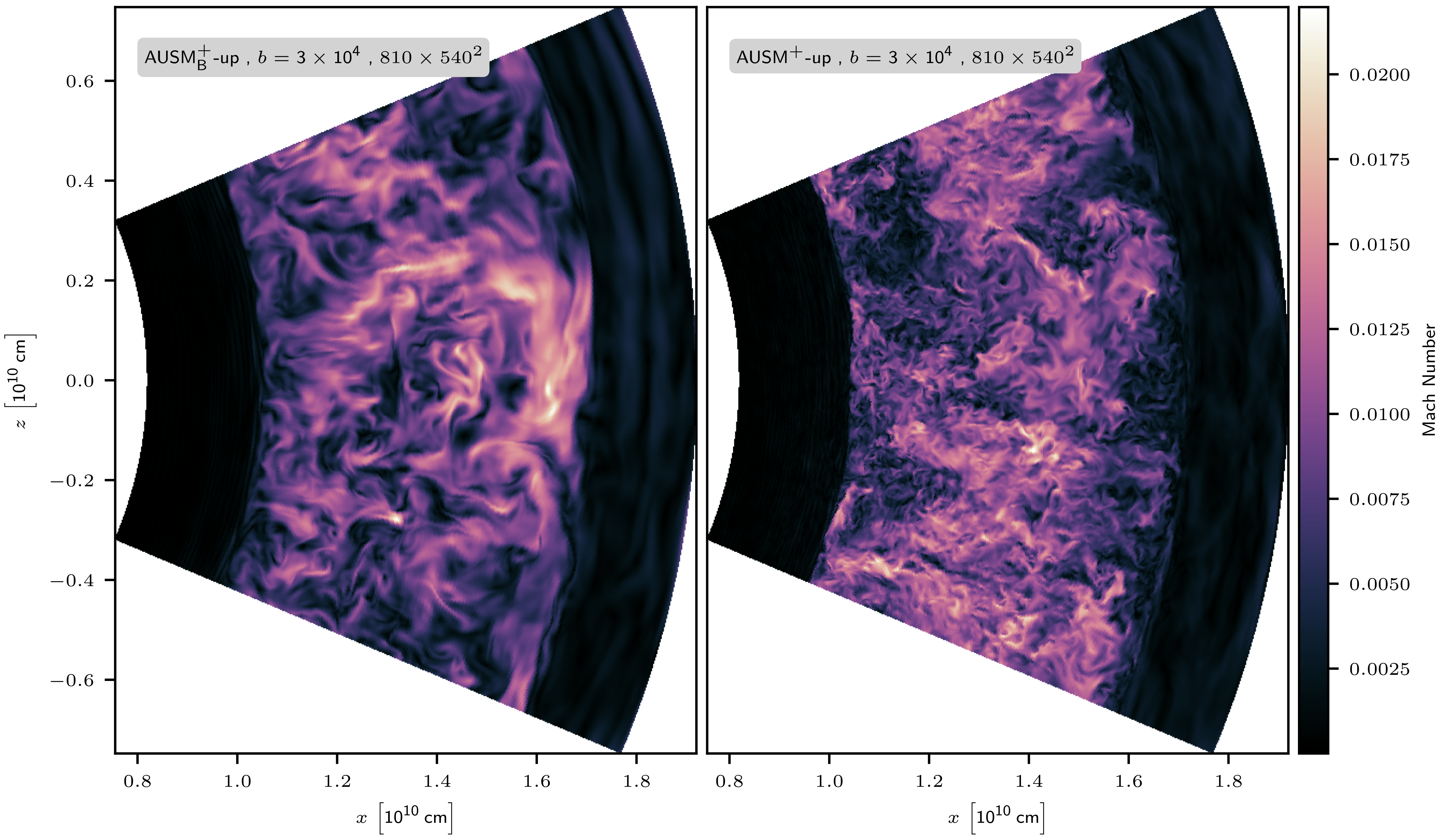
- averages over control volumes
- fluxes between volumes given by Riemann solver or other numerical flux
- most standard compressible methods show wrong scaling at low Mach numbers
  → high numerical diffusivity
- many fixes available: AUSM⁺-up (Liou, 2006), changed reconstruction (Thornber+, 2008), preconditioned Roe (Miczek+, 2015), …

flux Jacobian of the Euler equations

$$\begin{pmatrix} \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 & 0 \\ 0 & \mathcal{O}(1) & 0 & 0 & \mathcal{O}(\frac{1}{M_r^2}) & 0 \\ 0 & 0 & \mathcal{O}(1) & 0 & \mathcal{O}(\frac{1}{M_r^2}) & 0 \\ 0 & 0 & 0 & \mathcal{O}(1) & \mathcal{O}(\frac{1}{M_r^2}) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{O}(1) \end{pmatrix}$$

flux Jacobian of Roe solver

Helium Shell Burning



conventional scheme

all Mach scheme

Horst+ (2021)

RA-ILES Framework

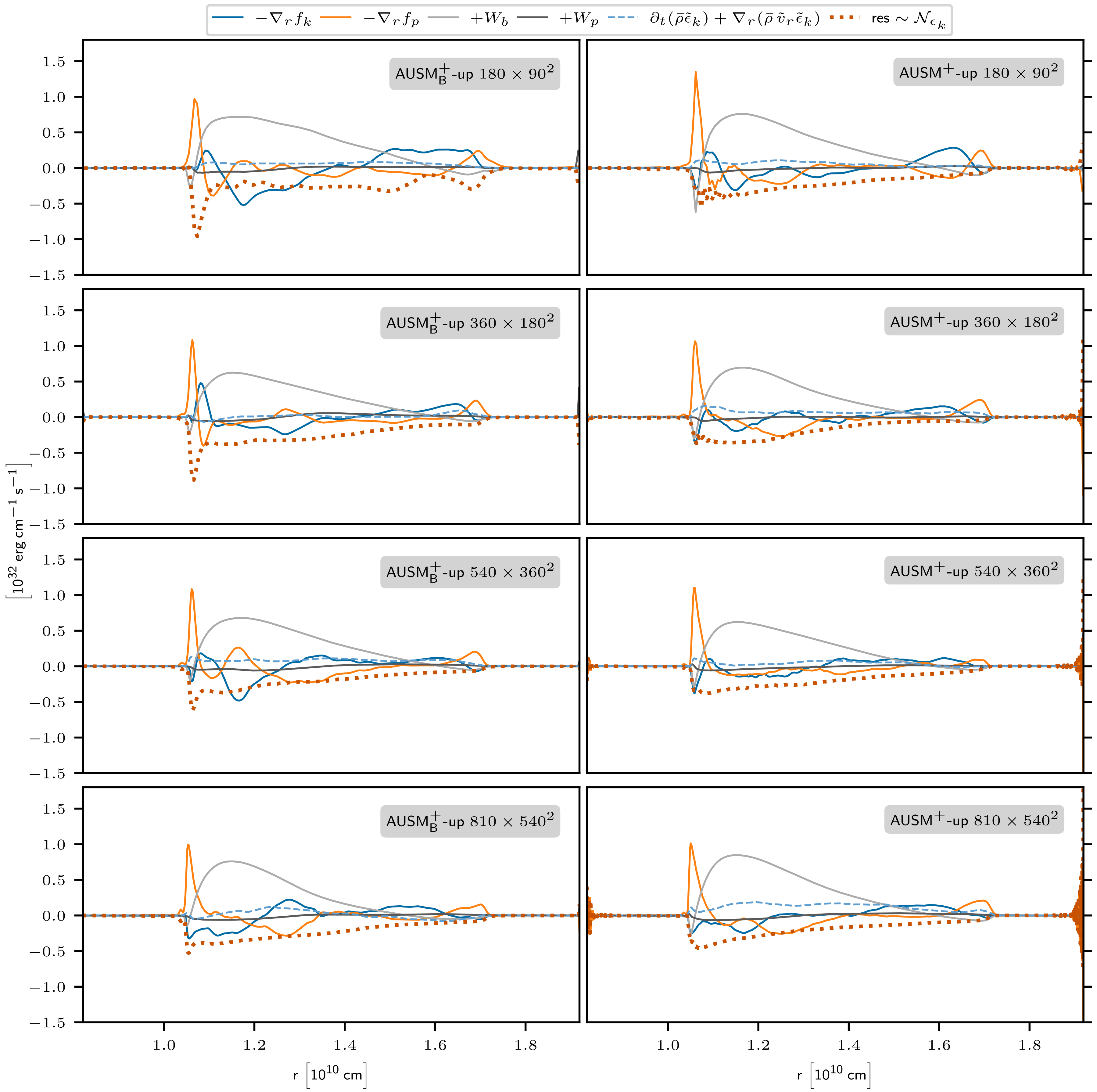
(Mocák+, 2014; Arnett+, 2019)

- Average:
  $\overline{q}(r) = \frac{1}{\Delta t \Delta\Omega} \int_{\Delta t} \int_{\Delta\Omega} q(r, \vartheta, \varphi, t)\, \mathrm{d}\Omega\, \mathrm{d}t$
- Favre average: $\tilde{q}(r) = \frac{\overline{\rho q}}{\overline{\rho}}$
- corresponding fluctuations: q', q''

Time evolution of kinetic energy

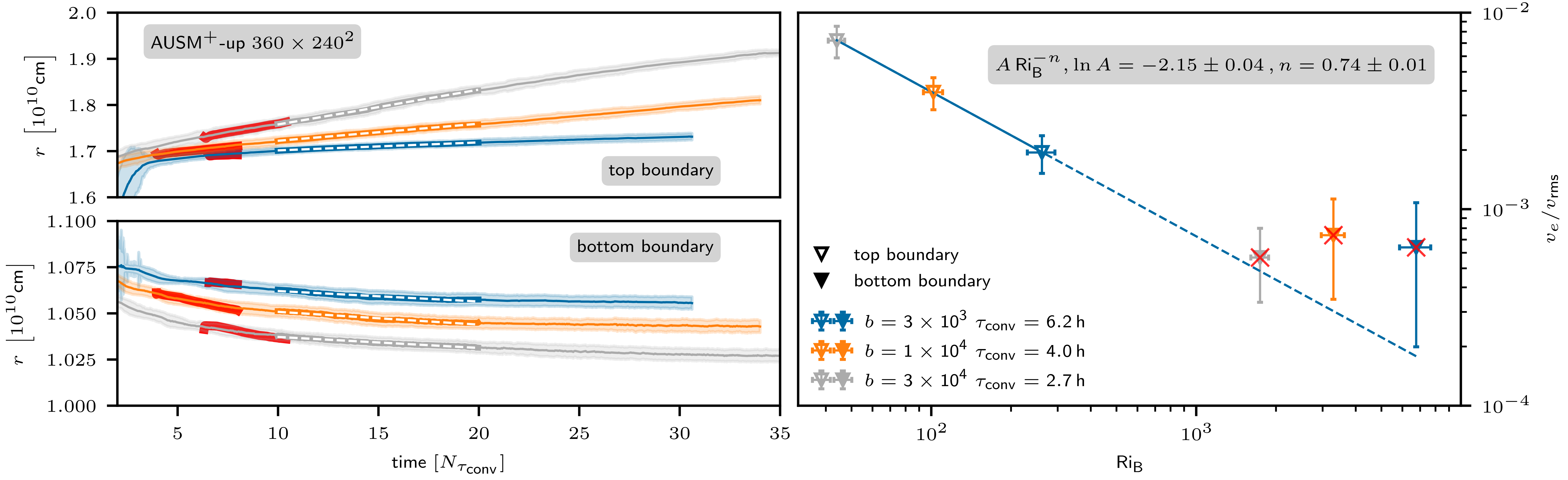
$$\partial_t(\bar{\rho}\,\tilde{\epsilon}_k) + \nabla_r(\bar{\rho}\,\tilde{v}_r\tilde{\epsilon}_k) = -\nabla_r\,(f_P + f_k) + W_b + W_P$$

- ϵ_k: specific kinetic energy
- $f_P = \overline{P' v_r'}$: acoustic flux
- $f_k = \overline{\rho v_r'' \epsilon_k''}$: turbulent kinetic energy flux
- $W_b = \overline{\rho}\, \overline{v_r''}\, \tilde{g}_r$: buoyancy work
- $W_P = \overline{P' d''}$: turbulent pressure dilatation
- $d = \nabla \cdot \vec{v}$: velocity divergence



Horst+ (2021)

Boundary tracking



Horst+ (2021)

