



HAL
open science

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

R. Andrassy, J. Higl, H. Mao, M. Mocák, D. G. Vlaykov, W. D. Arnett, I. Baraffe, S. W. Campbell, T. Constantino, P. V. F. Edelmann, et al.

► **To cite this version:**

R. Andrassy, J. Higl, H. Mao, M. Mocák, D. G. Vlaykov, et al.. Dynamics in a stellar convective layer and at its boundary: Comparison of five 3D hydrodynamics codes. *Astronomy & Astrophysics - A&A*, 2022, 659, <10.1051/0004-6361/202142557>. <insu-03711529>

HAL Id: insu-03711529

<https://insu.hal.science/insu-03711529v1>

Submitted on 1 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

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

R. Andrassy¹, J. Higl¹, H. Mao², M. Mocák³, D. G. Vlaykov⁴, W. D. Arnett⁵, I. Baraffe^{4,6}, S. W. Campbell^{7,8},
T. Constantino⁴, P. V. F. Edelmann⁹, T. Goffrey¹⁰, T. Guillet⁴, F. Herwig^{11,12}, R. Hirschi^{3,13}, L. Horst¹, G. Leidi^{1,14},
C. Meakin^{5,15}, J. Pratt¹⁶, F. Rizzuti³, F. K. Röpké^{1,17}, and P. Woodward^{2,12}

¹ Heidelberger Institut für Theoretische Studien, Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany
e-mail: robert.andrassy@h-its.org

² LCSE and Department of Astronomy, University of Minnesota, Minneapolis, MN 55455, USA

³ Astrophysics Group, Keele University, Keele, Staffordshire ST5 5BG, UK

⁴ Physics and Astronomy, University of Exeter, Exeter EX4 4QL, UK

⁵ Steward Observatory, University of Arizona, 933 N. Cherry Avenue, Tucson, AZ 85721, USA

⁶ École Normale Supérieure, Lyon, CRAL (UMR CNRS 5574), Université de Lyon, France

⁷ School of Physics and Astronomy, Monash University, Victoria 3800, Australia

⁸ ARC Centre of Excellence for All Sky Astrophysics in Three Dimensions (ASTRO-3D), Australia

⁹ X Computational Physics (XCP) Division and Center for Theoretical Astrophysics (CTA), Los Alamos National Laboratory, Los Alamos, NM 87545, USA

¹⁰ Centre for Fusion, Space and Astrophysics, Department of Physics, University of Warwick, Coventry CV4 7AL, UK

¹¹ Department of Physics and Astronomy, University of Victoria, Victoria, BC V8W 2Y2, Canada

¹² Joint Institute for Nuclear Astrophysics, Center for the Evolution of the Elements, Michigan State University, 640 South Shaw Lane, East Lansing, MI 48824, USA

¹³ Kavli Institute for the Physics and Mathematics of the Universe (WPI), University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa 277-8583, Japan

¹⁴ Zentrum für Astronomie der Universität Heidelberg, Astronomisches Rechen-Institut, Mönchhofstr. 12-14, 69120 Heidelberg, Germany

¹⁵ Pasadena Consulting Group, 1075 N Mar Vista Ave, Pasadena, CA 91104, USA

¹⁶ Department of Physics and Astronomy, Georgia State University, Atlanta, GA 30303, USA

¹⁷ Zentrum für Astronomie der Universität Heidelberg, Institut für Theoretische Astrophysik, Philosophenweg 12, 69120 Heidelberg, Germany

Received 31 October 2021 / Accepted 29 December 2021

ABSTRACT

Our ability to predict the structure and evolution of stars is in part limited by complex, 3D hydrodynamic processes such as convective boundary mixing. Hydrodynamic simulations help us understand the dynamics of stellar convection and convective boundaries. However, the codes used to compute such simulations are usually tested on extremely simple problems and the reliability and reproducibility of their predictions for turbulent flows is unclear. We define a test problem involving turbulent convection in a plane-parallel box, which leads to mass entrainment from, and internal-wave generation in, a stably stratified layer. We compare the outputs from the codes FLASH, MUSIC, PPMSTAR, PROMPT, and SLH, which have been widely employed to study hydrodynamic problems in stellar interiors. The convection is dominated by the largest scales that fit into the simulation box. All time-averaged profiles of velocity components, fluctuation amplitudes, and fluxes of enthalpy and kinetic energy are within $\lesssim 3\sigma$ of the mean of all simulations on a given grid (128^3 and 256^3 grid cells), where σ describes the statistical variation due to the flow's time dependence. They also agree well with a 512^3 reference run. The 128^3 and 256^3 simulations agree within 9% and 4%, respectively, on the total mass entrained into the convective layer. The entrainment rate appears to be set by the amount of energy that can be converted to work in our setup and details of the small-scale flows in the boundary layer seem to be largely irrelevant. Our results lend credence to hydrodynamic simulations of flows in stellar interiors. We provide in electronic form all outputs of our simulations as well as all information needed to reproduce or extend our study.

Key words. hydrodynamics – convection – turbulence – stars: interiors – methods: numerical

1. Introduction

Convection is the most important mixing process in stars. Due to the high density and opacity deep in the stellar interior, the convection is almost adiabatic and Mach numbers are typically $\mathcal{M} < 0.05$ in late evolutionary phases of massive stars and $10^{-4} \lesssim \mathcal{M} \lesssim 10^{-3}$ in early phases, such as core convection during H and He burning. Such slow flows are nearly incom-

pressible in the sense that ram pressure is much smaller than thermal pressure, although significant compression and expansion still occur when fluid packets are displaced radially in the strong, nearly hydrostatic pressure stratification. The spatial scales involved are so large that molecular viscosity is negligible and the flow is highly turbulent. The main effect of convection on the structure and evolution of stars is the transport of species, energy, and angular momentum. The mixing of chemical species

has important implications for the generation of nuclear energy, origin of elements, stellar lifetimes, or the observable properties of stars and stellar populations. The physics of convection is key to determining the endpoints of stellar evolution which in turn give rise to some of the most energetic events in the Universe, such as supernova explosions and compact-object mergers.

Despite its importance, the treatment of convective mixing and energy transport is still very rudimentary in stellar-evolution models. This is because the long thermal and nuclear timescales of stellar evolution make 1D, time-averaged models the only practical approach. The mixing-length theory (MLT; Prandtl 1925; Böhm-Vitense 1958) is the most common parametric description of convection in stellar-evolution codes. The MLT's main parameter, the mixing length α , is usually calibrated such that the code reproduces current properties of the Sun and α is then assumed to be the same in all convective layers in all stars. Three-dimensional hydrodynamic simulations of near-surface convection show that α varies across the Hertzsprung-Russell diagram (Trampedach et al. 2014; Magic et al. 2015; Sonoi et al. 2019). Trampedach et al. (2014) and Jørgensen et al. (2018) illustrate how such realistic 3D models can be used to replace the MLT in near-surface layers in stellar-evolution calculations. Deep in the stellar interior, which is the main focus of our work, convection is so efficient that the stratification becomes isentropic almost independently of α . The MLT is then used to get an estimate of mixing speed, which is important for the stratification of species when the timescale of some nuclear reaction(s) becomes comparable to or shorter than the mixing timescale across the convective layer. Nevertheless, the stellar model remains 1D, and the mixing is usually described as a diffusive process, although alternative approaches are being developed (e.g., Stephens et al. 2021).

The MLT is a local theory and as such it cannot describe any mixing caused by the non-local nature of convection and reaching beyond the formal convective boundary. Traditionally known as ‘convective overshooting’, this mixing may involve a number of distinct physical processes and it is better described by the more general term ‘convective boundary mixing’ (CBM; Denissenkov et al. 2013). CBM enlarges stellar convection zones and it is required to explain a large number of observations such as the position of the turn-off point in open clusters (Rosvick & Vandenberg 1998), the width of the main sequence in the Hertzsprung-Russell diagram (Castro et al. 2014), properties of double-lined eclipsing binaries (Claret & Torres 2019, CBM;), or asteroseismic observations of massive stars (Aerts 2021, and references therein). In calculations of stellar evolution, CBM is added in the form of a simplistic model, such as instantaneous overshooting (Maeder 1976; Ekström et al. 2012), a diffusive model (Herwig 2000; Battino et al. 2016; Baraffe et al. 2017), or an entrainment model (Staritsin 2013; Scott et al. 2021). Each of these formulations has at least one free parameter, which is usually calibrated by comparison with observations. However, there is no reason to expect that the same parameter(s) should apply across different convection zones in different stars at different stages of evolution.

Multiple research groups have started to use 2D and 3D hydrodynamic codes to simulate stellar convection and CBM in order to calibrate free parameters of the CBM models mentioned above, for example, Meakin & Arnett (2007), Jones et al. (2017), Pratt et al. (2017), Cristini et al. (2019), Denissenkov et al. (2019), and Horst et al. (2021). Although the growth rates of convective layers in such simulations seem to be reasonable for late stages of stellar evolution (Meakin & Arnett 2007;

Woodward et al. 2015; Jones et al. 2017; Mocák et al. 2018), they are much too high to be compatible with observations on the main-sequence (Meakin & Arnett 2007; Gilet et al. 2013; Scott et al. 2021). It is not clear whether this is because the simulations’ parameters are too far from the stellar regime, or it is due to some physics not being included in the simulations, or it is a sign of numerical problems.

Simulations of convection on the main sequence have become even more attractive with the recent boom in asteroseismology. The generation of waves by convection and their propagation through the rest of the star can be observed in 2D and 3D simulations directly, although most codes require an artificial increase in the energy generation rate to speed up the flow. The resulting wave spectra can, in principle, be compared with observations of real stars. However, some groups obtain spectra with many strong resonance lines (Lecoanet et al. 2021) whereas others get featureless, continuous spectra (Alvan et al. 2014; Edelmann et al. 2019; Horst et al. 2020). Again, it is unclear whether the difference can be tracked down to physical assumptions or they are of numerical origin.

Probably all major hydrodynamics codes used in the field have passed a series of tests using problems with known solutions such as shock tubes or various instabilities during their initial growth. However, the stellar hydrodynamic cases that we are interested in are much more complex and exact solutions do not exist. Instead, code verification is done by comparing solutions obtained with different codes. Such exercises have a long tradition in computational dynamics (Joggerst et al. 2014; Ramaprabhu et al. 2012; Dimonte et al. 2004) and also in astrophysics (Doherty et al. 2010; Beeck et al. 2012; McNally et al. 2012; Kim et al. 2014, 2016; Lecoanet et al. 2016; Christensen-Dalsgaard et al. 2020; Silva Aguirre et al. 2020; Fleck et al. 2021). Nonetheless, a test problem focused on the dynamics of convection and mixing processes in stellar interiors has been missing in the literature.

We constructed a complex test problem as detailed in Sect. 2.2. It involves stratified, turbulent convection, and CBM as well as the generation and propagation of internal gravity waves in a plane-parallel box. The presence of turbulence makes this problem fundamentally different from classical test cases such as shock tubes. Even if the initial and boundary conditions are given, the chaotic nature of the flow leads to rapid amplification of small perturbations in numerical simulations as well as in real flows. Nevertheless, many space- and time-averaged quantities are expected to converge upon grid refinement in a statistical way, for instance velocity profiles, spectra, fluxes, or the cumulative amount of CBM¹. The statistical variation left in these averages decreases as the length of the averaging time interval is increased. However, our test problem, designed to be as close as possible to real science cases, involves continuous heating and mass entrainment. The resulting secular evolution limits the extent of time averaging applicable and we must estimate the magnitude of statistical variation when comparing results of such simulations. Still, we believe that our test problem can be used to compare codes with accuracy appropriate to the current state of the field.