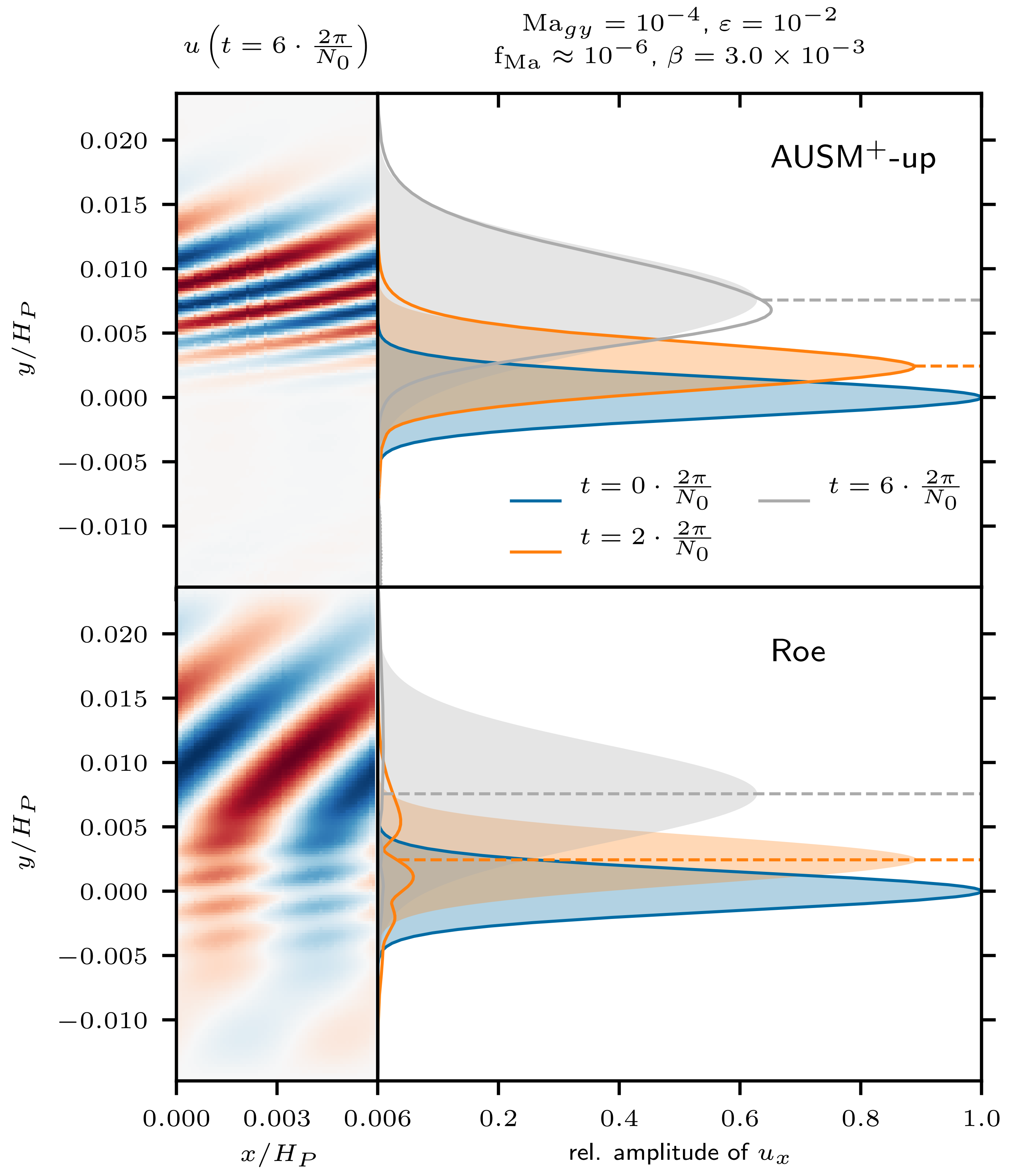
bulk Richardson number: $\text{Ri}_\text{B} = \frac{\Delta B l}{v_\text{rms}^2}$

buoyancy jump: $\Delta B = \int_{r_c-\Delta r}^{r_c-\Delta r} N^2 dr$

Internal Gravity Waves

Test using a planar wave packet

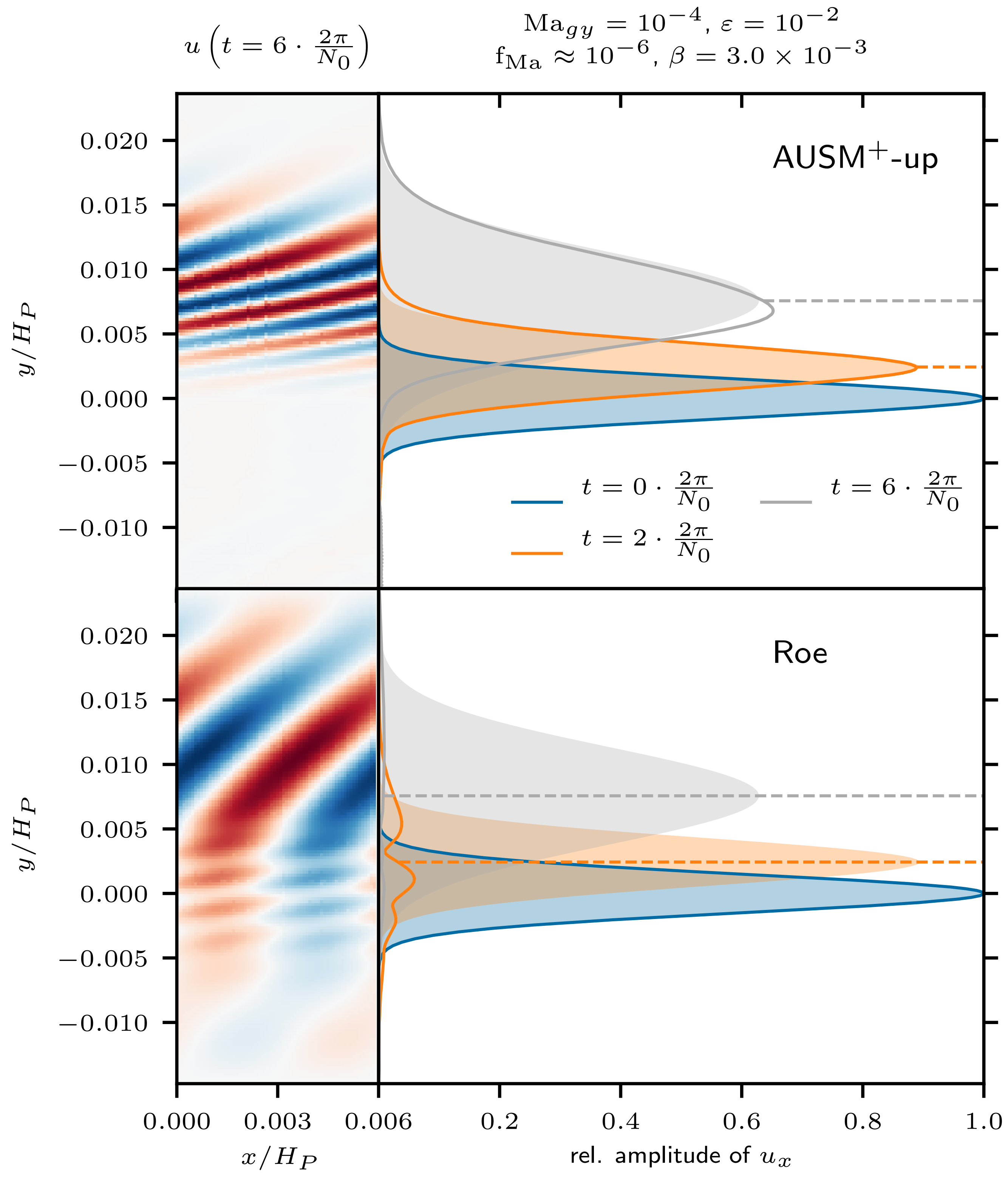
Horst+ (2020)

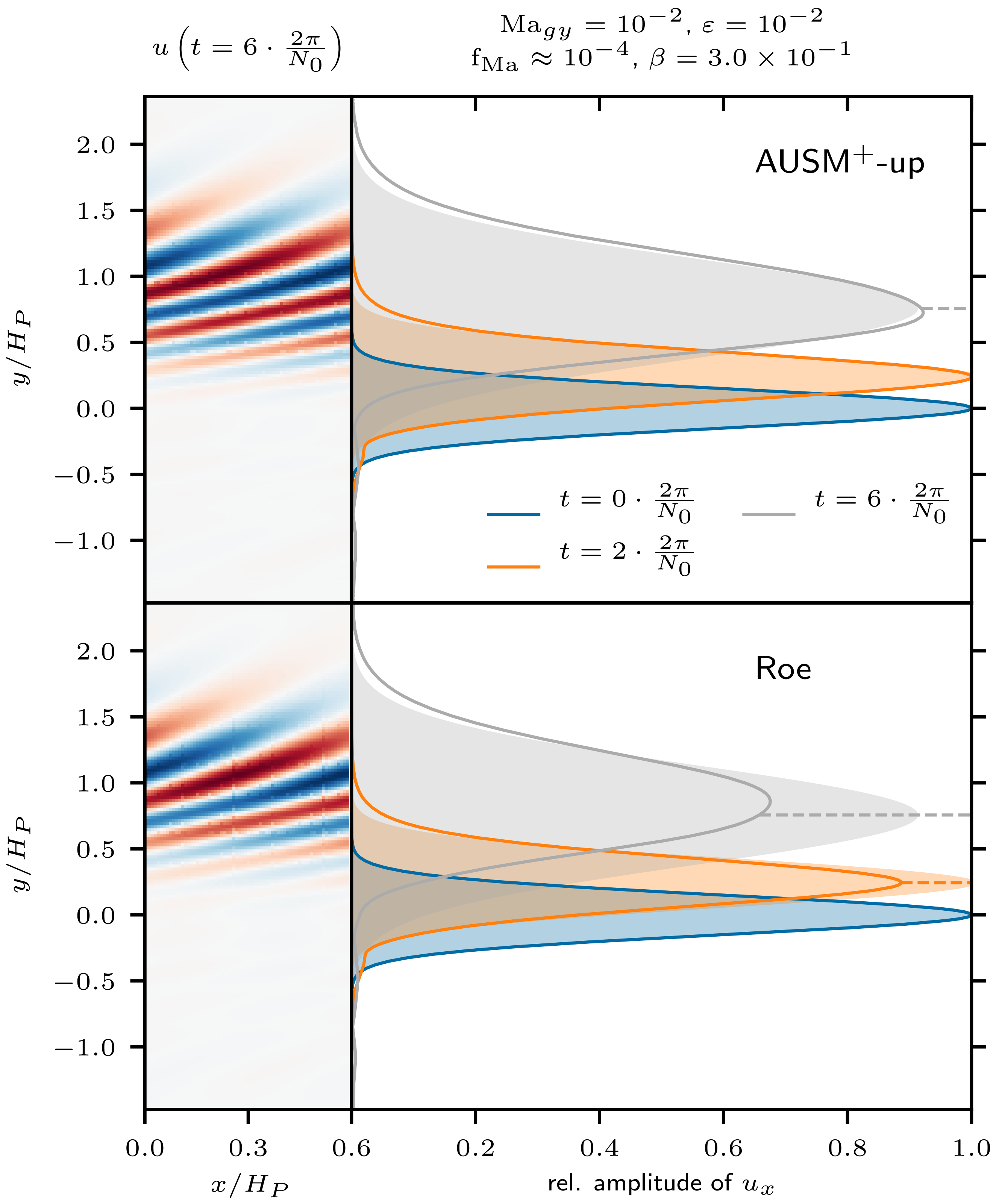


Internal Gravity Waves

Test using a planar wave packet

Horst+ (2020)





Euler Equations with Gravity

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{H}(\mathbf{U})}{\partial z} = \mathbf{S}(\mathbf{U})$$

$$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, E)^T \quad E = \rho\epsilon + \frac{1}{2}\rho|\mathbf{v}|^2 + \rho\phi$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ 0 \end{pmatrix}$$

The Concept of Well-Balancing

Euler Equations with Gravity

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{H}(\mathbf{U})}{\partial z} = \mathbf{S}(\mathbf{U})$$

$$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, E)^T \quad E = \rho \epsilon + \frac{1}{2}\rho\,|\,\mathbf{v}\,|^2 + \rho\phi$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ 0 \end{pmatrix}$$

example using single forward Euler step in 1D

The Concept of Well-Balancing

Euler Equations with Gravity

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{H}(\mathbf{U})}{\partial z} = \mathbf{S}(\mathbf{U})$$

$$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, E)^T \quad E = \rho\epsilon + \frac{1}{2}\rho|\mathbf{v}|^2 + \rho\phi$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ 0 \end{pmatrix}$$

example using single forward Euler step in 1D

$$\rho_i^1 = \rho_i^0 - \frac{\Delta t}{\Delta x}\left(\left(\hat{\mathbf{F}}^0_{i+\frac{1}{2}}\right)_1 - \left(\hat{\mathbf{F}}^0_{i-\frac{1}{2}}\right)_1\right)$$

$$(\rho u)_i^1 = (\rho u)_i^0 - \frac{\Delta t}{\Delta x}\left(\left(\hat{\mathbf{F}}^0_{i+\frac{1}{2}}\right)_2 - \left(\hat{\mathbf{F}}^0_{i-\frac{1}{2}}\right)_2\right) + \Delta t\left(\hat{\mathbf{S}}_i\right)_2$$

$$E_i^1 = E_i^0 - \frac{\Delta t}{\Delta x}\left(\left(\hat{\mathbf{F}}^0_{i+\frac{1}{2}}\right)_3 - \left(\hat{\mathbf{F}}^0_{i-\frac{1}{2}}\right)_3\right)$$

The Concept of Well-Balancing

Euler Equations with Gravity

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{H}(\mathbf{U})}{\partial z} = \mathbf{S}(\mathbf{U})$$

$$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, E)^T \quad E = \rho\epsilon + \frac{1}{2}\rho|\mathbf{v}|^2 + \rho\phi$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ 0 \end{pmatrix}$$

example using single forward Euler step in 1D

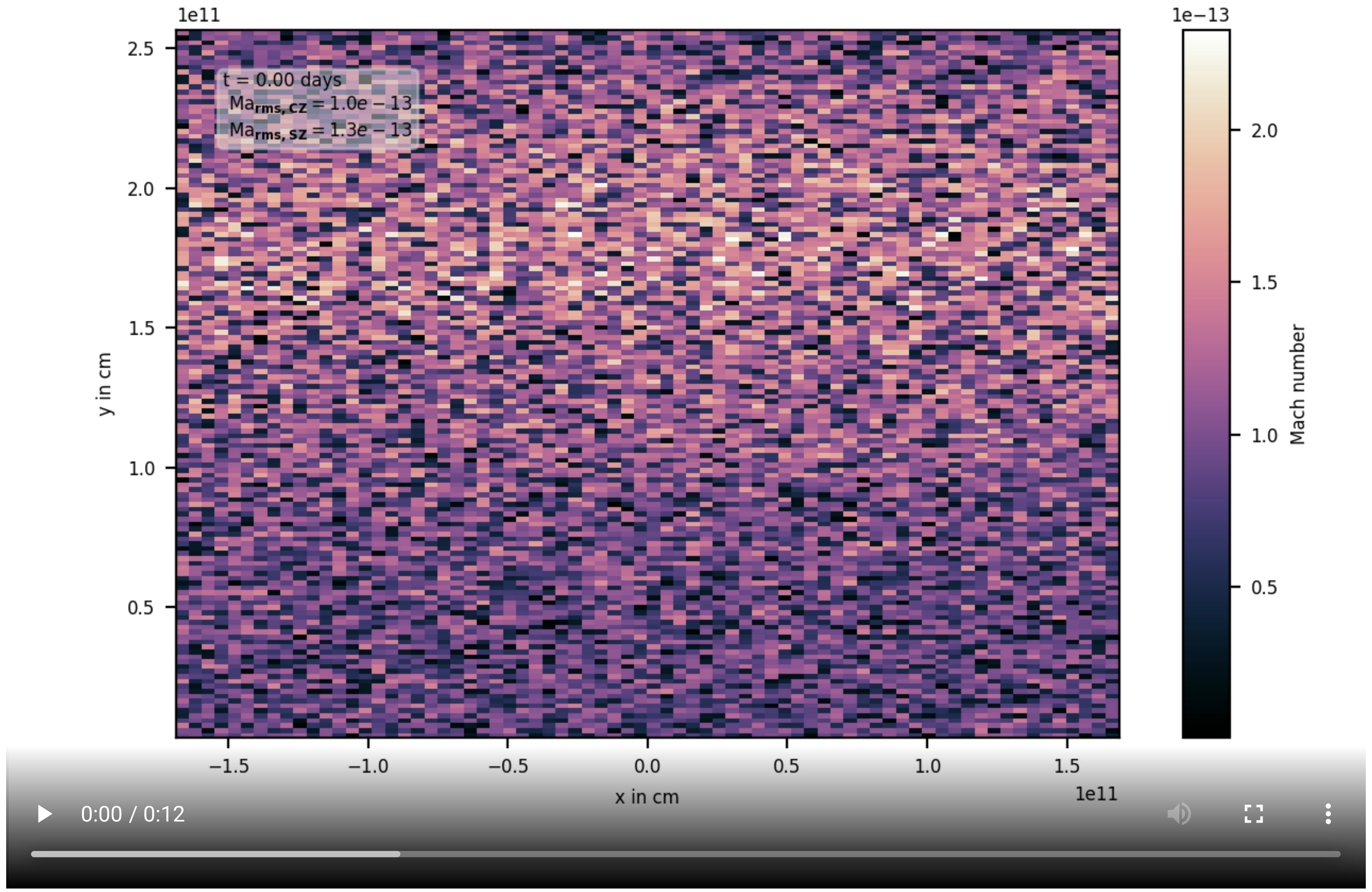
$$\rho_i^1 = \rho_i^0 - \frac{\Delta t}{\Delta x}\left(\left(\hat{\mathbf{F}}^0_{i+\frac{1}{2}}\right)_1 - \left(\hat{\mathbf{F}}^0_{i-\frac{1}{2}}\right)_1\right)$$