We ran simulations using the codes FLASH, MUSIC, PPMSTAR, PROMPI, and SLH, which have been widely used to study hydrodynamic problems in stellar interiors and are briefly described in Sect. 2.3. The simulations are compared in a number of metrics: velocity amplitudes, profiles, and spectra (Sect. 3.1),

¹ Vorticity is a good example of a quantity that diverges upon grid refinement in simulations of inviscid turbulence.

properties of the convective boundary and the mass entrainment rate (Sect. 3.2), the amplitudes of fluctuations, and energy fluxes (Sect. 3.3). Finally, we summarise our results in Sect. 4. Our selection of codes for this study is based on collaborative ties and availability of resources and it is far from being unbiased or complete. In particular, our study does not include any finite-difference and spectral schemes. However, we provide our test setup as well as all data-analysis scripts in electronic form for anyone interested in extending this study, see Appendix A.

2. Methods

2.1. Motivation of the oxygen shell test problem

In this study, we aim to define a problem that involves turbulent convection and CBM while being simple enough to be accessible to as many codes as possible. We define our test problem to be similar to the simulations of Jones et al. (2017) and Andrassy et al. (2020) of shell oxygen burning in a massive star. The stratification of the underlying stellar model is compared with that of our test problem (specified in Sect. 2.2) in Fig. 1. Although the oxygen shell is spherical, we use plane-parallel geometry both for simplicity and to reduce computational costs. The lower half of the simulation domain is initially isentropic and the upper half is stably stratified and approximately follows the density stratification of the stellar model. Because we intend to study the dynamics of a single convective boundary in isolation, we do not include the convective carbon-burning shell, the bottom of which corresponds to the discontinuity at $r \approx 2.35$ in the stellar model.

To further simplify the problem, we follow Jones et al. (2017) and Andrassy et al. (2020) and use the ideal-gas equation of state, neglect neutrino cooling, and replace nuclear reactions with a constant and easy-to-resolve heating profile. Compared with the original stellar model, the total luminosity driving convection is 22.5 times larger and the convective layer contains, due to its different geometry, 5.4 times less mass. The resulting rms Mach number of the convective flow is then ≈ 0.04 , making the problem well accessible to explicit and implicit codes as well as to codes using low-Mach approximations. Molecular viscosity, thermal conduction, and radiative diffusivity are not considered.

We adopt the speed of sound, density, and temperature at the bottom of the convective layer as units of velocity, density, and temperature, respectively, and we take the approximate depth of the convective layer to be the unit of length². The numerical values of these units as well as those of other units derived from them are summarised in Table 1. The dimensionless problem is specified in detail in the following section.

2.2. Problem specification

The initial hydrostatic stratification, shown in Fig. 1, is computed using the following profile of gravitational acceleration:

$$g(y) = g_0 f_g(y) y^{-5/4}, \quad (1)$$

where $g_0 = 1.414870$ is the gravitational acceleration at the bottom of the oxygen-burning shell in the stellar model. The vertical coordinate y runs from 1 to 3, which in principle allows the same stratification to be used in spherical coordinates simply by setting the radius $r = y$. This profile is similar to the stellar

² The convective shell's bottom radius happens to be close to its radial extent in the stellar model.

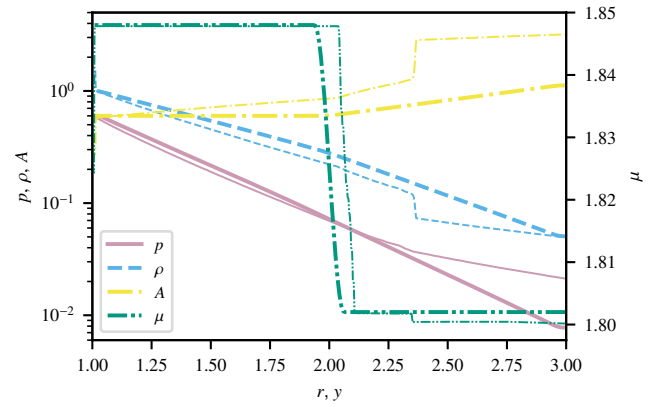


Fig. 1. Initial stratification in the problem units (Table 1) of the pressure p , density ρ , pseudo-entropy $A = p/\rho^\gamma$, and mean molecular weight μ in the stellar model (thin lines) and in the test problem (thick lines). The discontinuity visible in the stellar model at $r \approx 2.35$ is the bottom of the carbon-burning shell, which is not included in our test problem.

Table 1. Basic problem units (upper section), derived units (middle section), and adopted values of physical constants (lower section).

Quantity	Unit	
Density	1.820940×10^6	g cm^{-3}
Length	4.000000×10^8	cm
Temperature	3.401423×10^9	K
Velocity	5.050342×10^8	cm s^{-1}
Acceleration	6.376489×10^8	cm s^{-2}
Energy	2.972468×10^{49}	erg
Luminosity	3.752995×10^{49}	erg s^{-1}
Mass	1.165402×10^{32}	g
Pressure	4.644481×10^{23}	dyn cm^{-2}
Time	7.920256×10^{-1}	s
Volume	6.400000×10^{25}	cm^3
Atomic mass unit	1.660539×10^{-24}	g
Boltzmann constant	1.380649×10^{-16}	erg K^{-1}
Gas constant	8.314463×10^7	$\text{erg g}^{-1} \text{K}^{-1}$
Gravitational constant	6.674300×10^{-8}	$\text{dyn cm}^2 \text{g}^{-2}$

one and the reduction in gravity with height helps prevent a too-fast decrease in the pressure scale height. We ‘turn off’ gravity close to the lower and upper boundaries of the simulation domain using the factor

$$f_g(y) = \begin{cases} \frac{1}{2} \left\{ 1 + \sin \left[16\pi \left(y - \frac{1}{32} \right) \right] \right\}, & \text{for } 1 \leq y < 1 + \frac{1}{16}, \\ 1, & \text{for } 1 + \frac{1}{16} \leq y \leq 3 - \frac{1}{16}, \\ \frac{1}{2} \left\{ 1 + \sin \left[16\pi \left(y - \frac{1}{32} \right) \right] \right\}, & \text{for } 3 - \frac{1}{16} < y \leq 3, \end{cases} \quad (2)$$

which forces density and pressure to become constant close to the boundaries and thus makes possible the use of a simple reflective boundary condition there. The computational domain contains a mixture of two monatomic ideal gases with $\gamma = 5/3$ and mean molecular weights of $\mu_0 = 1.848$ and $\mu_1 = 1.802$. Initially, the convective layer is filled with the μ_0 fluid and the stable layer with the μ_1 fluid. There is a smooth transition between the two layers and the fractional volume η_1 of the μ_1 fluid varies as

$$\eta_1(y) = \begin{cases} 0, & \text{for } 1 \leq y < 2 - \frac{1}{16}, \\ \frac{1}{2} [1 + \sin(8\pi y)], & \text{for } 2 - \frac{1}{16} \leq y \leq 2 + \frac{1}{16}, \\ 1, & \text{for } 2 + \frac{1}{16} < y < 3. \end{cases} \quad (3)$$

The mixture is assumed to be in local pressure and thermal equilibrium everywhere and the stratification follows the piecewise-polytropic pressure-density relation

$$\frac{d \ln p}{d \ln \rho} = \begin{cases} \gamma_0, & \text{for } 1 \leq y < 2 - \frac{1}{16}, \\ \gamma_0 + \eta_1(y)(\gamma_1 - \gamma_0), & \text{for } 2 - \frac{1}{16} \leq y \leq 2 + \frac{1}{16}, \\ \gamma_1, & \text{for } 2 + \frac{1}{16} < y < 3, \end{cases} \quad (4)$$

where $\gamma_0 = 5/3$ and $\gamma_1 = 1.3$. Equations (1)–(4), together with the ideal-gas law define a unique hydrostatic state.

We impose friction-free, non-conductive wall boundary conditions at $y = 1$ and $y = 3$. The specific implementation of these boundary conditions is code-dependent, see Sect. 2.3. The computational domain is periodic in the two horizontal dimensions and spans the coordinate intervals $-1 \leq x \leq 1$ and $-1 \leq z \leq 1$. The initial aspect ratio of the convective layer is thus 2:1 (width to height).

Convection is driven using a time-independent heat source concentrated close to the lower boundary of the convective layer with energy generation rate per unit volume of

$$\dot{q}(y) = \begin{cases} \dot{q}_0 \sin(8\pi y), & \text{for } 1 \leq y \leq 1 + \frac{1}{8}, \\ 0, & \text{for } 1 + \frac{1}{8} < y \leq 3, \end{cases} \quad (5)$$

where $\dot{q}_0 = 3.795720 \times 10^{-4}$. The total luminosity of the heat source is 1.2082151×10^{-4} . When the problem is discretised, we take into account the fact that the average value of $\dot{q}(y)$ over a computational cell of height Δy centred around $y = y_0$ is $\dot{q}(y_0) \sin(4\pi\Delta y)/(4\pi\Delta y)$. The heat source is defined to be smooth and easy to resolve for this study; the energy generation profile in the stellar model is asymmetric and discontinuous (see Fig. 3 of Jones et al. 2017). We do not include any cooling term. The Kelvin-Helmholtz timescales of the convective layer and of the whole computational domain are 1.29×10^4 and 1.43×10^4 time units, respectively.

To break the initial symmetry, we use a density perturbation

$$\frac{\Delta\rho}{\rho_0} = 5 \times 10^{-5} \frac{\dot{q}(y)}{\dot{q}_0} [\sin(3\pi x) + \cos(\pi x)][\sin(3\pi z) - \cos(\pi z)], \quad (6)$$

where $\rho_0 = 1$ is the density at the bottom of the convective layer. The perturbation only affects the heating layer, so it is also concentrated close to the lower boundary of the convective layer. There is no pressure perturbation. The smooth density perturbation, although small, allows us to produce a well-resolved initial transient, which should be similar in all codes.