$$(\rho u)_i^1 = (\rho u)_i^0 - \frac{\Delta t}{\Delta x}\left(\left(\hat{\mathbf{F}}^0_{i+\frac{1}{2}}\right)_2 - \left(\hat{\mathbf{F}}^0_{i-\frac{1}{2}}\right)_2\right) + \Delta t\left(\hat{\mathbf{S}}_i\right)_2$$

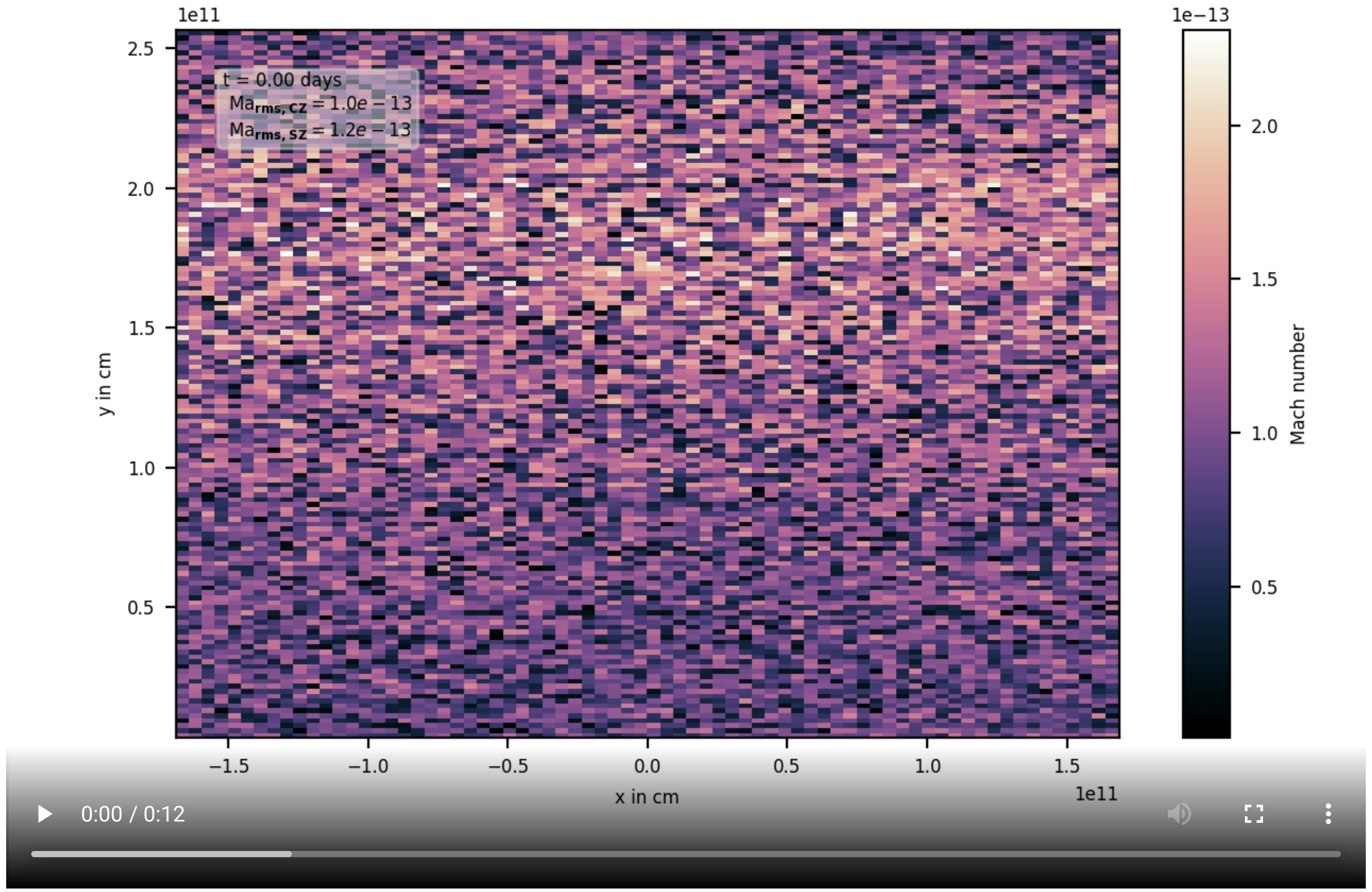
$$E_i^1 = E_i^0 - \frac{\Delta t}{\Delta x}\left(\left(\hat{\mathbf{F}}^0_{i+\frac{1}{2}}\right)_3 - \left(\hat{\mathbf{F}}^0_{i-\frac{1}{2}}\right)_3\right)$$

hydrostatic solution

Comparison of Methods



no well-balancing



with well-balancing

courtesy of Leo Horst (formerly HITS)

Seven-League Hydro code

- solves the compressible Euler equations in 1-, 2-, 3-D
- explicit and implicit time integration
- flux preconditioning to ensure correct behavior at low Mach numbers
- other low Mach number schemes (e.g., AUSM$^+$-up)
- works for low and high Mach numbers on the same grid
- hybrid (MPI, OpenMP) parallelization (tested up to 458 752 cores)
- several solvers for the linear system:
  BiCGSTAB, GMRES, Multigrid, (direct)
- arbitrary curvilinear meshes
  using a rectangular computational mesh
- gravity solver (monopole, Multigrid)
- radiation in the diffusion limit
- magnetohydrodynamics (MHD)
- well-balanced gravity (Edelmann+, 2021)
- general equation of state
- general nuclear reaction network

Code-Comparison Project

Dynamics in a stellar convective layer and at its boundary: Comparison of five 3D hydrodynamics codes

R. Andrassy[1], J. Higl[1], H. Mao[2], M. Mocák[3], D. G. Vlaykov[4], W. D. Arnett[5], I. Baraffe[4,6], S. W. Campbell[7,8], T. Constantino[4], P. V. F. Edelmann[9], T. Goffrey[10], T. Guillet[4], F. Herwig[11,12], R. Hirschi[3,13], L. Horst[1], G. Leidi[1,14], C. Meakin[5,15], J. Pratt[16], F. Rizzuti[3], F. K. Röpke[1,17], and P. Woodward[2,12]

- idea conceived at Stellar Hydro Days V (2019, Exeter, UK)
- comparison of 5 hydrodynamics codes: FLASH, MUSIC, PPMSTAR, PROMPI, SLH
- convective boundary inspired by a star
- tracking the mixing of two fluids

256^3

FLASH

MUSIC

PPMSTAR

PROMPI

SLH

Code-Comparison Project

Dynamics in a stellar convective layer and at its boundary: Comparison of five 3D hydrodynamics codes

R. Andrassy[1], J. Higl[1], H. Mao[2], M. Mocák[3], D. G. Vlaykov[4], W. D. Arnett[5], I. Baraffe[4,6], S. W. Campbell[7,8], T. Constantino[4], P. V. F. Edelmann[9], T. Goffrey[10], T. Guillet[4], F. Herwig[11,12], R. Hirschi[3,13], L. Horst[1], G. Leidi[1,14], C. Meakin[5,15], J. Pratt[16], F. Rizzuti[3], F. K. Röpke[1,17], and P. Woodward[2,12]

- idea conceived at Stellar Hydro Days V (2019, Exeter, UK)
- comparison of 5 hydrodynamics codes: FLASH, MUSIC, PPMSTAR, PROMPI, SLH
- convective boundary inspired by a star
- tracking the mixing of two fluids

- mass entrainment within 4% of each other (256^3 resolution)
- time-averaged profiles within 3σ
- all simulation outputs available for analysis through a JupyterHub at www.ppmstar.org/coco

256^3

FLASH

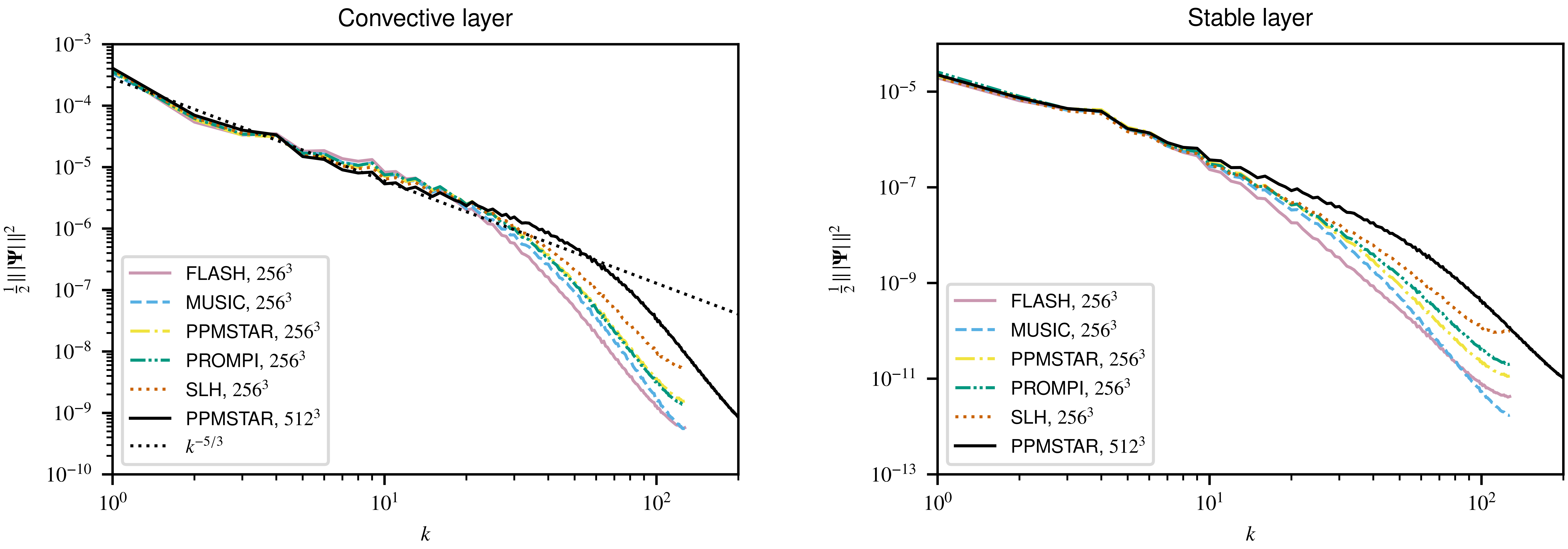
MUSIC

PPMSTAR

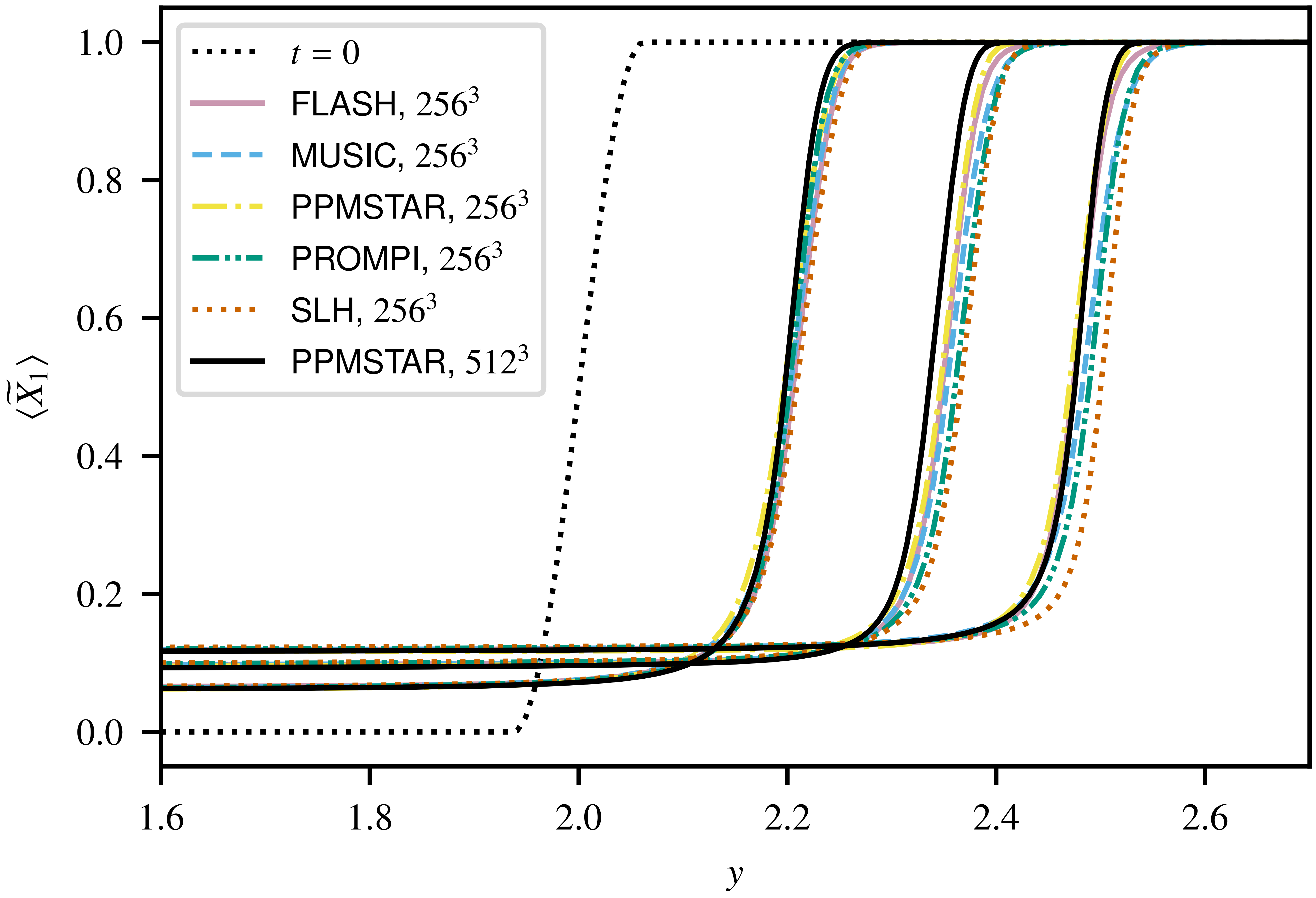
PROMPI

SLH

Convective Spectra



Mass Entrainment



Another Approach: Anelastic Spectral Simulations

Another Approach: Anelastic Spectral Simulations

- anelastic approximation

$$\nabla \cdot \bar{\rho}\mathbf{v} = 0,$$

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v}\cdot\nabla)\mathbf{v} - \nabla P - C\bar{g}\hat{\mathbf{r}} + 2(\mathbf{v}\times\hat{\mathbf{z}}\Omega) + \bar{\nu}\left(\nabla^2\mathbf{v} + \frac{1}{3}\nabla(\nabla\cdot\mathbf{v})\right),$$

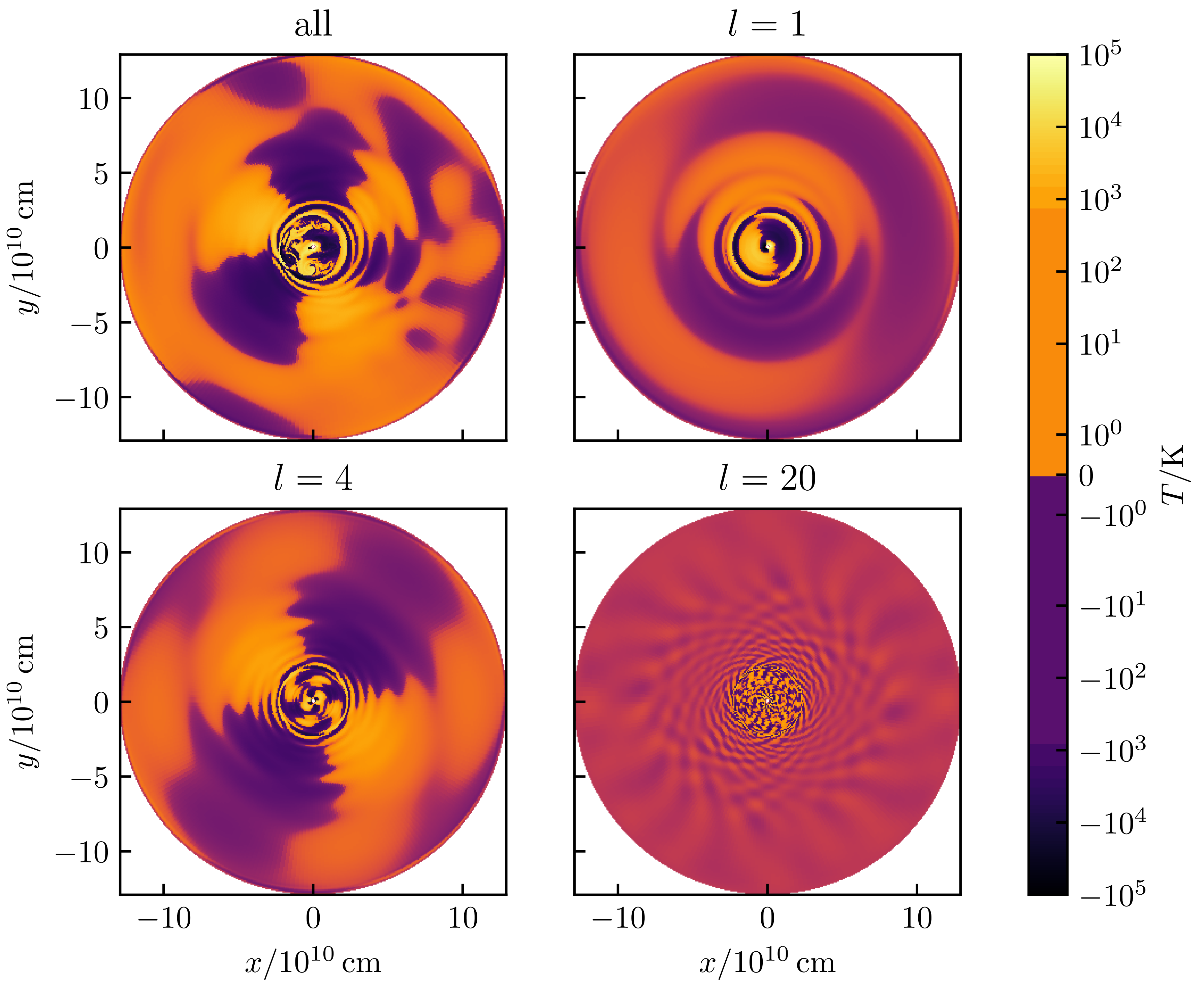
$$\frac{\partial T}{\partial t} = -(\mathbf{v}\cdot\nabla)T + (\gamma - 1)Th_\rho v_r - v_r\left(\frac{\partial \bar{T}}{\partial r} - (\gamma-1)\bar{T}h_\rho\right) + \frac{\bar{Q}}{c_v\bar{\rho}} + \frac{1}{c_v\bar{\rho}}\nabla\cdot(c_p\kappa\bar{\rho}\nabla T) + \frac{1}{c_v\bar{\rho}}\nabla\cdot(c_p\kappa_r\bar{\rho}\nabla\bar{T}).$$

no sound waves (or p modes) possible in simulation

Another Approach: Anelastic Spectral Simulations

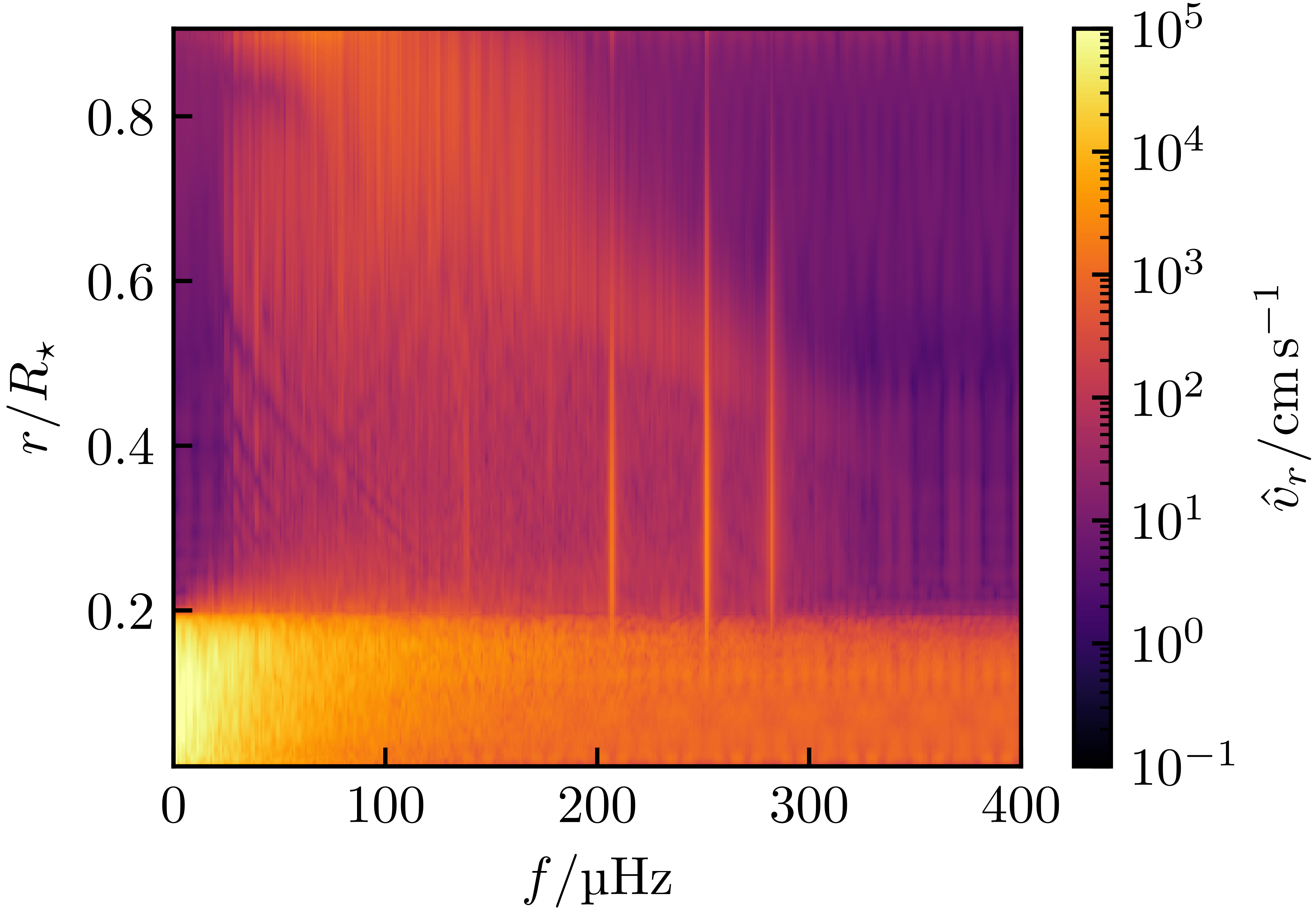
- anelastic approximation
- decomposition into spherical harmonics