The problem is described by the inviscid Euler equations with gravity and volume heating,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0, \quad (7)$$

$$\frac{\partial (\rho \mathbf{V})}{\partial t} + \nabla \cdot (\rho \mathbf{V} \otimes \mathbf{V} + p \mathbf{I}) = \rho \mathbf{g}, \quad (8)$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot [(\rho E + p) \mathbf{V}] = \rho \mathbf{V} \cdot \mathbf{g} + \dot{q}, \quad (9)$$

where \mathbf{V} is the velocity vector, \mathbf{I} the unit tensor, \mathbf{g} the gravitational acceleration vector pointed towards the negative y axis, and $E = e + \frac{1}{2} \|\mathbf{V}\|^2$ the specific total energy, which includes the specific internal energy e and the specific kinetic energy $\frac{1}{2} \|\mathbf{V}\|^2$. Some of our codes, as indicated in Table 2, include the specific potential energy Φ in the total energy $E_\Phi = E + \Phi$ and, instead of evolving Eq. (9), they evolve the equivalent equation

$$\frac{\partial (\rho E_\Phi)}{\partial t} + \nabla \cdot [(\rho E_\Phi + p) \mathbf{V}] = \dot{q}. \quad (10)$$

The system of equations is closed by the ideal-gas law

$$p = (\gamma - 1) \rho e \quad (11)$$

with $\gamma = 5/3$. We track the mixing between the two layers by advecting the partial density ρX_i of either of the two fluids,

$$\frac{\partial (\rho X_i)}{\partial t} + \nabla \cdot (\rho X_i \mathbf{V}) = 0, \quad (12)$$

and the other mass fraction follows from the requirement $X_0 + X_1 = 1$. The mass fraction X_i acts as a passive tracer of mixing in our setup, which is a consequence of the set of assumptions we make. The passive nature of composition may be somewhat surprising since composition has a direct influence on buoyancy. However, fluctuations in composition are in our setup advected together with fluctuations in entropy and the latter determines the buoyancy. It does not matter whether the entropy difference between a fluid parcel and its surroundings is a result of a difference in composition or heat content or both. This special property of our setup would be lost if we introduced a composition dependence in Eqs. (7)–(10) by using a more complex equation of state, or by including temperature- and composition-dependent terms such as heat-conduction, radiative-diffusion, or nuclear reactions.

2.3. Codes

We run simulations of the problem described above using five 3D hydrodynamic codes that are well established in the field of stellar convection: FLASH, MUSIC, PPMSTAR, PROMPTI, and SLH. All of these codes are based on the finite-volume method but there are many differences between the numerical schemes as shown in Table 2 and in the following subsections.

2.3.1. FLASH

FLASH (Fryxell et al. 2000) is a modular multidimensional hydrodynamics code that originated from the combination of the legacy PROMETHEUS code (Fryxell et al. 1989) and the AMR library PARAMESH (MacNeice et al. 2000). FLASH was originally developed to simulate Type Ia supernovae (e.g., Plewa et al. 2004) and has since been extended by a large variety of modules including magnetic fields, radiation transfer, and the consideration of a cosmological redshift. Due to its great flexibility, FLASH has since been used to address various astrophysical problems including core-collapse (e.g., Couch & Ott 2015) and type Ia (e.g., Willcox et al. 2016) supernova explosions, galaxy evolution (e.g., Scannapieco & Brügggen 2015), or interstellar turbulence (e.g., Federrath et al. 2010). For this work we used version 4.6.2 of FLASH with its default unsplit hydrodynamics solver. Compared to the default settings, we increase the order of reconstruction to the 3rd order PPM (Colella & Woodward 1984) and apply the HLLC Riemann solver. The top and bottom boundary are implemented as reflective boundaries. FLASH uses double precision (64-bit) floating-point arithmetic to perform the computations, but the output that is used for post-processing has been written in single precision (32-bit) to save disk space.

2.3.2. MUSIC

The MULTI-dimensional Stellar Implicit Code MUSIC (Viallet et al. 2016; Goffrey et al. 2017) is a time-implicit hydrodynamics code designed to study key phases of stellar evolution in two and three dimensions in spherical or Cartesian coordinates. The

Table 2. Basic characteristics of the codes used in this study.

Code	Energy equation	Grid	Reconstruction	Numerical flux function	Dimensional splitting	Time stepper	FP precision
FLASH	Eq. (9)	Collocated	PPM	HLLC Toro (2009)	No	Lee & Deane (2009)	64-bit (*)
MUSIC	Eq. (9)	Staggered	Van Leer (1974)	Upwinded advection Viallet et al. (2016)	No	Crank-Nicolson	64-bit
PPMSTAR	Eq. (9)	Collocated	PPM+PPB	Woodward (2007)	Yes	PPM (**)	32-bit
PROMPI	Eq. (9)	Collocated	PPM	Fryxell et al. (2000)	Yes	Fryxell et al. (2000)	64-bit (*)
SLH	Eq. (10)	Collocated	PPM	AUSM+-up Liou (2006)	No	RK3	64-bit

Notes. Section 2.3 contains further references and details. (*)Output is written in 32-bit precision. (**)Exact algorithm to be described in a future publication.

code solves the Euler equations, optionally supplemented with diffusive radiation transport, gravity, the Coriolis and centrifugal terms, and active and/or passive scalars. The equation set is closed using either an ideal gas or a tabulated equation of state. The equations are spatially discretised using a finite volume method, on a staggered mesh, with scalar quantities defined at cell centres, and vector components at cell faces. The advection step uses a second-order interpolation, and a gradient limiter originally described by van Leer (Van Leer 1974). Time discretisation is carried out using the Crank-Nicolson method, and a physics-based preconditioner is used to accelerate the convergence of the implicit method (Viallet et al. 2016). The boundary conditions are implemented via appropriate ghost zone layers (reflective along the top and bottom walls and periodic along the side walls), and the code uses 64-bit precision throughout.

The code has been benchmarked against a number of standard hydrodynamical test problems (e.g., Goffrey et al. 2017) and has been applied to a number of stellar physics problems, including accretion (Geroux et al. 2016) and convective overshooting (Pratt et al. 2017, 2020; Baraffe et al. 2017).

The concentration of the μ_1 fluid is advected as a passive scalar. The corresponding flux is reconstructed using the mass fractions of the μ_1 fluid. For comparison with the other codes, all output is linearly interpolated onto a cell-centred grid as a first post-processing step.

2.3.3. PPMSTAR

The explicit Cartesian compressible gas-dynamics code PPMSTAR is based on the Piecewise-Parabolic Method (PPM; Woodward & Colella 1981, 1984; Colella & Woodward 1984; Woodward 1986, 2007). In its most recent version (Woodward et al. 2019) it solves the conservation equations in a perturbation formulation with regard to an initial base state that is valid for perturbations of any size. This allows the computation to be carried out with only 32-bit precision and roughly doubles the execution speed. The time-stepping algorithm has been revised and will be described in a future publication. Another key feature of PPMSTAR is tracking the advection of the concentrations in a two-fluid scheme using the Piecewise-Parabolic Boltzmann moment-conserving advection scheme (PPB; Woodward 1986; Woodward et al. 2015). Nuclear reactions and energy production is taken into account with approximate networks (Herwig et al. 2014; Andrassy et al. 2020; Stephens et al. 2021). Both radiation pressure and diffusion can be taken into account

(Mao et al., in prep.). Reflective boundary conditions are used at the top and bottom boundaries.

2.3.4. PROMPI

PROMPI is a multidimensional hydrodynamics code based on an Eulerian implementation of the piecewise parabolic method PPM by Colella & Woodward (1984) capable of treating realistic equations of state (Colella & Glaz 1985) and multi-species advection. It is equipped with an equation of state to handle the semi-degenerate stellar plasma (Timmes & Swesty 2000), gravity, radiative diffusion and a general nuclear reaction network. PROMPI is a version of the legacy PROMETHEUS code (Fryxell et al. 1991) parallelised with MPI (Message-Passing-Interface) by Meakin & Arnett (2007). Notable scientific work enabled by PROMPI can be found in Meakin & Arnett (2007), Arnett et al. (2009), Viallet et al. (2013), Mocák et al. (2018), or Cristini et al. (2019). Latest development of PROMPI includes GPU acceleration (Hirschi, priv. comm.) and runtime calculation of space-time averaged mean-fields for extensive Reynolds-Averaged-Navier Stokes (or RANS) analysis³. PROMPI uses 64-bit precision internally but it writes output in 32-bit precision to save disk space. Reflective boundary conditions are used at the top and bottom boundaries.

2.3.5. SLH

The Seven-League Hydro (SLH) code, initially developed by Miczek (2013), solves the fully compressible Euler equations in one to three spatial dimensions. It contains a general equation of state including radiation pressure and electron degeneracy (Timmes & Swesty 2000) and supports radiative transfer in the diffusion limit. A monopole and a full 3D gravity solvers are also available. An arbitrary number of fluids can be advected and interactions between them can be simulated using a nuclear-reaction module (Edelmann 2014).

The equations are discretised on logically rectangular, but otherwise arbitrary, curvilinear grids using a finite-volume scheme. The code specialises in slow, nearly hydrostatic flows in the stellar interior. Various methods to treat the hydrostatic background stratification (Edelmann et al. 2021) are used in combination with several low-Mach flux functions (Liou 2006; Li & Gu 2008; Miczek et al. 2015) to reduce dissipation at low

³ For more details on the PROMPI's mean-fields utilisation see the ransX framework <https://github.com/mmicromegas/ransX>

Mach numbers, which is unacceptably high with standard flux functions. Reconstruction schemes available range from constant through linear with several optional slope limiters and unlimited parabolic to the PPM reconstruction of [Colella & Woodward \(1984\)](#). SLH supports both implicit and explicit time stepping. The code has been shown to scale up to several hundred thousand cores ([Edelmann et al. 2016](#); [Hammer et al. 2016](#)) and applied to problems involving mixing processes ([Edelmann et al. 2017](#); [Horst et al. 2021](#)) and wave generation ([Horst et al. 2020](#)) in stellar interiors.

In the present work, we use PPM reconstruction with a slightly modified version of the AUSM+-up flux function ([Liou 2006](#)) and the Deviation well-balancing method of [Berberich et al. \(2021\)](#). The wall boundary conditions are implemented as flux-based boundaries such that mass and energy fluxes through the walls are exactly zero. Ghost cells are used at the wall boundaries in the reconstruction process: we perform parabolic extrapolation for all conserved variables with the exception of composition variables, which are assumed to be constant at the wall boundaries. Because the flow in the test problem is relatively fast, we employ explicit time stepping with the RK3 scheme to reduce computational costs. The code uses 64-bit floating-point arithmetic. Unlike the other codes, we add another density perturbation on top of that defined by Eq. (6) in the form of white noise with an amplitude of 5×10^{-7} because SLH otherwise preserves the reflection symmetry of Eq. (6) with respect to the plane $x = -z$.

2.4. Simulations and their output

We use the same Cartesian computational grids with constant spacing in all codes: a low-resolution grid with 128^3 cells, a medium-resolution grid of 256^3 cells and a high-resolution grid of 512^3 cells. Because all quantities we compare in Sect. 3 converge rapidly upon grid refinement, we only perform a full 512^3 run with PPMSTAR and a short one with PROMPI to save computing time.

All simulations are stopped at time $t_{\text{end}} = 2 \times 10^3$, which corresponds to 25 convective turnover timescales, and we write output every 5 time units with the exception of the 512^3 PROMPI run, in which the output is written every 1.266 time units. The output includes the full 3D state information, which is post-processed to obtain 1D horizontal averages for a number of quantities, kinetic-energy spectra, and 2D slices through the simulation box, see Appendix A for details. We use both horizontal volume-weighted averages

$$\bar{q}_j = \frac{1}{N_x N_z} \sum_{i,k} q_{i,j,k}, \quad (13)$$

and horizontal mass-weighted averages

$$\tilde{q}_j = \frac{\sum_{i,k} q_{i,j,k} \rho_{i,j,k}}{\sum_{i,k} \rho_{i,j,k}}, \quad (14)$$

where q is the quantity to be averaged, ρ is the density, i , j , and k , are grid cell indices along the x , y , and z axes, respectively, and N_x and N_z are the total numbers of cells along the axis given in the subscript. The cell volume does not appear in Eqs. (13) and (14), because it is the same for all cells. We use the notation $\langle q \rangle$ for time averages. We always give the averaging time interval in the text or in the figure caption. In some cases, we need to smooth a time series $q(t)$ to suppress noise and make it easier to visually compare different simulations. To do so, a centred top-hat convolution filter is employed. Its width τ_{av} is specified in

each case individually. We suppress boundary effects by padding the time series with the time average of $q(t)$ during the first and last $\frac{\tau_{\text{av}}}{2}$ time units before performing the convolution.