can easily extract different (l, m) components of the variables

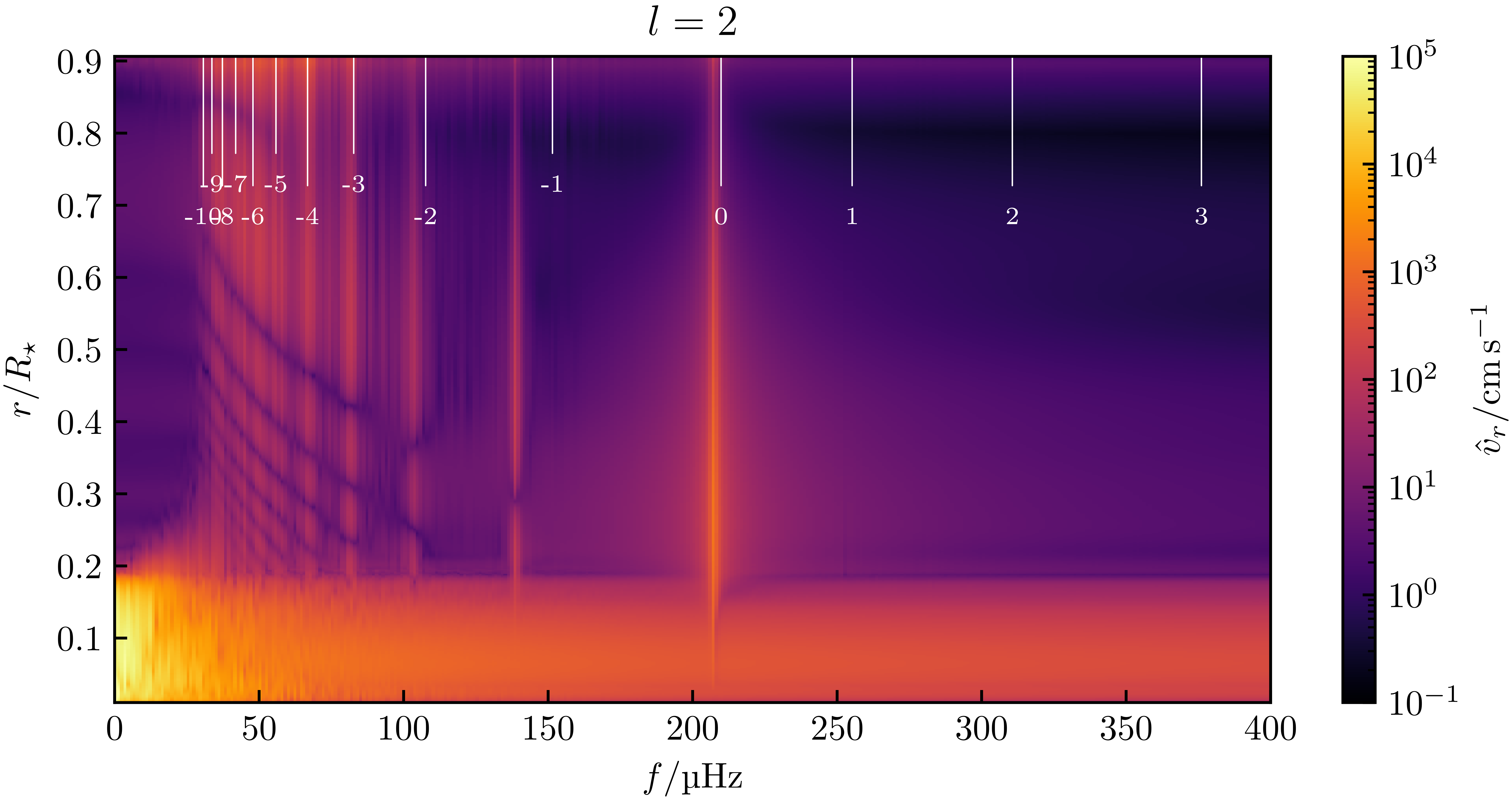


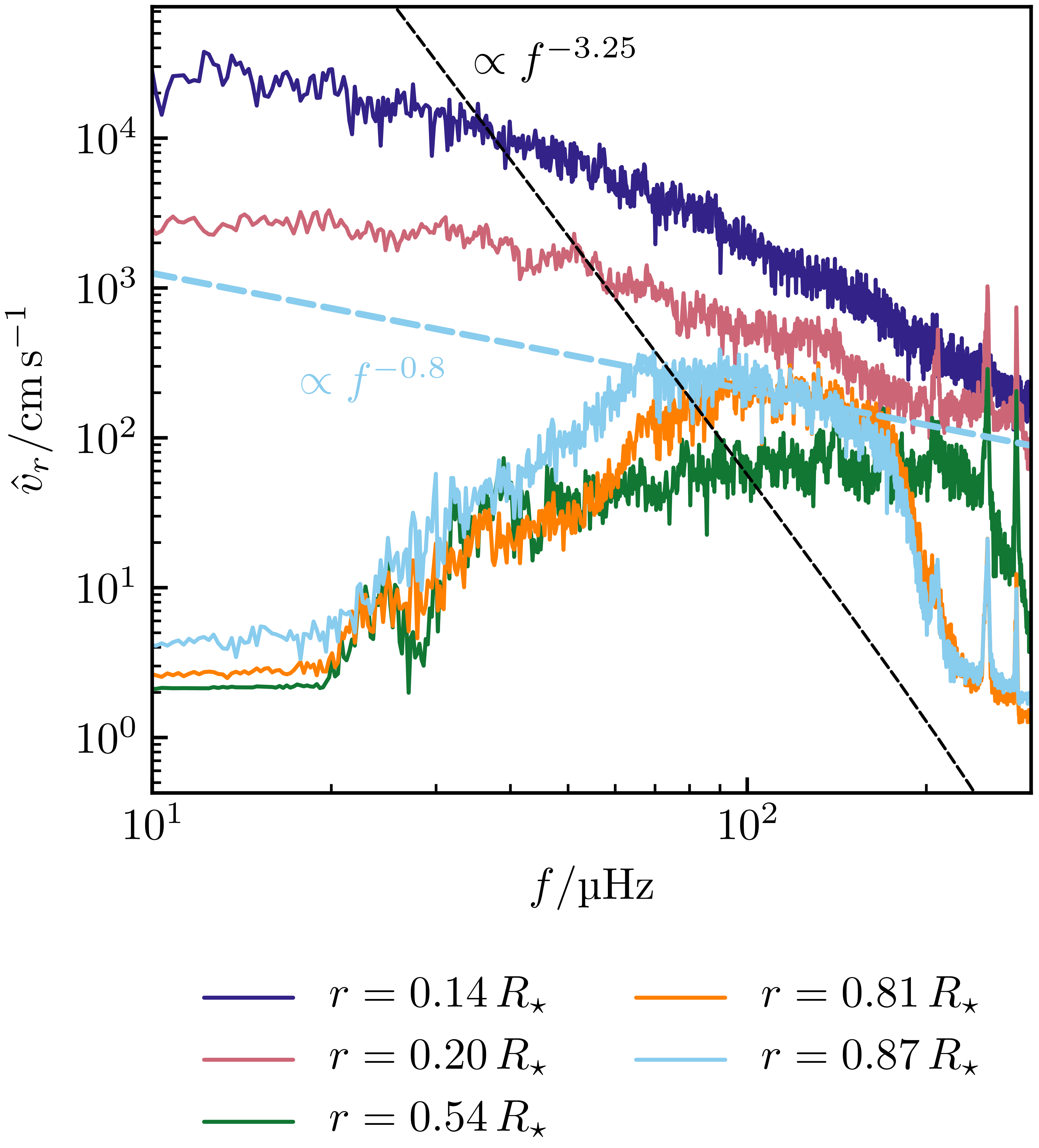
Core convection in a 3 solar mass star

Wave signatures in envelope



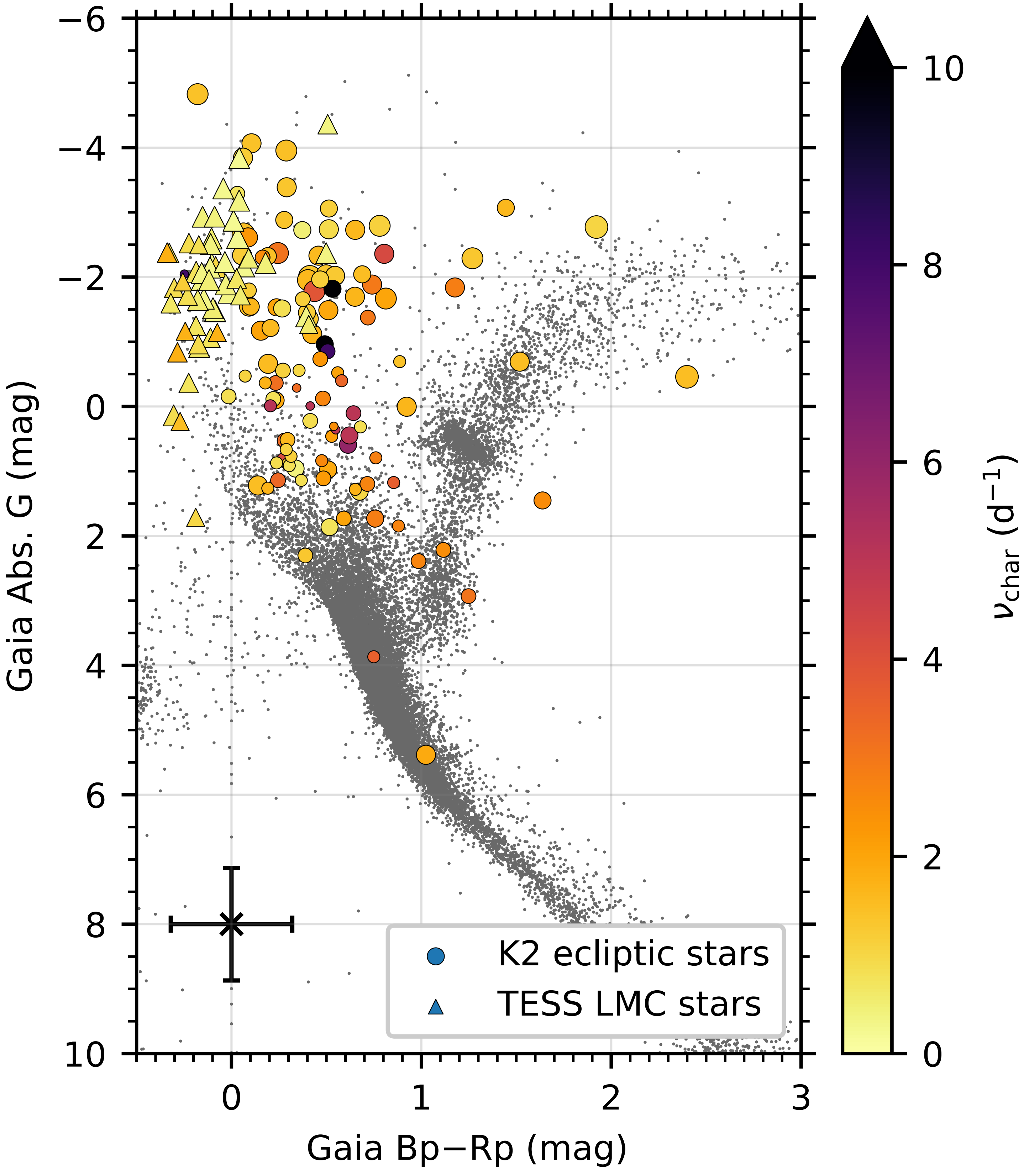
Edelmann+ (2021)


$l = 2$
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
$r/R_\star$
0
50
100
150
200
250
300
350
400
$f/\mu\mathrm{Hz}$
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
10^5
10^4
10^3
10^2
10^1
10^0
10^{-1}
$\hat{v}_r/\mathrm{cm\,s^{-1}}$



$\propto f^{-3.25}$
$\propto f^{-0.8}$
10^4
10^3
10^2
10^1
10^0
$\hat{v}_r/\mathrm{cm\,s^{-1}}$
10^1
10^2
$f/\mu\mathrm{Hz}$
$r = 0.14\,R_\star$
$r = 0.20\,R_\star$
$r = 0.54\,R_\star$
$r = 0.81\,R_\star$
$r = 0.87\,R_\star$

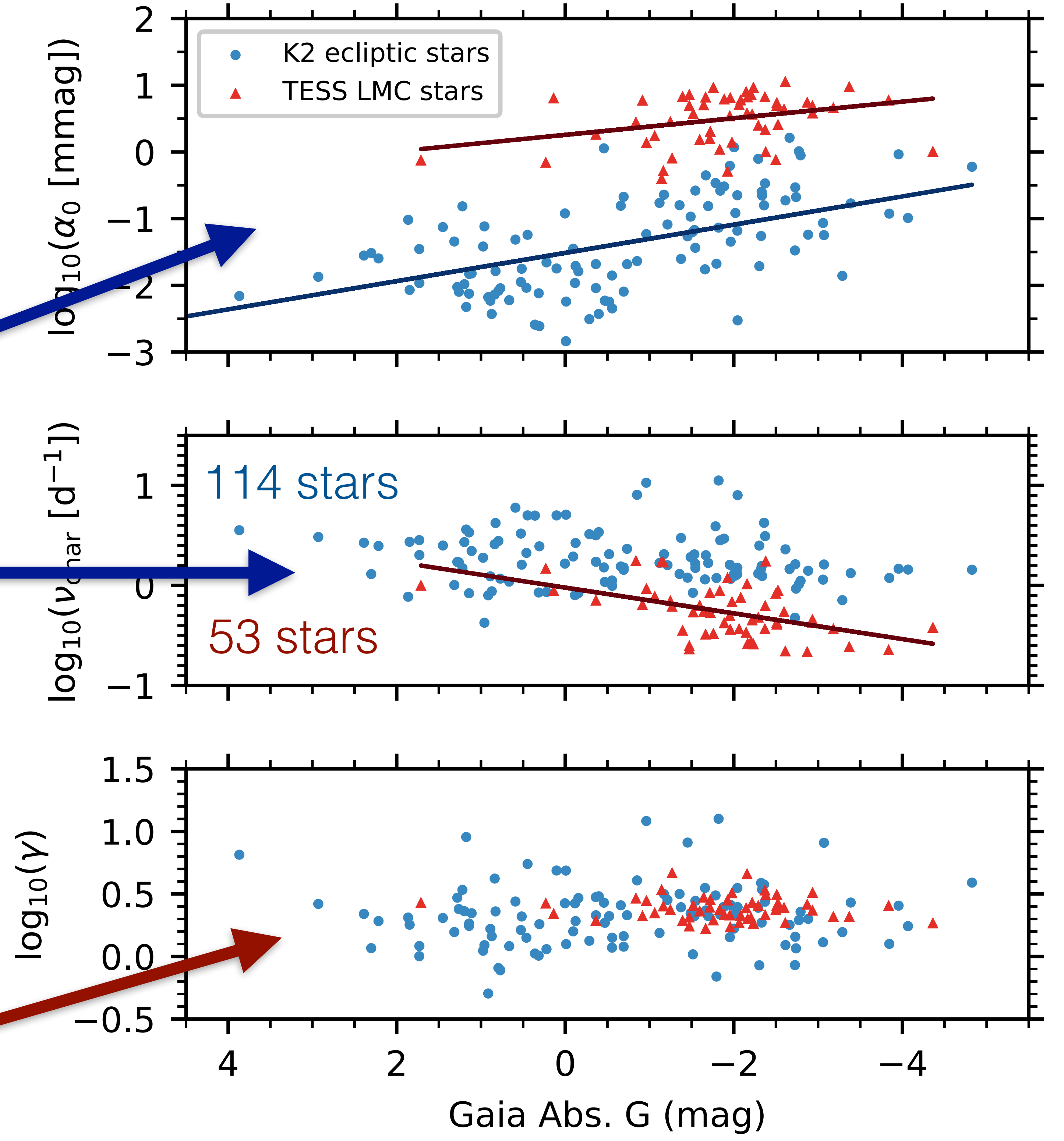
K2 and TESS photometry of blue supergiants



$$\alpha(\nu) = \frac{\alpha_0}{1 + \left(\frac{\nu}{\nu_c}\right)^{\gamma}} + C$$

Brighter and more massive stars have larger IGW amplitudes and lower IGW frequencies

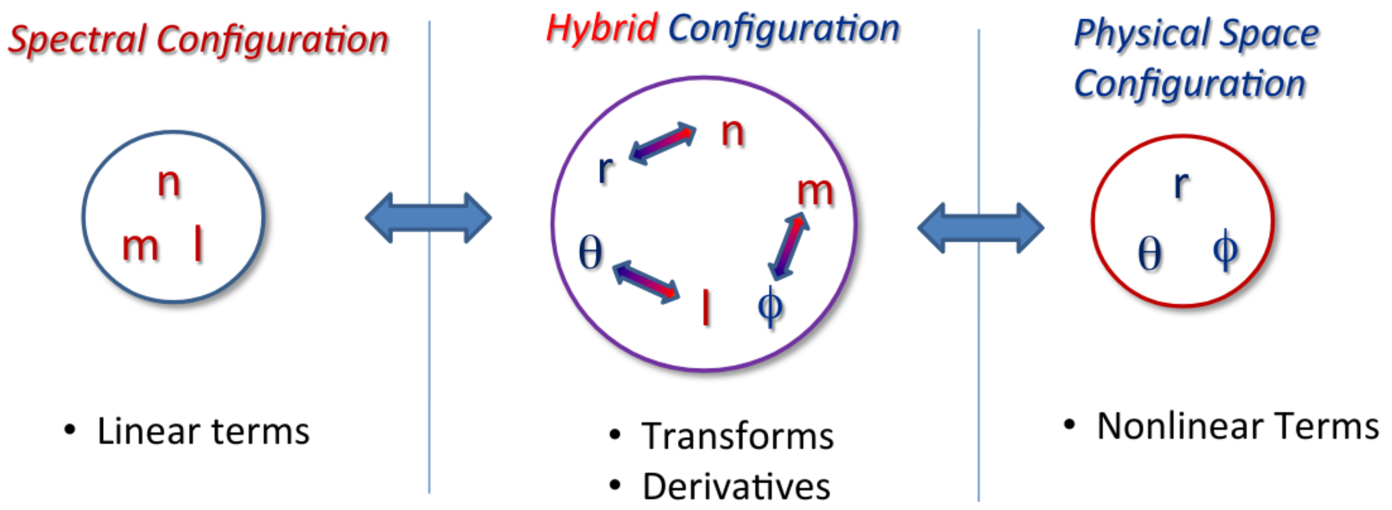
IGW morphology is insensitive to metallicity