3. Results

Owing to the well-resolved initial perturbation, the initial growth of convection proceeds at the same rate in all of the codes considered (see Fig. 3 and Sect. 3.1). As the animations available on Zenodo⁴ show, a substantial amount of small-scale structure appears in the large-scale hot bubbles during their initial rise from the heating layer towards the top of the convective layer. They deform the convective-stable interface significantly when they impact it at $t \approx 60$, generating the first upward-propagating internal gravity waves (IGW). The stable nature of the upper layer forces the convective upflows to decelerate and, ultimately, reverse. As this happens, the flows drag some of the μ_1 fluid into the convective layer, starting the process of mass entrainment, see Fig. 2. The flow keeps accelerating during the first few convective timescales $\tau_{\text{conv}} = 80$ (see Eq. (16)) and we conservatively define the end of this initial transient to be $t_0 = 500 \approx 6\tau_{\text{conv}}$. We focus our analysis on the remaining 1500 time units ($\approx 19\tau_{\text{conv}}$) of steady convection accompanied with continuous increase in the convective layer's mass due to mass entrainment. In the following subsections, we present different aspects of the simulations and compare their evolution in the five codes in detail.

3.1. Velocity field

We first compare the simulations in terms of horizontally averaged rms velocity fluctuations

$$\bar{v}_{\text{rms}}(y) = \left(\bar{\sigma}_{v,x}^2 + \bar{\sigma}_{v,y}^2 + \bar{\sigma}_{v,z}^2 \right)^{\frac{1}{2}}, \quad (15)$$

where $\bar{\sigma}_{v,x}$, $\bar{\sigma}_{v,y}$, and $\bar{\sigma}_{v,z}$ are standard deviations of the three components of the velocity vector. We further average the velocity profiles over the convective and stable layer to obtain (mass-weighted) bulk measures of typical velocity fluctuations in the two layers⁵. Because a substantial amount of mass gets entrained into the convective layer during the simulations, we track the vertical coordinate y_{ub} of the upper boundary of the convective layer in time as described in Sect. 3.2. We compute the averages in the regions with $y < y_{\text{ub}}(t) - 0.1$ (convective layer) and $y > y_{\text{ub}}(t) + 0.1$ (stable layer). The offsets of 0.1 length units are used to exclude the transition zone.

The time evolution of the rms velocity fluctuations in the convective and stable layer is shown in Fig. 3. During the initial transient ($t < t_0 = 500$), the rms velocity in the convective layer increases until it has saturated at a mean value of ≈ 0.034 . Because the speed of sound is approximately 0.8 in the middle of the convective layer, the typical Mach number is ≈ 0.04 . The flow speed is statistically the same in all of our simulations. This means that even in the 128^3 simulations the convection is a fully developed turbulent flow, in which kinetic energy is dissipated at a rate independent of the code-dependent numerical viscosity close to the grid scale. The 3σ fluctuations around the mean value of \bar{v}_{rms} range from 11% to 19% with a median value of 13%. The 512^3 PPMSTAR run contains two

⁴ <https://doi.org/10.5281/zenodo.5796842>

⁵ We do not use any special notation for the bulk averages. It should be clear from the text whether a bulk quantity or the vertical dependence $\bar{v}_{\text{rms}}(y)$ is being discussed.

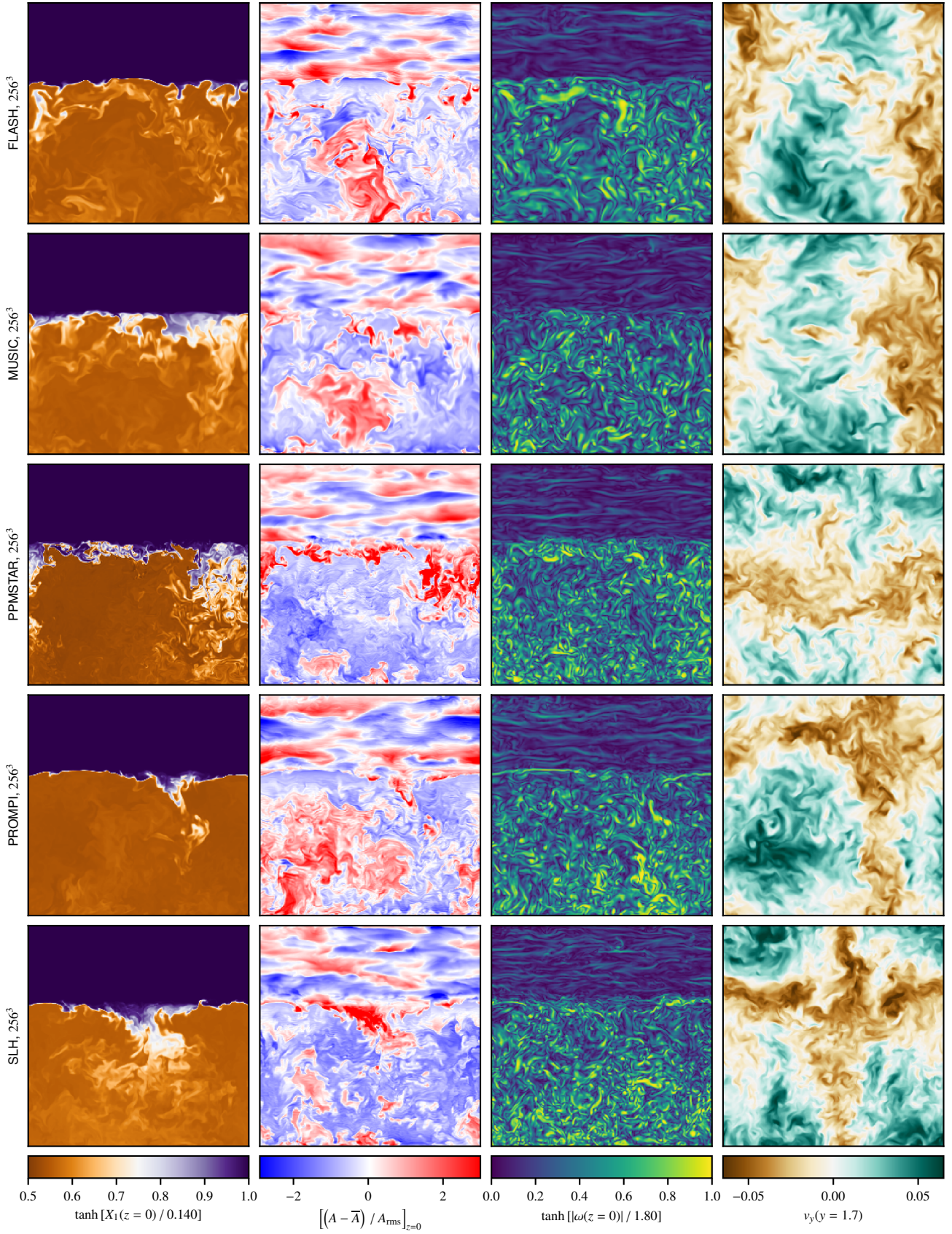


Fig. 2. Renderings of the flow field at $t = 1000$. The variables shown (from left to right) are the mass fraction X_1 , relative entropy fluctuations $(A - \bar{A})/A_{\text{rms}}$, magnitude of vorticity $|\omega|$, and the vertical component of velocity v_y . The first three variables correspond to slices through the simulations box in the $z = 0$ plane and the last one in the $y = 1.7$ plane. The non-linear scaling of X_1 and $|\omega|$ and the normalisation of entropy fluctuations by their horizontal rms spread A_{rms} (see the colour bars) are introduced for visualisation purposes.

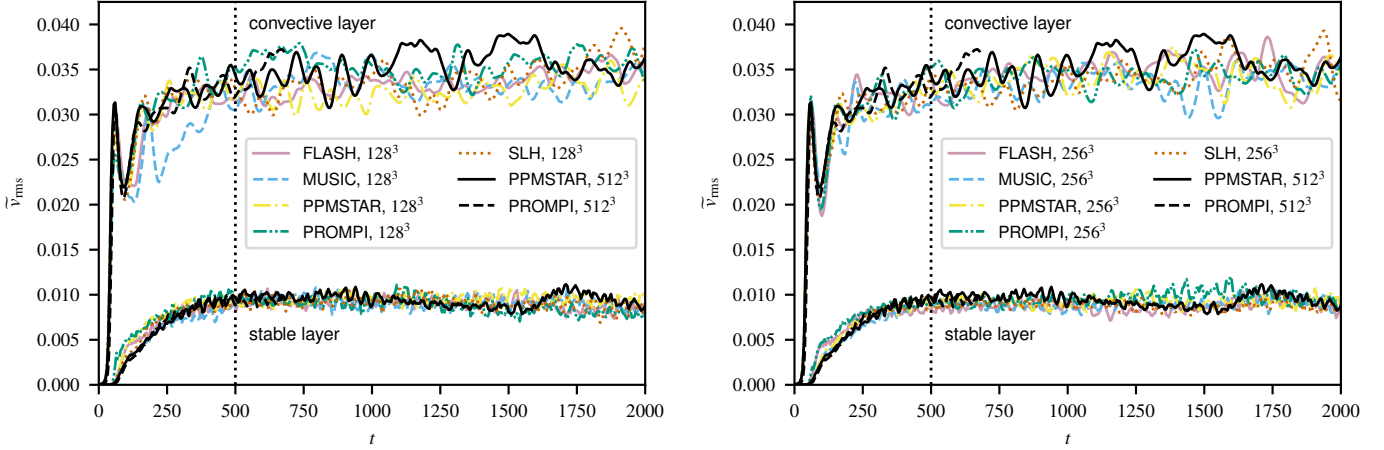


Fig. 3. Time evolution of the mass-weighted average rms velocity fluctuations in the convection zone ($y < y_{\text{ub}}(t) - 0.1$, upper set of curves) and in the stable layer ($y > y_{\text{ub}}(t) + 0.1$, lower set of curves). The location $y_{\text{ub}}(t)$ of the convective boundary is tracked in time as shown in Fig. 9. The dotted vertical line marks the end of the initial transient excluded from the analysis.

high-velocity episodes in the time intervals $1090 \lesssim t \lesssim 1250$ and $1440 \lesssim t \lesssim 1620$, both $\approx 2\tau_{\text{conv}}$ long, during which the bulk convective velocity increases by up to $\approx 15\%$ as compared with the 128^3 and 256^3 runs. These episodes are likely of statistical origin, although we would need an ensemble of 512^3 runs to confirm this hypothesis. When all of the simulations are considered, there is some weak evidence for a slight systematic increase in velocity in the interval $500 < t < 2000$ with a median value of 4%. However, run-to-run values range from -1% to 15% and seem to be dominated by statistical variation, so we do not subtract the linear trend when computing the magnitude of fluctuations.

The rms velocity in the stable layer initially increases slowly as IGW generated at the convective-stable interface propagate through the stable layer. To understand the slow vertical propagation of the waves, we refer to the 2D renderings of entropy fluctuations in Fig. 2. The renderings show a number of wavefronts spanning almost the full width of the computational domain and inclined at angles of $5^\circ \lesssim \vartheta \lesssim 15^\circ$ with respect to the horizontal, ignoring projection effects. The Brunt-Väisälä frequency N_{BV} does not change much across the stable layer and its typical value is $N_{\text{BV}} = 0.42$ at $t = 0$. Using linear theory of IGW (see e.g., Sutherland 2010), we estimate that such long waves have periods in the range $2\tau_{\text{conv}} \gtrsim P_{\text{IGW}} \gtrsim 0.7\tau_{\text{conv}}$. The magnitude of the vertical component of their group velocity is $0.001 \lesssim v_{y,\text{IGW}} \lesssim 0.01$, implying that it takes the waves between ≈ 100 and ≈ 1000 time units to propagate from the initial convective-stable interface at $y = 2$ to the upper boundary condition at $y = 3$. Projection effects, which decrease the apparent angle ϑ in the 2D slices, make this estimate slightly biased towards longer timescales. Nevertheless, Fig. 3 shows that the final velocity amplitude of ≈ 0.009 in the stable layer is reached by $t = t_0 = 500$ in all of our simulations. This observation, combined with the estimate above suggests that there cannot be strong resonant wave amplification caused by many wave reflections between the upper boundary of the simulation box and the convective-stable interface because that would cause the rms velocity to increase on much longer timescales. Finally, the fact that the rms velocity reaches the same value on grids ranging from 128^3 to 512^3 implies that the dominant wave patterns are well resolved even on the coarsest grid. This conclusion is further supported by kinetic-energy spectra, which we discuss at the end of this section.

We define the convective turnover timescale to be

$$\tau_{\text{conv}} = \frac{2\langle\Delta y_{\text{cl}}\rangle}{\langle\tilde{v}_{\text{rms}}\rangle} = 80, \quad (16)$$

where $\langle\tilde{v}_{\text{rms}}\rangle = 0.034$ is the above-mentioned characteristic convective velocity and

$$\langle\Delta y_{\text{cl}}\rangle = \langle y_{\text{ub}}\rangle - y_{\text{bot}} = 1.36 \quad (17)$$

is the average depth of the convective layer, that is the distance between the layer's bottom at $y_{\text{bot}} = 1$ and the vertical coordinate y_{ub} of the upper convective boundary averaged in the time interval $500 \leq t \leq 2000$, see Fig. 9 and Sect. 3.2 for details. To show how well this timescale describes variability in the global convective velocity field, we compute the autocorrelation function

$$R(\Delta t) = \frac{\int_{t_0}^{t_{\text{end}}} \tilde{v}_{\text{rms}}^*(t) \tilde{v}_{\text{rms}}^*(t + \Delta t) dt}{\int_{t_0}^{t_{\text{end}}} \tilde{v}_{\text{rms}}^*(t)^2 dt}, \quad (18)$$

where Δt is a time shift and $\tilde{v}_{\text{rms}}^*(t)$ is constructed from the bulk convective velocity $\tilde{v}_{\text{rms}}(t)$ as follows: (1) the initial transient ($t < t_0$) is discarded, (2) the best-fit linear trend is subtracted to suppress spurious correlations caused by any slight systematic changes in $\tilde{v}_{\text{rms}}(t)$ on long timescales, and (3) the resulting time series is made periodic by appending to it a time-reversed version of itself. Figure 4 shows that $R(\Delta t)$ reaches high values for $\Delta t \lesssim 0.5\tau_{\text{conv}}$, implying that the time series is strongly correlated on such short timescales. However, $R(\Delta t)$ decreases steeply and, although the function's first zero crossing occurs at $\Delta t < \tau_{\text{conv}}$ in some runs and at $\Delta t > \tau_{\text{conv}}$ in others due to stochasticity, $\Delta t \approx \tau_{\text{conv}}$ is generally a good estimate of the temporal spacing between different, largely independent, flow realisations.

Figure 5 shows time-averaged profiles of rms vertical and horizontal velocity fluctuations,

$$\tilde{v}_{y,\text{rms}}(y) = \tilde{\sigma}_{v,y}, \quad (19)$$

$$\tilde{v}_{xz,\text{rms}}(y) = \left(\tilde{\sigma}_{v,x}^2 + \tilde{\sigma}_{v,z}^2\right)^{\frac{1}{2}}. \quad (20)$$

The vertical fluctuations $\langle\tilde{v}_{y,\text{rms}}\rangle$ reach a broad maximum close to the middle of the convective layer. They vanish at the bottom of the simulations box as required by the wall boundary conditions but they also drop by as much as a factor of four at

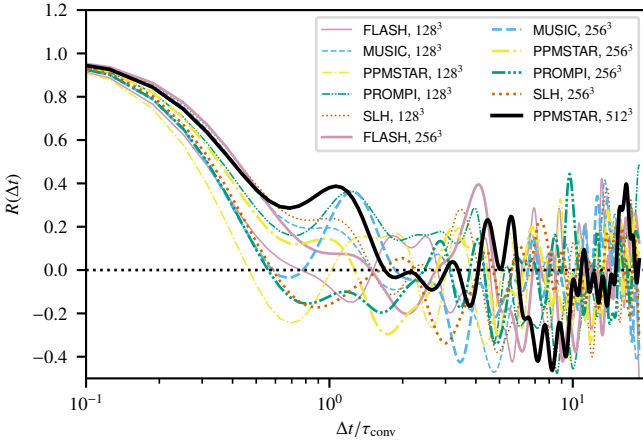


Fig. 4. Autocorrelation function $R(\Delta t)$ of the convective velocity as a function of the time shift Δt in units of the convective turnover timescale τ_{conv} . See Eq. (18) and the associated text in detail.

the transition to the stable layer. The velocity field is dominated by the horizontal component $\langle \bar{v}_{xz,\text{rms}} \rangle$ close to the boundaries of the convective layer, where the flow has to turn around, and in the stable layer filled with internal gravity waves. The velocity profiles are nearly constant in the bulk of the stable layer with $\langle \bar{v}_{xz,\text{rms}} \rangle / \langle \bar{v}_{y,\text{rms}} \rangle \approx 2$ and only a mild increase in velocity towards lower densities. There is another drop in $\langle \bar{v}_{y,\text{rms}} \rangle$ at $y \gtrsim 2.9$ as gravity is turned off (Eq. (2)), removing the waves' restoring force, and the upper wall boundary condition forces $\langle \bar{v}_{y,\text{rms}} \rangle$ to vanish at $y = 3$.

It is clear from Fig. 5 that all five codes produce similar time-averaged velocity profiles. However, we have to take the stochastic character of turbulent convection into account to see whether the remaining differences are significant. Instead of running ensembles of simulations with randomised initial conditions, which would be rather expensive, we obtain the $\pm 3\sigma$ statistical-variation bands shown in Fig. 5 as follows. The central curve of each band corresponds to the arithmetic average of all velocity profiles available on a given computational grid. We also compute the standard deviations of each time series in the averaging time window at each height y and we average the standard deviation profiles over the same set of runs to obtain one standard deviation profile $\sigma_0(y)$. The profile $\sigma_0(y)$ is our best estimate of the statistical variation to be expected in any of the runs, provided that they are statistically similar, and it does not depend on any small systematic differences between the velocity amplitudes predicted by different codes. Obviously, the statistical variation associated with the time averages should decrease as the length of the averaging interval is increased. We show above that there is approximately one independent realisation of the convective flow per turnover timescale τ_{conv} , so we estimate the statistical variation associated with the time-averaged profiles to be $\sigma(y) = \sigma_0(y) / \sqrt{N_{\text{conv}}}$, where N_{conv} is the length of the averaging time window in units of τ_{conv} .

However, we must keep in mind that $\sigma(y)$ is just an estimate, which involves both statistical and systematic uncertainties. It depends on our assumption that independent flow realisations are spaced by τ_{conv} in time. Figure 4 shows that the spacing could also be estimated to be $0.5\tau_{\text{conv}}$ or $2\tau_{\text{conv}}$, depending on which set of runs we use and on the very definition of 'decorrelation'. Moreover, the autocorrelation function $R(\Delta t)$ is based on the time evolution of the bulk convective velocity. Different parts of the simulation domain may have different characteristic timescales

and our use of τ_{conv} everywhere may bias the estimate of $\sigma(y)$. We should thus use the statistical-variation bands as a general guideline in quantifying differences between simulations but this simple approach does not allow us to calculate the probability that a deviation of a given magnitude is observed under the null hypothesis that the codes do not differ.

Figure 5 shows that the profiles of both velocity components in the 128^3 and 256^3 runs fall within or close to the respective estimated $\pm 3\sigma$ statistical-variation bands. This means that the small code-to-code differences are dominated by stochasticity. The bands of both the 128^3 and 256^3 runs are slightly below the velocity curves of the 512^3 PPMSTAR run in the convective layer, suggesting that velocity profiles may not be fully converged on the 256^3 grid. However, the time-averaging window overlaps with the two above-mentioned episodes visible in Fig. 3, during each of which the 512^3 PPMSTAR run reaches above-average bulk velocities for as much as $\approx 2\tau_{\text{conv}}$. Given the uncertainty in determining the width of the statistical-variation bands and the fact that we only have a single full-length 512^3 run⁶, we do not find the tension between the velocity profiles significant.

The arguments leading to our scaling the width of the statistical-variation bands with $1/\sqrt{N_{\text{conv}}}$ do not apply to the stable layer directly because the response timescale of the stable layer is longer than τ_{conv} (see above). Nevertheless, Fig. 5 shows that the statistical-variation bands describe the range of code-to-code variation rather well. The velocity profiles of all 128^3 and 256^3 runs closely match that of the 512^3 PPMSTAR run both in shape and amplitude, which further supports our conclusion that the dominant wave patterns are essentially converged already on the 128^3 grid.

The renderings of the vertical component of velocity v_y in the horizontal midplane of the convective layer presented in Fig. 2 show that the convective flow is dominated by the largest possible scales with wavelengths equal to the width of the computational box. This corresponds to convective cells with an aspect ratio of about unity early in the simulations, which later become slightly elongated in the vertical direction as the convective boundary moves upwards. The cells are turbulent on smaller scales with large amounts of small-scale vorticity ω , also shown in Fig. 2 and in the animations available on Zenodo⁷. We compute spatial Fourier spectra of the velocity vector,

$$\Psi_{jl} = \frac{1}{N_x N_z} \sum_{m=0}^{N_x-1} \sum_{n=0}^{N_z-1} v_{mn} \exp \left\{ -2\pi i \left(\frac{mj}{N_x} + \frac{nl}{N_z} \right) \right\}, \quad (21)$$

$$j = 0, \dots, N_x - 1, \quad (22)$$

$$l = 0, \dots, N_z - 1, \quad (23)$$

where N_x and N_z are the total numbers of computational cells along the x and z axes, respectively. The velocity array v_{mn} corresponds to a horizontal slice through the simulation box at $y = 1.7$, which is close to the midplane of the convective layer at $t \approx 1250$. We compare the kinetic energy per unit mass $\frac{1}{2} \|\Psi\|^2$ binned over all wavenumbers $k = (k_x^2 + k_z^2)^{1/2}$, where

$$k_x = \begin{cases} j, & 0 \leq j < \frac{N_x}{2} - 1, \\ -N_x + j, & \frac{N_x}{2} \leq j < N_x, \end{cases} \quad (24)$$

$$k_z = \begin{cases} l, & 0 \leq l < \frac{N_z}{2} - 1, \\ -N_z + l, & \frac{N_z}{2} \leq l < N_z. \end{cases} \quad (25)$$

⁶ The 512^3 PROMPI run is too short to be included in this comparison, see Fig. 3.