(Bowman et al. 2019b)

Rayleigh code

- 3D pseudo-spectral MHD code
- original developer: Nick Featherstone (SWRI Boulder)
- openly developed on GitHub with a team of 6 core developers
- GPLv3 licensed github.com/geodynamics/Rayleigh
- 2D domain decomposition (more efficient parallelization)
- efficient scaling up to 10^4 cores (see Matsui et al., 2016)
- custom reference states (e.g., from MESA stellar evolution code)



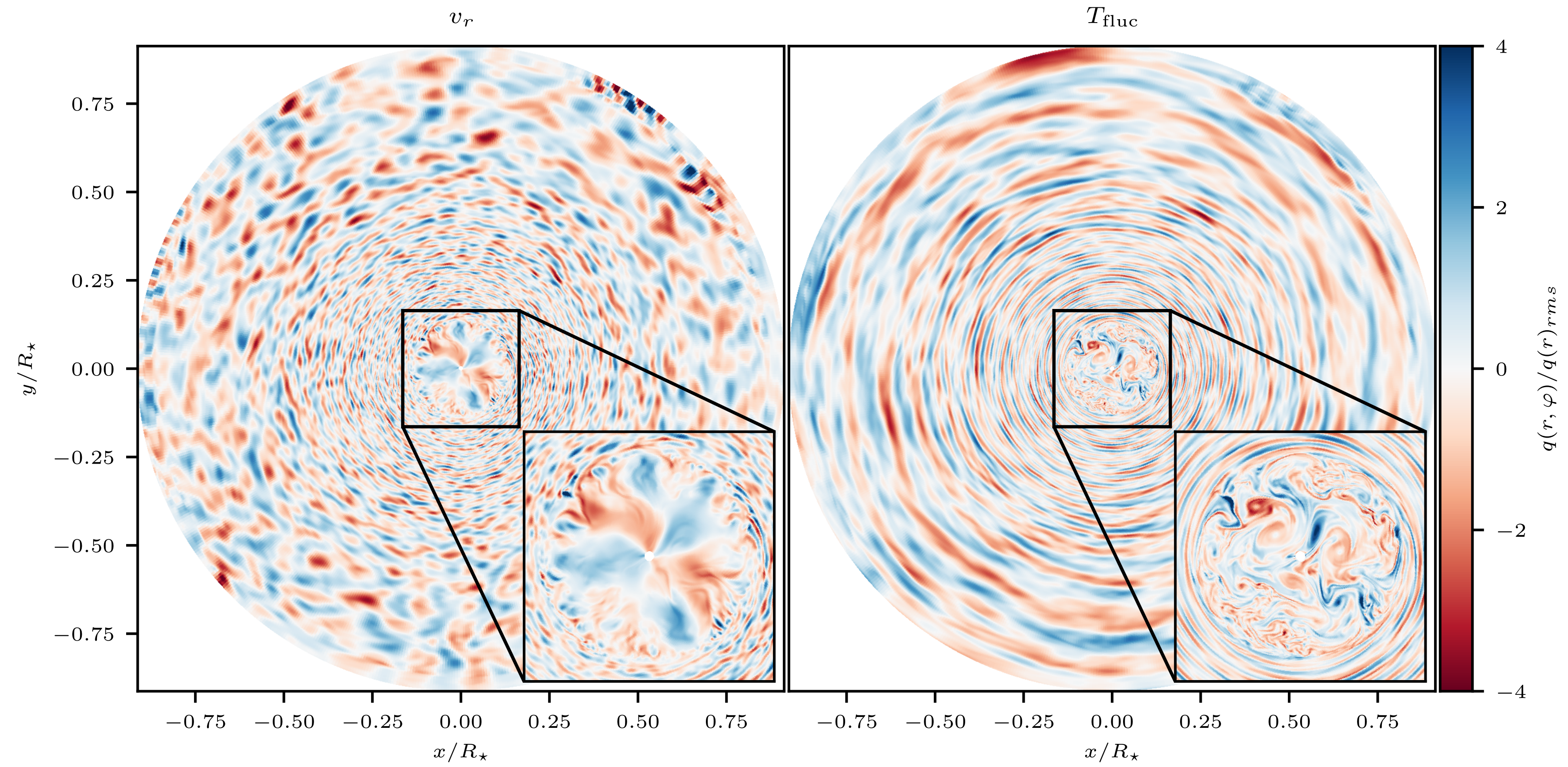
credit: Featherstone (2015)

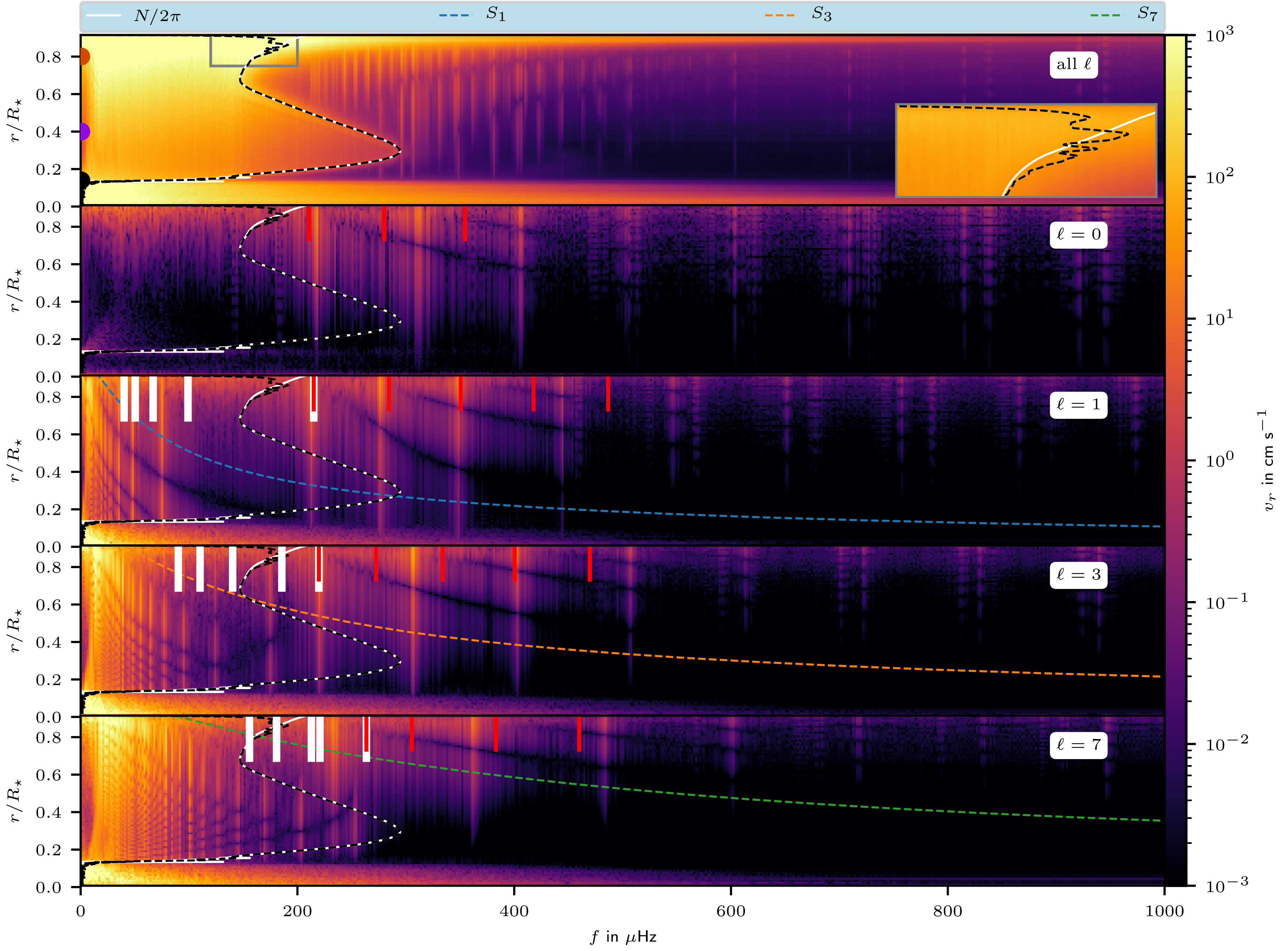
COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

Comparision with FV codes

SLH simulations (Leo Horst, formerly HITS Heidelberg)

- 2D (equatorial annulus)
- no explicit viscosity, stellar thermal diffusivity
- $L = 10^3 \, L_\star$
- fully compressible
- low Mach solver: AUSM$^+$-up
- both IGWs and pressure modes





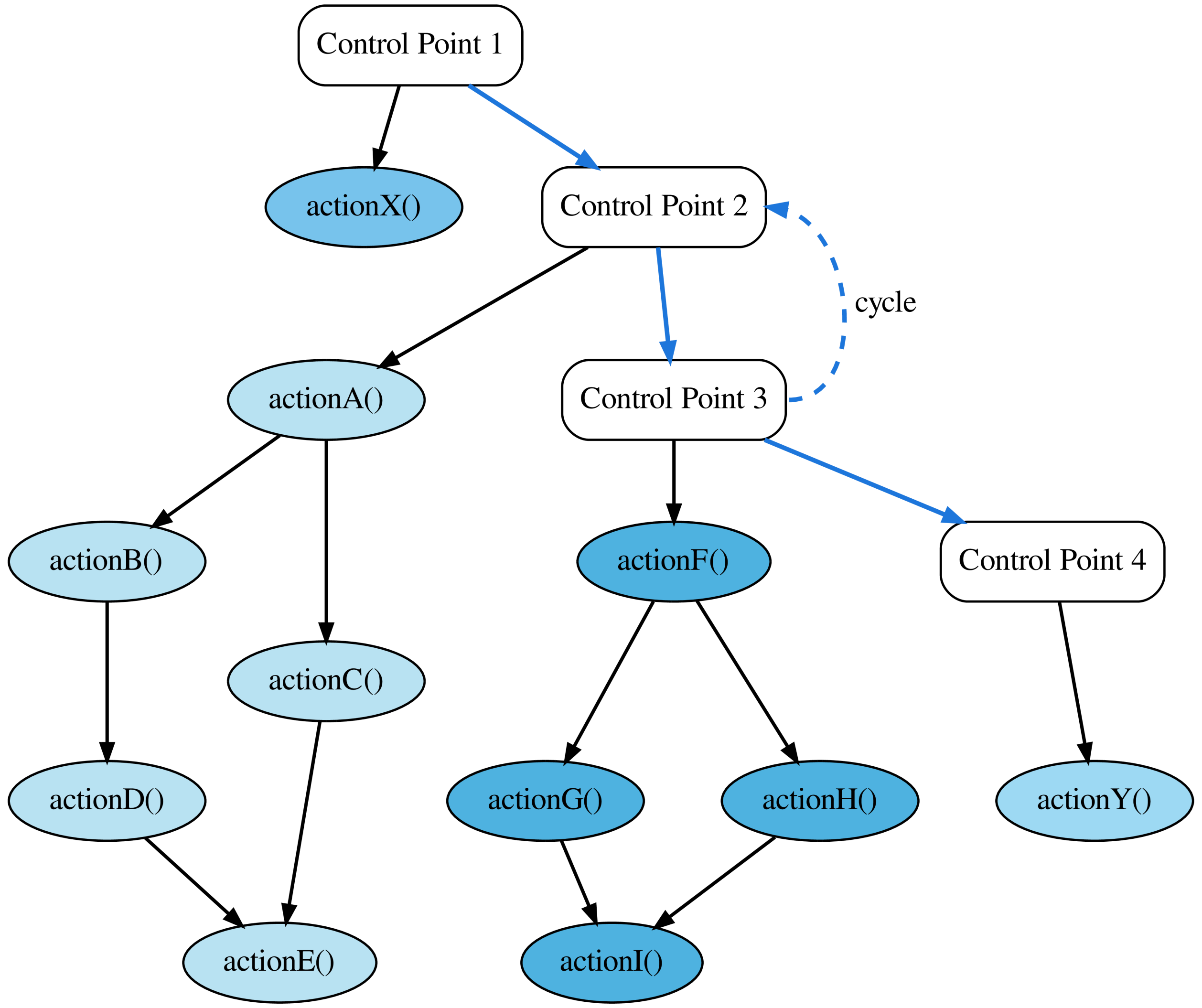
Horst+ (2020)

Some Self-Advertising

Flexible Computational Science (FleCSI) Project

- task-based, data-centric C++ programming model for multiphysics codes
- abstraction layer for (task-based) parallelism backends: MPI, Legion, HPX
- not tied to any specific topology: n-dimensional array, unstructured mesh, tree of particles, …
- tasks are organized by control points for later extensibility without modifying other parts of the code
- Kokkos for shared memory and accelerator support
- permissive license github.com/flecsi/flecsi

FleCSI



Ristra Project

Build multiphysics codes using FleCSI

- FleCSALE: unstructued mesh Eulerian and ALE hydro code
- FleCSALE-mm: + multi-material hydrodynamics
- Symphony: multi-material radiation hydrodynamics
- FleCSPH: smoothed-particle hydrodynamics (SPH) code

github.com/laristra

Conclusions

- Hydrodynamics can be used to study the behavior of convective boundaries.
- It is important to use the right schemes compatible with low Mach numbers and gravity.
- There is reasonable agreement between different hydro codes.
  (But there still needs to be more comparison to spectral methods.)
- The waves in radiation zones allow us to infer interior properties from observations.
- Hydrodynamics will not replace stellar evolution codes but allows us to check assumptions.