⁷ <https://doi.org/10.5281/zenodo.5796842>

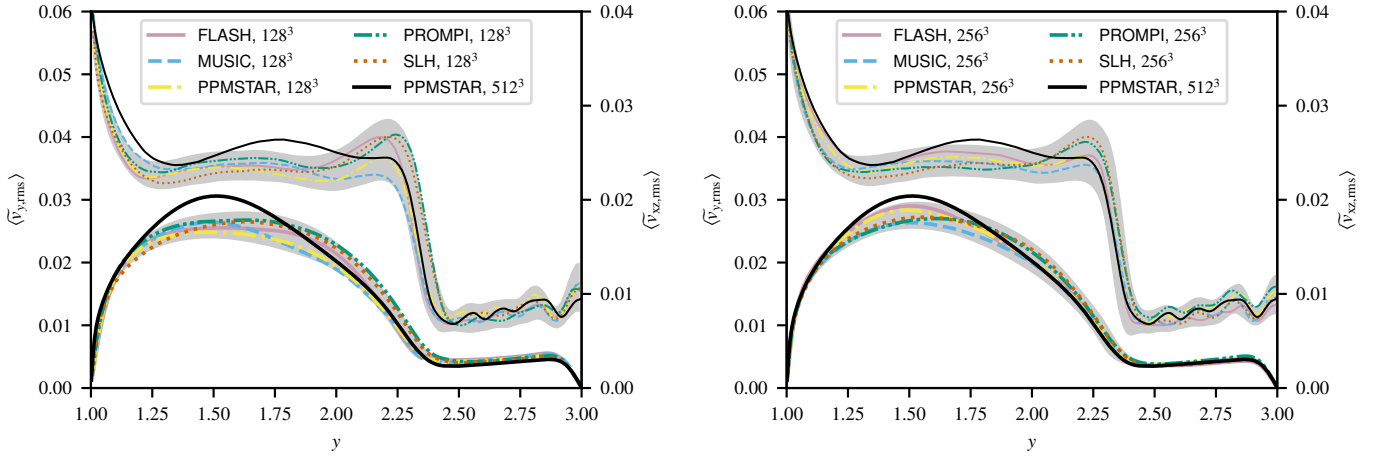


Fig. 5. Vertical profiles of the rms velocity components $\langle \overline{v_{y,rms}} \rangle$ (thick lines) and $\langle \overline{v_{xz,rms}} \rangle$ (thin lines) in the vertical and horizontal directions, respectively, averaged in a time interval $\tau_{av} = 6\tau_{conv} = 480$ wide and centred on time 1250. The two velocity axes in each plot have different scales to avoid the curves' overlapping. The grey bands give an estimate of $\pm 3\sigma$ statistical variation in the averages due to stochasticity, see Sect. 3.1 for details.

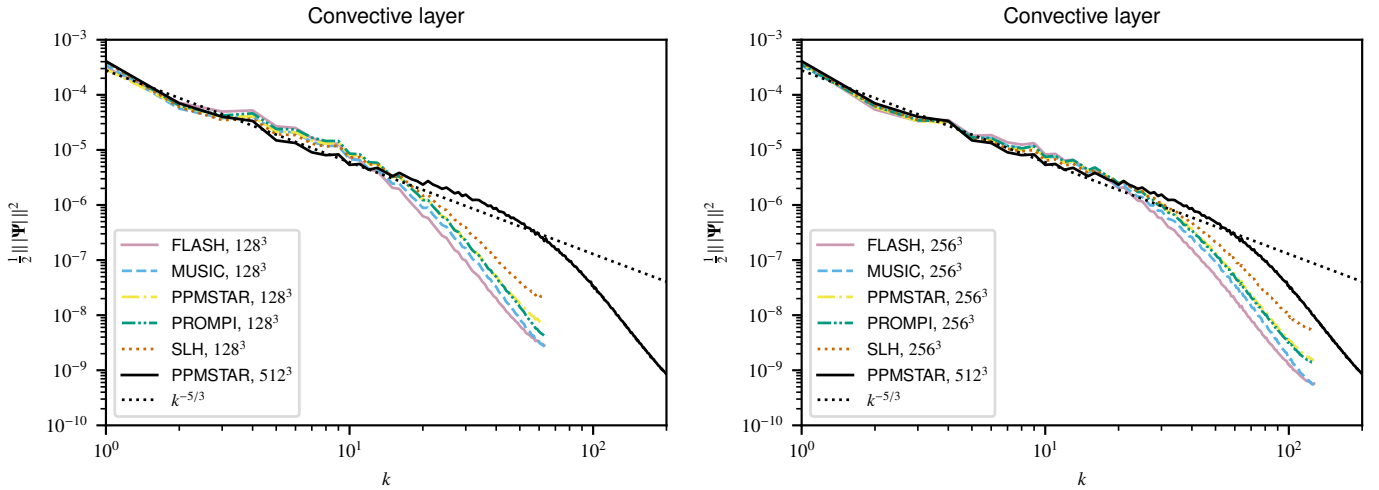


Fig. 6. Spectra of kinetic energy as functions of the wavenumber k in a horizontal slice through the convective layer at $y = 1.7$. The spectra have been averaged over the whole analysis time interval ($500 \leq t \leq 2000$, approximately $19\tau_{conv}$). The Kolmogorov scaling $k^{-5/3}$ is shown for comparison.

These expressions hold for even values of N_x and N_z , which is the case for all of our computational grids. Figure 6 shows the spectra averaged in the whole time interval of analysis ($500 \leq t \leq 2000$). Although the turbulence is anisotropic at small wavenumbers (large scales), it looks close to being isotropic at the larger wavenumbers (smaller scales) that we see in the renderings of vorticity in Fig. 2. All of the codes converge to the same kinetic energy spectrum upon grid refinement, which is consistent with Kolmogorov's $-5/3$ law. Although we only use the k_x and k_z components of the wavenumber vector for simplicity, the kinetic-energy spectrum of isotropic turbulence should have the same slope along all three axes of the wavenumber space in the inertial range. If this was not the case there would be more power along one or two of the axes on small scales in conflict with the assumption of isotropy.

The 512^3 PPMSTAR run illustrates that a rather fine grid is needed to obtain a wide and well-converged inertial range. This is due to the well-known bottleneck effect – a power excess observed in numerical simulations of turbulence between the inertial and dissipation ranges (e.g., Falkovich 1994; Sytine et al. 2000; Dobler et al. 2003). All of our spectra have similar shapes

even in the dissipation range, although they diverge with increasing k and they reach a spread of as much as a factor of 10 at the Nyquist frequency. The spectrum in the dissipation range depends on the behaviour of the numerical scheme close to the grid scale. However, this dependence is largely irrelevant thanks to the fact that the dissipation rate becomes independent of the magnitude of small-scale viscosity (be it of physical or numerical origin) in turbulent flows as long as the viscosity is small enough.

We use the same procedure to characterise the spatial spectra at $y = 2.7$ in the stable layer, see Fig. 7. These spectra are also dominated by the largest scales, which can also be seen in the 2D renderings of entropy⁸ fluctuations in Fig. 2. Although we do not have any analytic prediction of the wave spectrum, all five codes predict essentially the same spectrum at $k \lesssim 5$ on the 128^3 grid and at $k \lesssim 10$ on the 256^3 grid, respectively. This corresponds to horizontal wavelengths $\gtrsim 26$ computational cells in both cases. Although this seems to be a large number, the actual challenge is to resolve the vertical wavelength of the IGW,

⁸ We refer to the pseudo-entropy $A = p/\rho'$ as 'entropy' for simplicity.

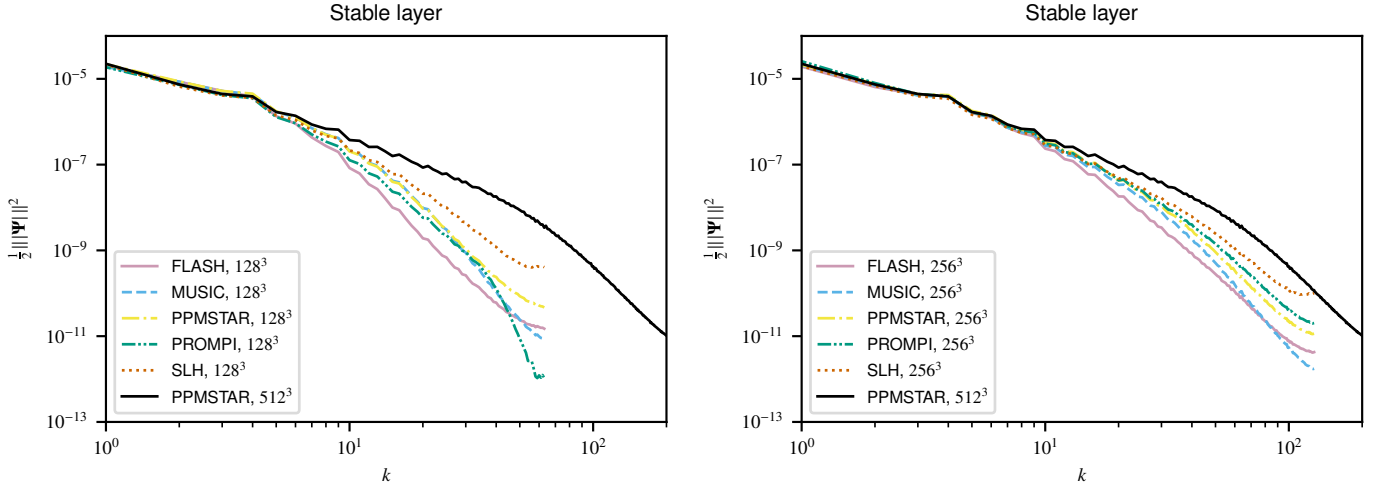


Fig. 7. Spectra of kinetic energy as functions of the wavenumber k in a horizontal slice through the stable layer at $y = 2.7$. The spectra have been averaged over the whole analysis time interval ($500 \leq t \leq 2000$, approximately $19\tau_{\text{conv}}$).

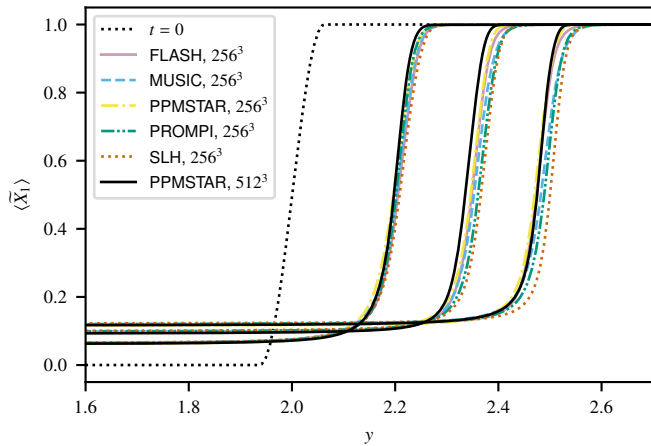


Fig. 8. Mass entrainment process visualised using vertical profiles of the mass fraction of the μ_1 fluid. Only the uppermost parts of the convective layer and part of the stable layer are shown. The dotted line shows the initial condition and the remaining three sets of lines correspond to averages over time windows $\tau_{\text{av}} = 2\tau_{\text{conv}} = 160$ wide and centred on times of (from left to right) 580, 1250, and 1920.

which is several times shorter as the waves are nearly horizontal. The most extreme of these are revealed in the 2D renderings of vorticity ω in Fig. 2, which put more emphasis on shorter vertical wavelengths as compared with the renderings of entropy fluctuations.

3.2. Convective boundary and mass entrainment

When upflows reach the upper boundary of the convective layer, they stir and entrain some of the μ_1 fluid in the transition zone and carry it down into the convective layer as they turn around⁹. The distribution of this fluid's mass fraction X_1 , shown in Fig. 2, provides a good visual representation of the entrainment process. The process also involves the mixing of entropy, which flattens the entropy gradient in the transition zone and makes the convective layer grow. The gradual growth can be seen in the profiles of the mass fraction X_1 shown in Fig. 8.

⁹ Woodward et al. (2015) provide a detailed description of this process.

We define the vertical coordinate y_{ub} of the convective boundary to be where $\partial_y \tilde{X}_1$ reaches its global maximum. Discretisation noise is reduced by fitting a parabola to the three data points closest to the discrete estimate of the maximum's vertical coordinate. Figure 9 shows that the 128^3 runs diverge in $y_{\text{ub}}(t)$ slightly with FLASH and PROMPI predicting the slowest and fastest boundary motion, respectively. However, the relative difference in the distance travelled by the end of the simulations, $\Delta y_{\text{ub}}(t_{\text{end}}) = y_{\text{ub}}(t_{\text{end}}) - y_{\text{ub}}(0)$, is only $\approx 14\%$ between these two extremes. The spread is reduced by another factor of ≈ 3 on the 256^3 grid, on which all five codes agree on $\Delta y_{\text{ub}}(t_{\text{end}})$ within 5%. Moreover, the $y_{\text{ub}}(t)$ curves derived from the 256^3 runs closely track those from the 512^3 PROMPI and PPMSTAR runs.

We characterise the thickness of the convective boundary using the scale height

$$H_{\text{ub}} = \left[\tilde{X}_1 \left(\frac{\partial \tilde{X}_1}{\partial y} \right)^{-1} \right]_{y=y_{\text{ub}}} . \quad (26)$$

Because this quantity is highly variable on short timescales, we smooth the time series using a top-hat filter $\tau_{\text{av}} = \tau_{\text{conv}} = 80$ wide and show the result in Fig. 10. In all five codes, the boundary is up to 50% thicker on the 128^3 grid than on the 256^3 grid. However, differences between the 256^3 and 512^3 runs are substantially smaller, suggesting that H_{ub} is close to being converged on the 256^3 grid at a value of ≈ 0.04 . The converged value is close to the initial value $H_{\text{ub}}(t=0) = 0.0404$. This fact, however, is purely coincidental. At $t=0$, H_{ub} characterises the steepness of the 1D transition zone as we define it. Once convection has started, H_{ub} is a product of spatial averaging along a 3D convective boundary that is as sharp as the numerical scheme allows at some places but much wider at other places, see Fig. 2. The converged thickness $H_{\text{ub}} \approx 0.04$ corresponds to 10 computational cells on the 512^3 grid, although the PPB advection method implemented in PPMSTAR can preserve gradients spanning only about two computational cells (Woodward et al. 2015). This suggests that the 3D deformation of the boundary dominates the thickness of the spatially averaged boundary. However, it is much more challenging for the codes to resolve the physical thickness of the boundary on the 256^3 and 128^3 grids, on which 0.04 units of length correspond to only about 5 and 2.5 computational cells, respectively.

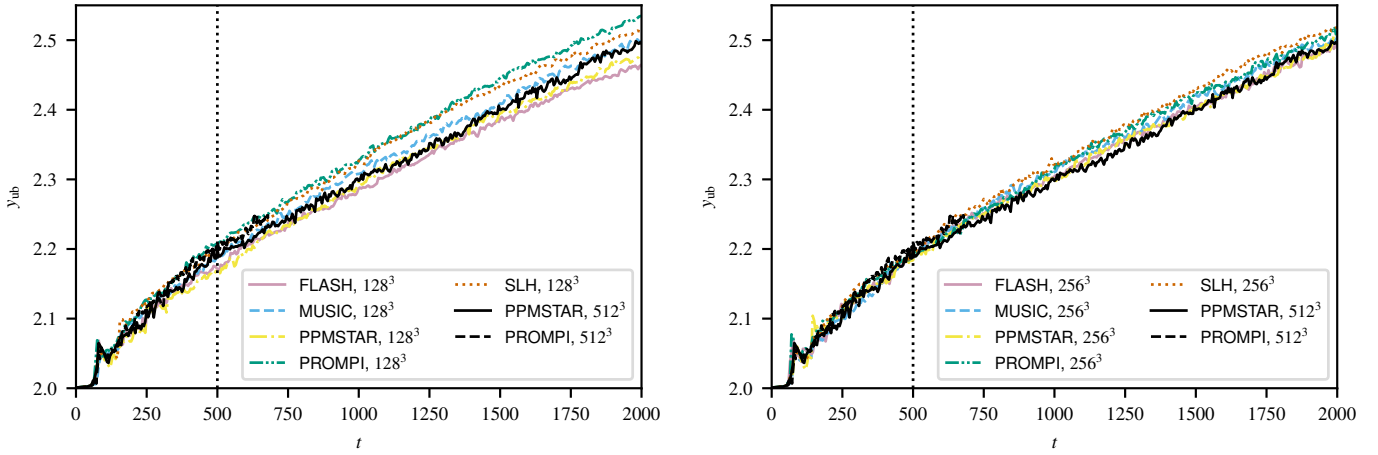


Fig. 9. Time evolution of the y coordinate of the upper boundary of the convection zone as determined from 1D averages. The dotted vertical line marks the end of the initial transient excluded from the analysis.

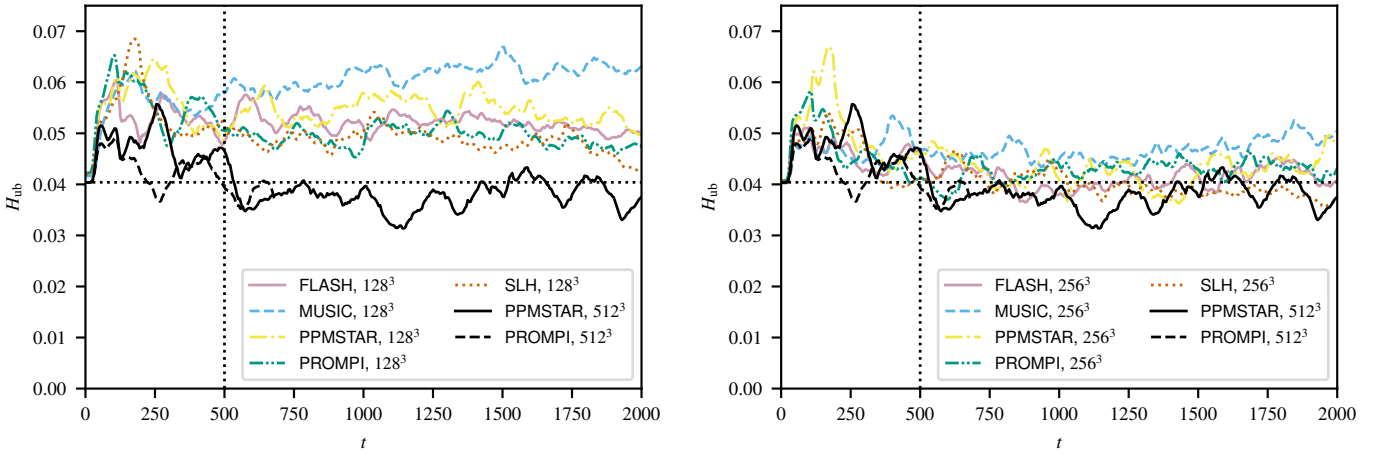


Fig. 10. Time evolution of the thickness H_{ub} (Eq. (26)) of the upper convective boundary. The time series have been smoothed using convolution with a top-hat kernel $\tau_{\text{av}} = \tau_{\text{conv}} = 80$ wide. The dotted vertical line marks the end of the initial transient excluded from the analysis. The dotted horizontal line shows $H_{\text{ub}}(t = 0) = 0.0404$ for comparison.

The total amount of the μ_1 fluid entrained by time t is

$$M_e(t) = \int_1^{y_{\text{ub}}(t)} \bar{X}_1(y, t) \bar{\rho}(y, t) dy \quad (27)$$

per unit of horizontal surface area and is shown in Fig. 11. This entrainment metric differs slightly from $y_{\text{ub}}(t)$ because of the density stratification. The agreement between the codes' predictions is slightly better as compared with the $y_{\text{ub}}(t)$ metric and the 128^3 and 256^3 simulations agree within 9% and 4%, respectively, on the total amount of the μ_1 fluid entrained by the end of the simulations. All of the initial transition layer is entrained early on during the initial transient as indicated in Fig. 11.

It is also clear from Fig. 11 that the mass entrainment rate $\dot{M}_e(t)$ is relatively high early on and slowly decreases throughout the simulation time. We compute $\dot{M}_e(t)$ using second-order central differences. The differencing greatly amplifies noise, which we suppress using convolution with a centred top-hat kernel $\tau_{\text{av}} = 3\tau_{\text{conv}} = 240$ wide. The mass entrainment rates, shown in Fig. 12, agree between all of the codes within their statistical fluctuations already on the 128^3 grid and they remain unchanged as the grid is refined up to 512^3 . However, $\dot{M}_e(t)$ varies randomly on timescales as long as several convective turnover timescales, which likely contributes to the small spread in $M_e(t)$ in Fig. 11. The entrainment rate decreases from $\approx 1.0 \times 10^{-4}$ at the beginning

of the initial transient to $\approx 4.5 \times 10^{-5}$ at $t = t_0$ and to $\approx 2.0 \times 10^{-5}$ at $t = t_{\text{end}}$. Part of the decrease may be attributed to the density at the convective boundary, which decreases from 0.28 at $t = 0$ to 0.15 at $t = t_{\text{end}}$.