Time for questions/discussion

Taylor–Green Vortex

Taylor & Green (1937)

- decaying 3D vortex to test development and decay of turbulent velocity spectrum
- Cartesian box
- no gravity
- gives a measure of numerical Re

Taylor–Green Vortex

Taylor & Green (1937)

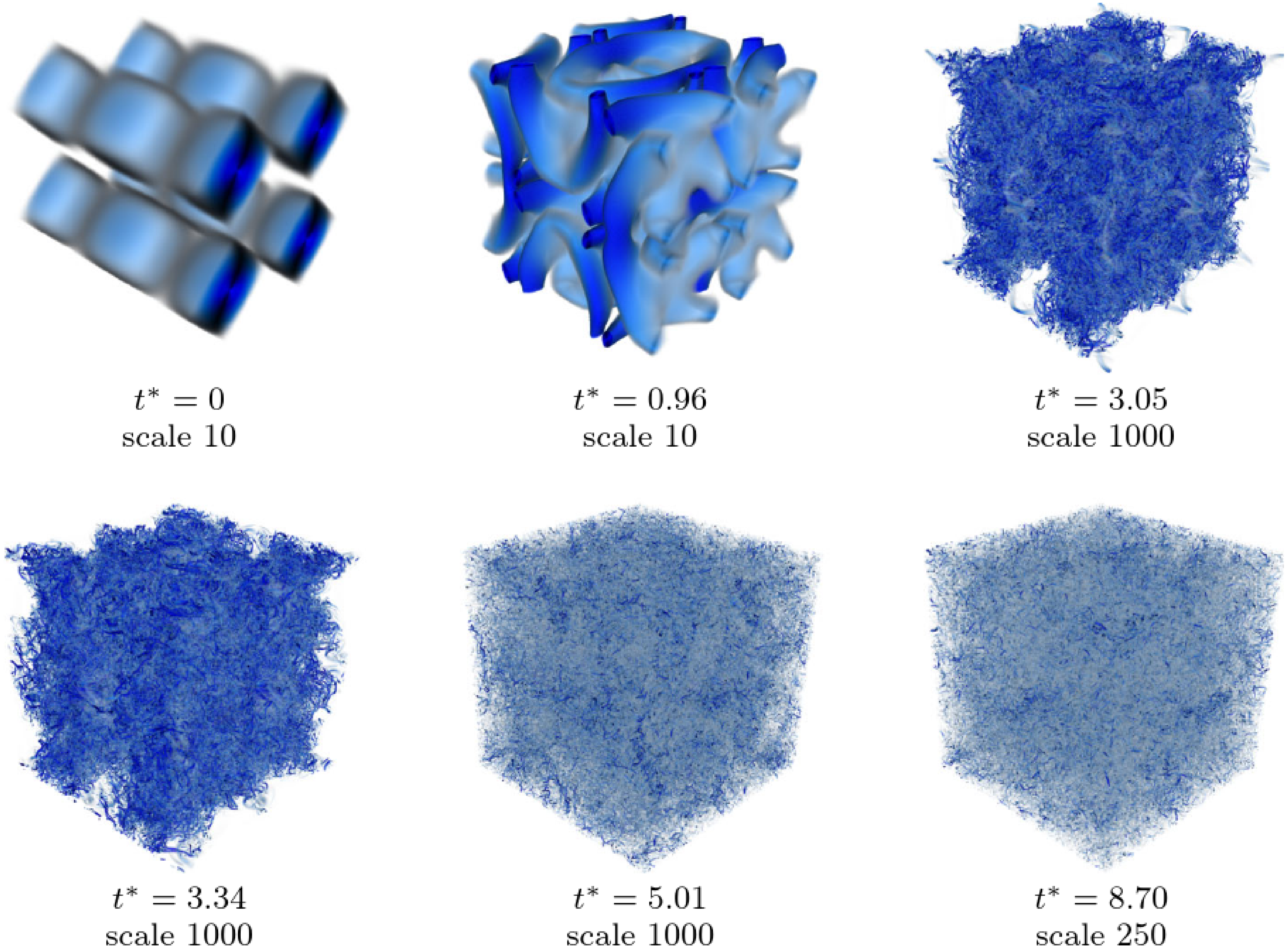
- decaying 3D vortex to test development and decay of turbulent velocity spectrum
- Cartesian box
- no gravity
- gives a measure of numerical *Re*

Initial Condition

Drikakis+ (2007)

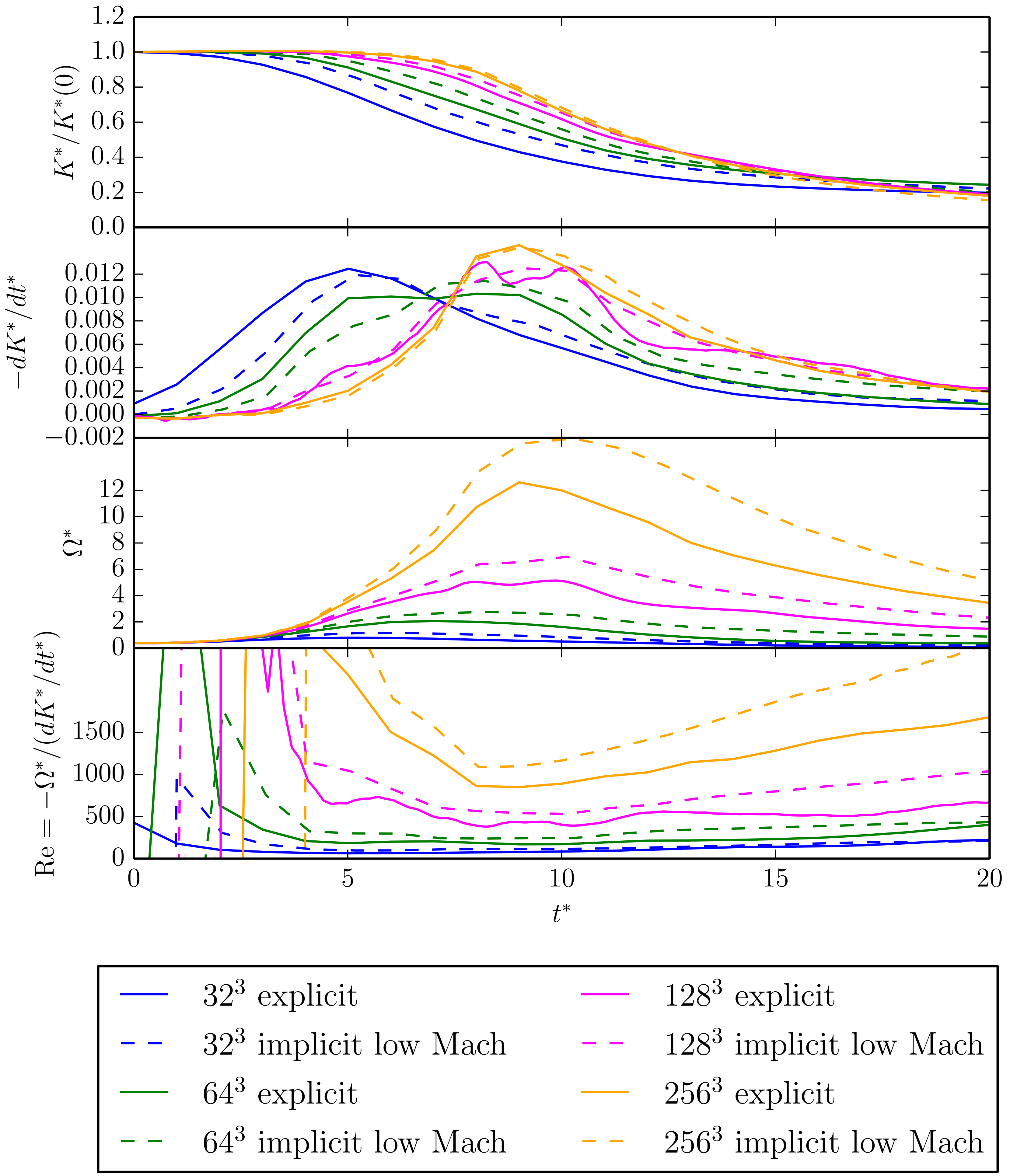
$$\rho(t=0) = \rho_0 = 1.178 \times 10^{-3},$$
$$u(t=0) = u_0 \sin(kx)\cos(ky)\cos(kz),$$
$$v(t=0) = -u_0 \cos(kx)\sin(ky)\cos(kz),$$
$$w(t=0) = 0,$$
$$u_0 = 10^4,$$
$$k = 10^{-2}$$

$$p(t=0) = p_0 + \left[u_0{}^2\rho/16\right]\left[2 + \cos\frac{2z}{100}\right]\left[\cos\frac{2x}{100} + \cos\frac{2y}{100}\right],$$
$$p_0 = 10^6.$$

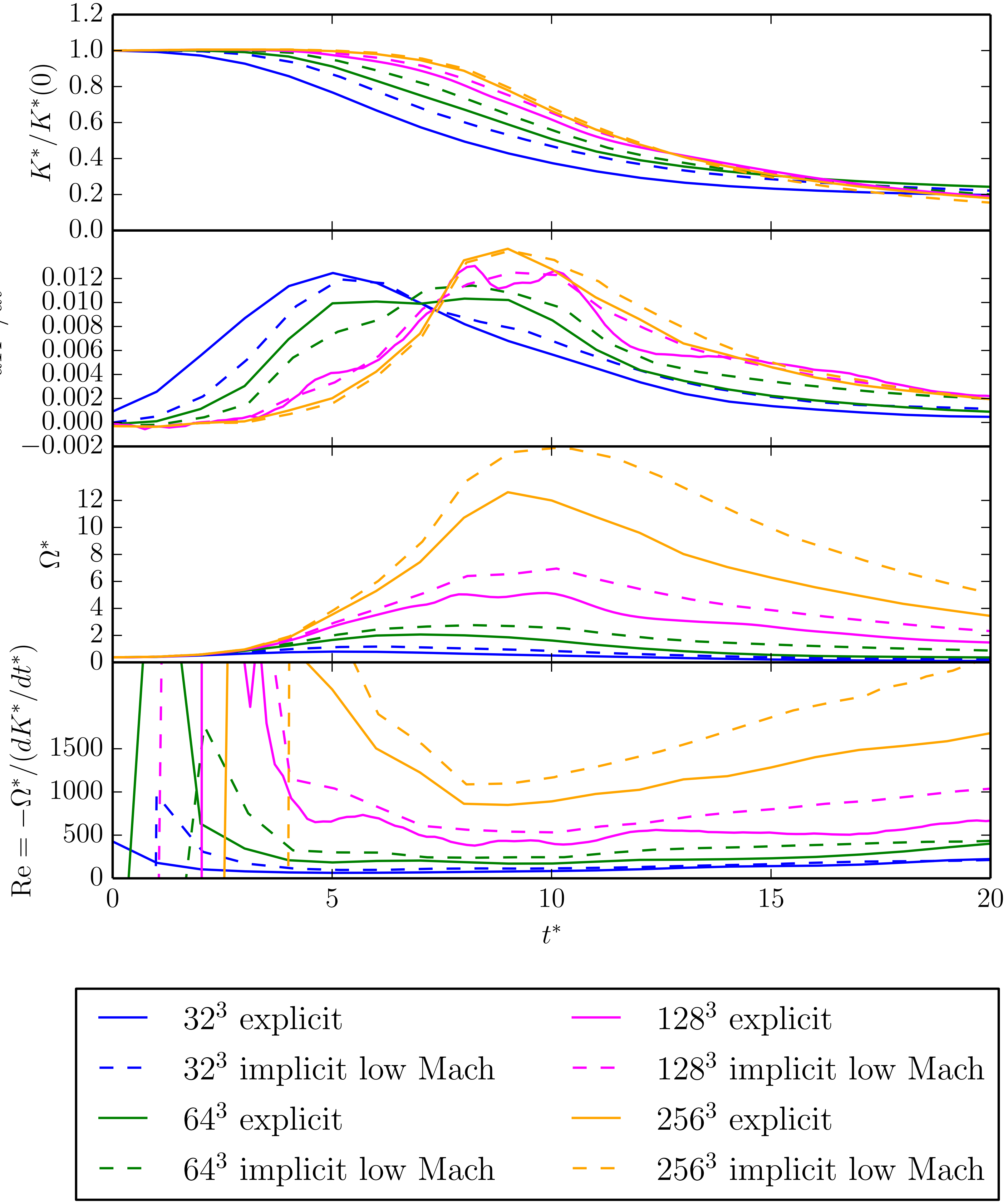

$t^* = 0$
scale 10
$t^* = 0.96$
scale 10
$t^* = 3.05$
scale 1000
$t^* = 3.34$
scale 1000
$t^* = 5.01$
scale 1000
$t^* = 8.70$
scale 250

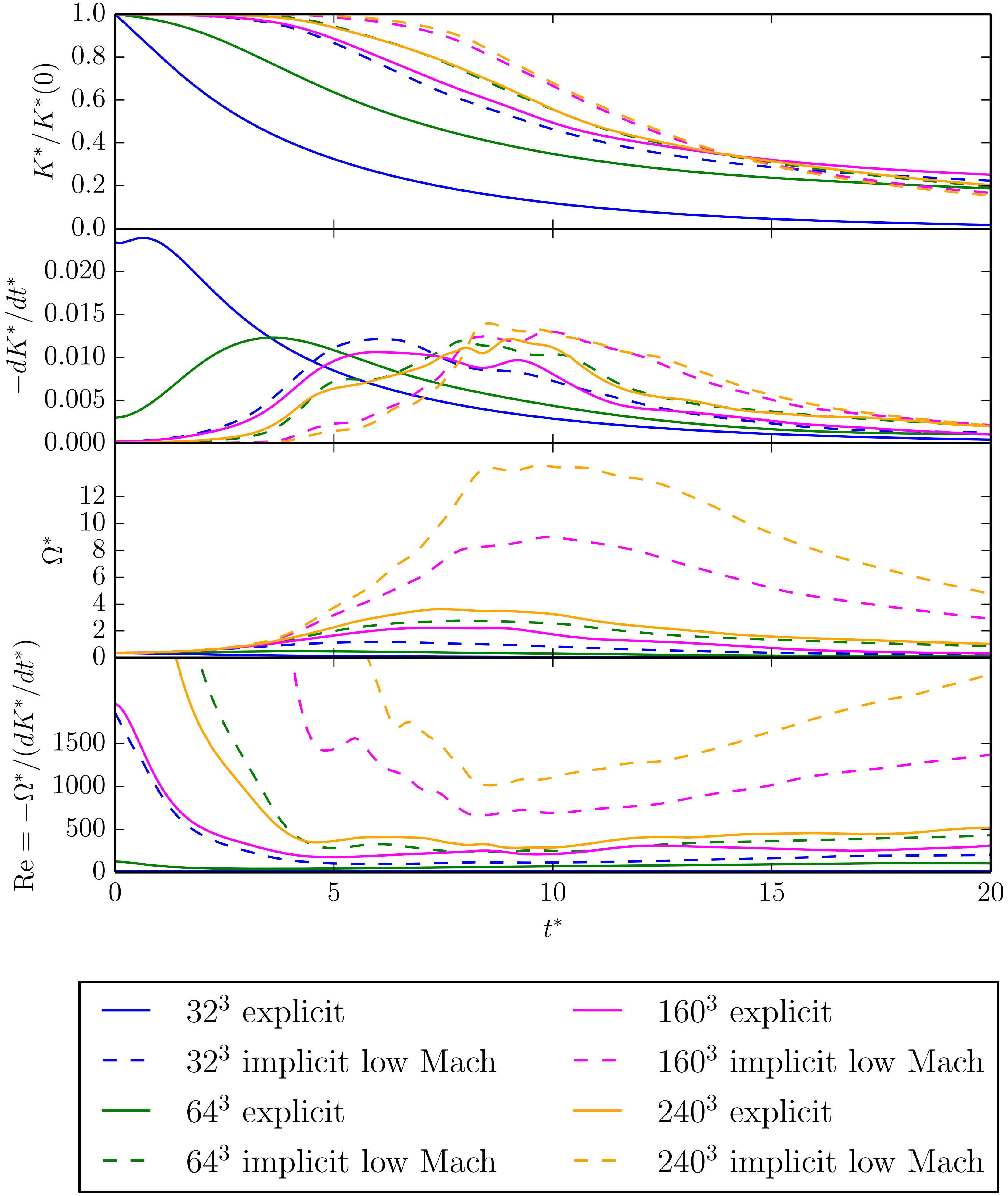
Numerical Reynolds Number

- kinetic energy dissipation rate: $\frac{dK}{dt}$
- enstrophy: $\Omega = \frac{1}{2}\langle|\nabla \times \mathbf{v}|^2\rangle$
- in incompressible limit: $\frac{dK}{dt} = -\eta\Omega$
- non-dimensional: $\frac{dK^*}{dt^*} = -\frac{\Omega^*}{Re}$



$K^*/K^*(0)$
$-dK^*/dt^*$
Ω^*
$\mathrm{Re} = -\Omega^*/(dK^*/dt^*)$
t^*
32^3 explicit
32^3 implicit low Mach
64^3 explicit
64^3 implicit low Mach
128^3 explicit
128^3 implicit low Mach
256^3 explicit
256^3 implicit low Mach


K*/K*(0)
−dK*/dt*
Ω*
Re = −Ω*/(dK*/dt*)
t*
32³ explicit
32³ implicit low Mach
64³ explicit
64³ implicit low Mach
128³ explicit
128³ implicit low Mach
256³ explicit
256³ implicit low Mach


K*/K*(0)
−dK*/dt*
Ω*
Re = −Ω*/(dK*/dt*)
t*
32³ explicit
32³ implicit low Mach
64³ explicit
64³ implicit low Mach
160³ explicit
160³ implicit low Mach
240³ explicit
240³ implicit low Mach

Deviation Method

Berberich+ (2020)

known stationary solution $\tilde{\mathbf{U}}$ ($\mathbf{v}$ can be nonzero): $\frac{\partial \tilde{\mathbf{U}}}{\partial t} = 0$

$$\frac{\partial \mathbf{F}(\tilde{\mathbf{U}})}{\partial x} + \frac{\partial \mathbf{G}(\tilde{\mathbf{U}})}{\partial y} + \frac{\partial \mathbf{H}(\tilde{\mathbf{U}})}{\partial z} = \mathbf{S}(\tilde{\mathbf{U}})$$

Deviation Method

Berberich+ (2020)

known stationary solution $\tilde{\mathbf{U}}$ ($\mathbf{v}$ can be nonzero): $\dfrac{\partial \tilde{\mathbf{U}}}{\partial t} = 0$

$$\frac{\partial \mathbf{F}(\tilde{\mathbf{U}})}{\partial x} + \frac{\partial \mathbf{G}(\tilde{\mathbf{U}})}{\partial y} + \frac{\partial \mathbf{H}(\tilde{\mathbf{U}})}{\partial z} = \mathbf{S}(\tilde{\mathbf{U}})$$

subtract equilibrium eq. from Euler eq. for arbitrary $\mathbf{U}$, expressed using $\Delta \mathbf{U} = \mathbf{U} - \tilde{\mathbf{U}}$

Deviation Method (continued)

$\Delta\mathbf{U}$ at next step is calculated via:

$$\frac{\partial(\Delta\mathbf{U})_{i,j,k}}{\partial t} = \mathbf{F}^{\mathrm{dev}}_{i-\frac{1}{2},j,k} - \mathbf{F}^{\mathrm{dev}}_{i+\frac{1}{2},j,k} + \mathbf{G}^{\mathrm{dev}}_{i,j-\frac{1}{2},k} - \mathbf{G}^{\mathrm{dev}}_{i,j+\frac{1}{2},k} + \mathbf{H}^{\mathrm{dev}}_{i,j,k-\frac{1}{2}} - \mathbf{H}^{\mathrm{dev}}_{i,j,k+\frac{1}{2}} + \mathbf{S}^{\mathrm{dev}}_{i,j,k}$$

Deviation Method (continued)

$\Delta\mathbf{U}$ at next step is calculated via:

$$\frac{\partial(\Delta\mathbf{U})_{i,j,k}}{\partial t} = \mathbf{F}^{\text{dev}}_{i-\frac{1}{2},j,k} - \mathbf{F}^{\text{dev}}_{i+\frac{1}{2},j,k} + \mathbf{G}^{\text{dev}}_{i,j-\frac{1}{2},k} - \mathbf{G}^{\text{dev}}_{i,j+\frac{1}{2},k} + \mathbf{H}^{\text{dev}}_{i,j,k-\frac{1}{2}} - \mathbf{H}^{\text{dev}}_{i,j,k+\frac{1}{2}} + \mathbf{S}^{\text{dev}}_{i,j,k}$$

perform reconstruction of $\Delta\mathbf{U}$ only

$$\mathbf{F}^{\text{dev}}_{i+\frac{1}{2},j,k} = \underbrace{\mathbf{F}_{i+\frac{1}{2},j,k}}_{\text{calculated using exact } \tilde{\mathbf{U}} \text{ at interface and reconstructed } \Delta\mathbf{U}} - \underbrace{\mathbf{F}(\tilde{\mathbf{U}})_{i+\frac{1}{2},j,k}}_{\text{a priori known exact value at interface}}$$

Deviation Method (continued)

$\Delta\mathbf{U}$ at next step is calculated via:

$$\frac{\partial(\Delta\mathbf{U})_{i,j,k}}{\partial t} = \mathbf{F}^{\text{dev}}_{i-\frac{1}{2},j,k} - \mathbf{F}^{\text{dev}}_{i+\frac{1}{2},j,k} + \mathbf{G}^{\text{dev}}_{i,j-\frac{1}{2},k} - \mathbf{G}^{\text{dev}}_{i,j+\frac{1}{2},k} + \mathbf{H}^{\text{dev}}_{i,j,k-\frac{1}{2}} - \mathbf{H}^{\text{dev}}_{i,j,k+\frac{1}{2}} + \mathbf{S}^{\text{dev}}_{i,j,k}$$

perform reconstruction of $\Delta\mathbf{U}$ only

$$\mathbf{F}^{\text{dev}}_{i+\frac{1}{2},j,k} = \underbrace{\mathbf{F}_{i+\frac{1}{2},j,k}}_{\text{calculated using exact } \tilde{\mathbf{U}} \text{ at interface and reconstructed } \Delta\mathbf{U}} - \underbrace{\mathbf{F}(\tilde{\mathbf{U}})_{i+\frac{1}{2},j,k}}_{\text{a priori known exact value at interface}}$$

$$\mathbf{S}^{\text{dev}}_{i,j,k} = \mathbf{S}(\Delta\mathbf{U}_{i,j,k} + \tilde{\mathbf{U}}_{i,j,k}) - \underbrace{\mathbf{S}(\tilde{\mathbf{U}})_{i,j,k}}_{\text{a priori known exact value at cell center}}$$

Deviation Method (continued)

$\Delta\mathbf{U}$ at next step is calculated via:

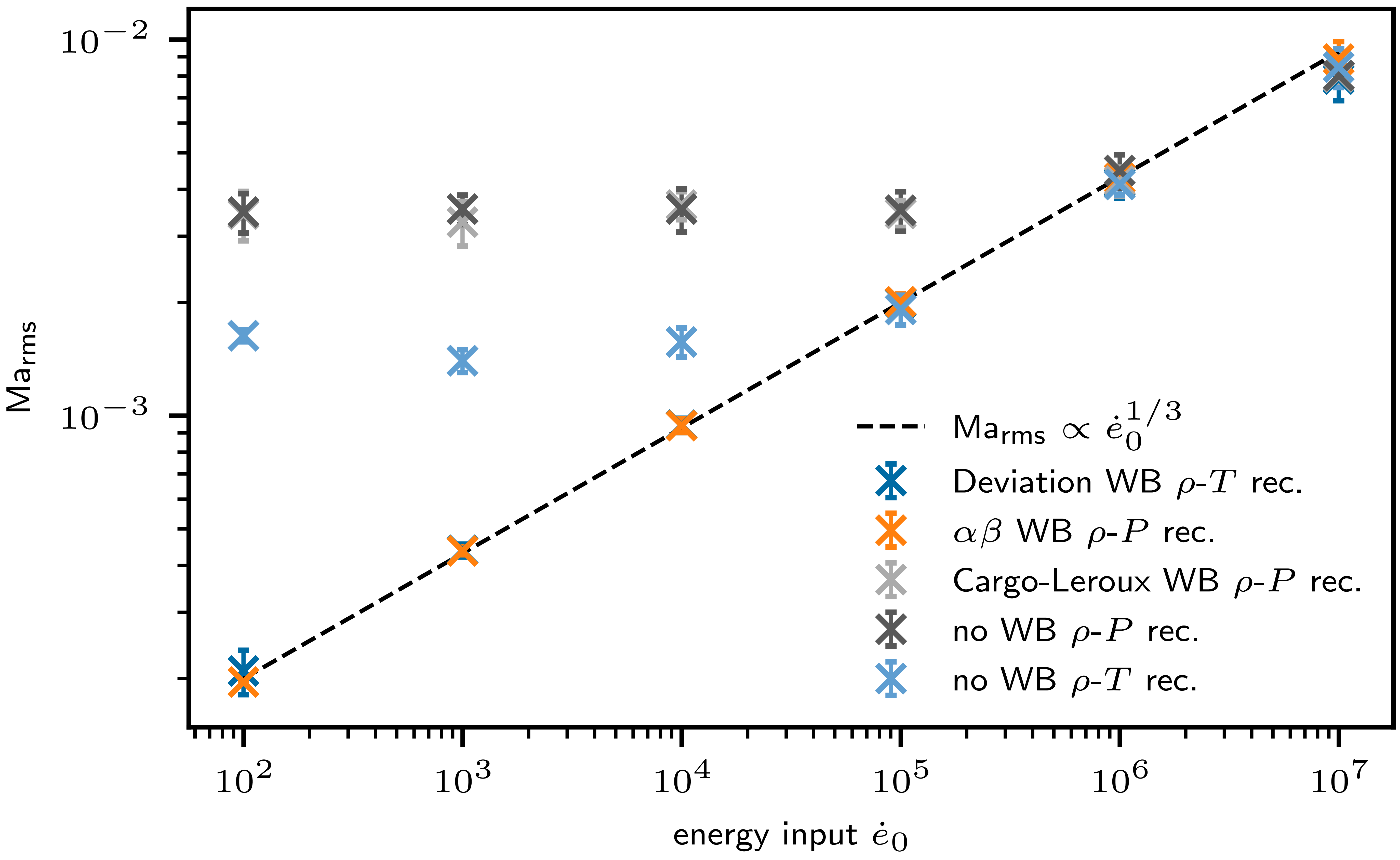
$$\frac{\partial(\Delta\mathbf{U})_{i,j,k}}{\partial t} = \mathbf{F}^{\text{dev}}_{i-\frac{1}{2},j,k} - \mathbf{F}^{\text{dev}}_{i+\frac{1}{2},j,k} + \mathbf{G}^{\text{dev}}_{i,j-\frac{1}{2},k} - \mathbf{G}^{\text{dev}}_{i,j+\frac{1}{2},k} + \mathbf{H}^{\text{dev}}_{i,j,k-\frac{1}{2}} - \mathbf{H}^{\text{dev}}_{i,j,k+\frac{1}{2}} + \mathbf{S}^{\text{dev}}_{i,j,k}$$

perform reconstruction of $\Delta\mathbf{U}$ only

$$\mathbf{F}^{\text{dev}}_{i+\frac{1}{2},j,k} = \underbrace{\mathbf{F}_{i+\frac{1}{2},j,k}}_{\text{calculated using exact } \tilde{\mathbf{U}} \text{ at interface and reconstructed } \Delta\mathbf{U}} - \underbrace{\mathbf{F}(\tilde{\mathbf{U}})_{i+\frac{1}{2},j,k}}_{\text{a priori known exact value at interface}}$$

$$\mathbf{S}^{\text{dev}}_{i,j,k} = \mathbf{S}(\Delta\mathbf{U}_{i,j,k} + \tilde{\mathbf{U}}_{i,j,k}) - \underbrace{\mathbf{S}(\tilde{\mathbf{U}})_{i,j,k}}_{\text{a priori known exact value at cell center}}$$

This can be combined with any high-order method and works for any stationary solution.



Edelmann+ (2021)