One may find it surprising that the mass entrainment rates obtained on the coarse 128^3 grid agree so well with one another. Perhaps even more surprisingly, they also agree with the rates obtained in the 512^3 PROMPI and PPMSTAR runs. Convective mass entrainment is usually thought of as a complex process sensitive to small-scale flows and instabilities in the boundary layer. However, [Spruit \(2015\)](#) argues that the mass entrainment rate is set by the amount of energy available to overcome the buoyancy of the fluid being entrained; this was also observed in a laboratory study by [Linden \(1975\)](#). Applying this constraint in models of stars with helium-burning convective cores improves their agreement with both asteroseismology and population studies of globular clusters ([Constantino et al. 2017](#)). The entrainment rate is then proportional to the luminosity L driving the convective flow as confirmed by the 3D simulations of [Jones et al. \(2017\)](#) and [Andrassy et al. \(2020\)](#) of oxygen burning in a setup similar to our present test setup. More evidence comes from calibrations of the entrainment law $\dot{M}_e \propto v_{\text{rms}} \text{Ri}_B^{-n}$, where v_{rms} is the rms velocity of convection and Ri_B is the bulk Richardson number proportional to v_{rms}^{-2} for a given convective boundary (for

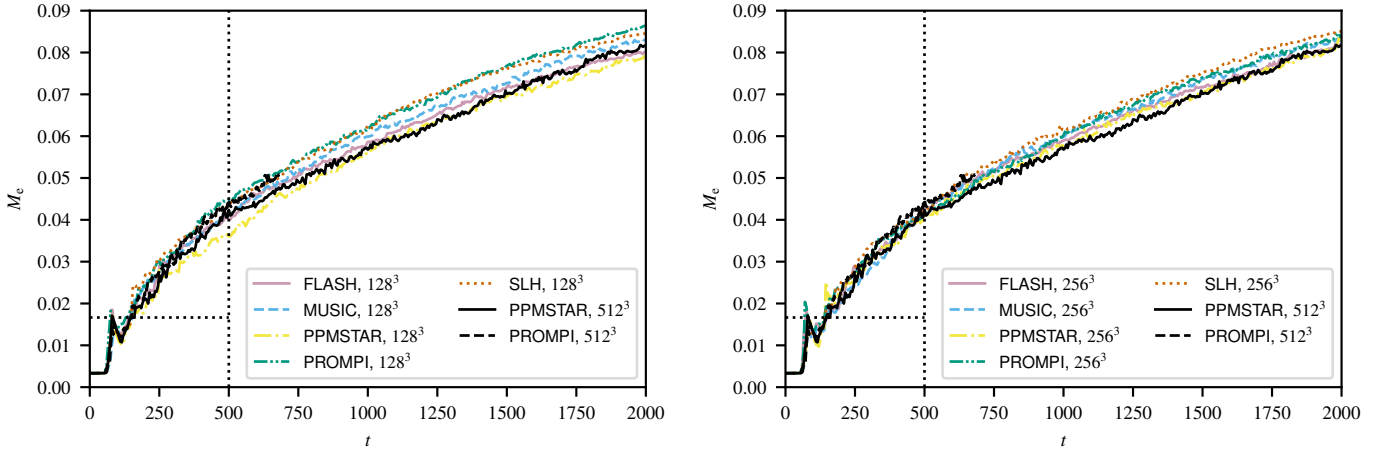


Fig. 11. Mass per unit horizontal area of the μ_1 fluid entrained into the convective layer as a function of time. The dotted vertical line marks the end of the initial transient excluded from the analysis. The dotted horizontal line shows the amount of the μ_1 fluid contained in the initial transition zone between the convective and stable layer.

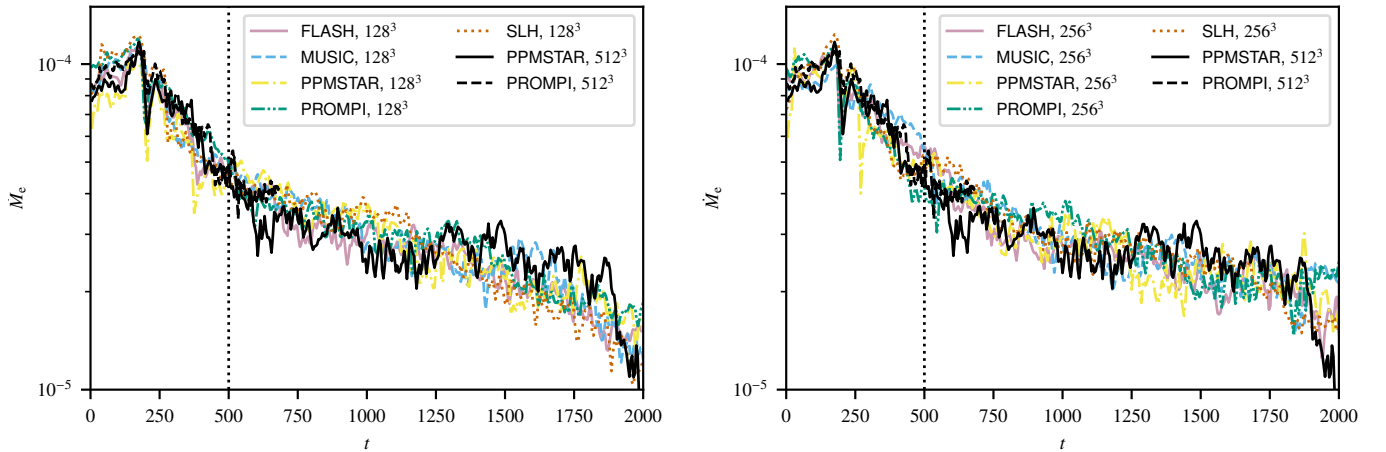


Fig. 12. Mass entrainment rates corresponding to the curves shown in Fig. 11. The time series have been smoothed using convolution with a top-hat kernel $\tau_{\text{av}} = 3\tau_{\text{conv}} = 240$ wide. The dotted vertical line marks the end of the initial transient excluded from the analysis.

a complete definition, see Meakin & Arnett 2007). Assuming that $v_{\text{rms}} \propto L^{1/3}$, we have $\dot{M}_e \propto L^{(1+2n)/3}$ and $n = 1$ corresponds to $\dot{M}_e \propto L$. Values of n measured in different numerical simulations range from 0.74 ± 0.04 (Cristini et al. 2019) and 0.74 ± 0.01 (Horst et al. 2021) through 1.05 ± 0.21 (Meakin & Arnett 2007) to 1.32 ± 0.79 (Higl et al. 2021). If buoyancy is the dominant factor one can expect to obtain a good estimate of the entrainment rate as soon as the largest downflows are reasonably resolved.

Finally, the very fact that we keep increasing the mean entropy of the convective layer by heating it at the bottom contributes to mass entrainment. This process, previously mentioned by Meakin & Arnett (2007), Andrassy et al. (2020), and Horst et al. (2021), can be understood as follows. The convective layer is well mixed and its entropy is essentially constant in space. On the other hand, entropy increases with height in the stable layer (Fig. 1). If the mean entropy of the convective layer increases it becomes higher than that of a thin layer at the very bottom of the stable layer. This thin layer is thus negatively buoyant and it must sink and mix with the convective layer. In our simulations, we compute the rate of change of the mean entropy in the lower $2/3$ of the convective layer to eliminate any influence of the convective boundary. The value of dA/dt reaches 4.7×10^{-5} by $t \approx 150$ in our 256^3 runs and it remains statistically constant until the end of the simulations. The entrainment rate due to the pro-

cess described above is $\dot{M}_{e,\text{heating}} = (dA/dt) / (dA/dm_1)$, where m_1 is the cumulative mass of the μ_1 fluid integrated from the bottom of the simulation box upwards. We measure the entropy gradient dA/dm_1 in the initial stratification. Its value is 1.6 at the top of the initial transition layer between the two fluids and 3.8 where $m_1 = M_e(t = t_{\text{end}}) \doteq 0.083$ (see Fig. 11). This implies that $\dot{M}_{e,\text{heating}} = 2.9 \times 10^{-5}$ early on during the initial transient, which is $\approx 30\%$ of the entrainment rate \dot{M}_e measured. This contribution increases to as much as $\approx 60\%$ by the end of the simulations.

3.3. Fluctuations and transport of energy

The rates of convective transport of energy and chemical composition scale with the flow speed and with the magnitude of fluctuations in the flow field. We have already shown that the flow speed is code-independent, see Sect. 3.1. Figure 13 shows that the same holds for the time-averaged relative fluctuations in entropy $\langle \Delta \bar{A} / \bar{A} \rangle$ and in the mass fraction of the μ_1 fluid $\langle \Delta \bar{X}_1 / \bar{X}_1 \rangle$. The figure includes statistical-variation bands computed as described in Sect. 3.1. The profiles of the fluctuations as produced by all of the five codes fall within or close to the corresponding statistical-variation band. The profiles of $\langle \Delta \bar{A} / \bar{A} \rangle$ agree better on the 256^3 grid in the bulk of the convection zone,

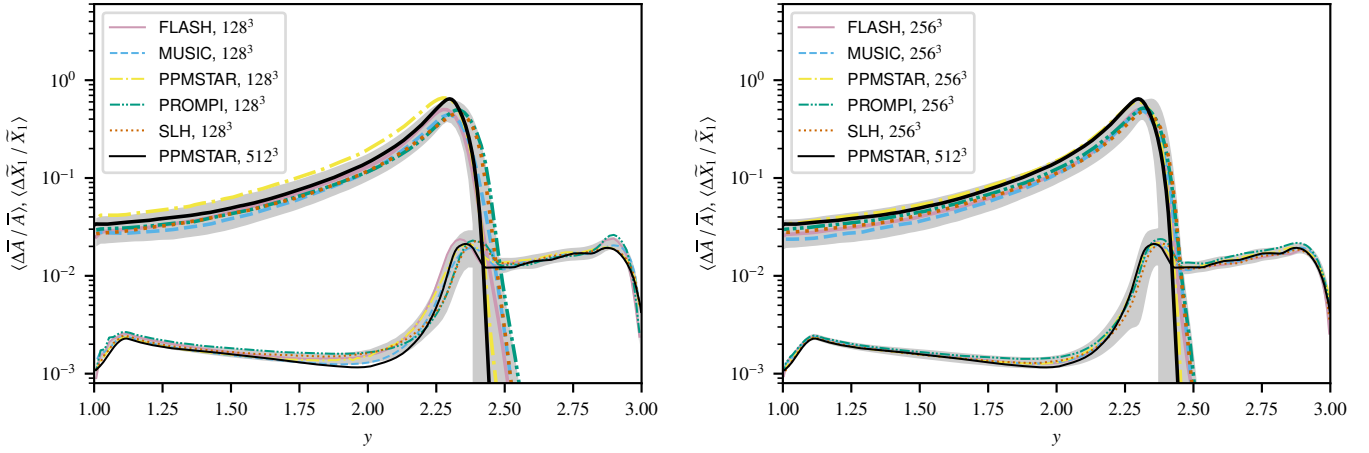


Fig. 13. Relative fluctuations in entropy \bar{A} (thin lines) and mass fraction \bar{X}_1 of the μ_1 fluid (thick lines) averaged in a time interval $\tau_{\text{av}} = 6\tau_{\text{conv}} = 480$ wide and centred on time 1250. The grey bands give an estimate of $\pm 3\sigma$ statistical variation in the averages due to stochasticity, see Sect. 3.1 for details.

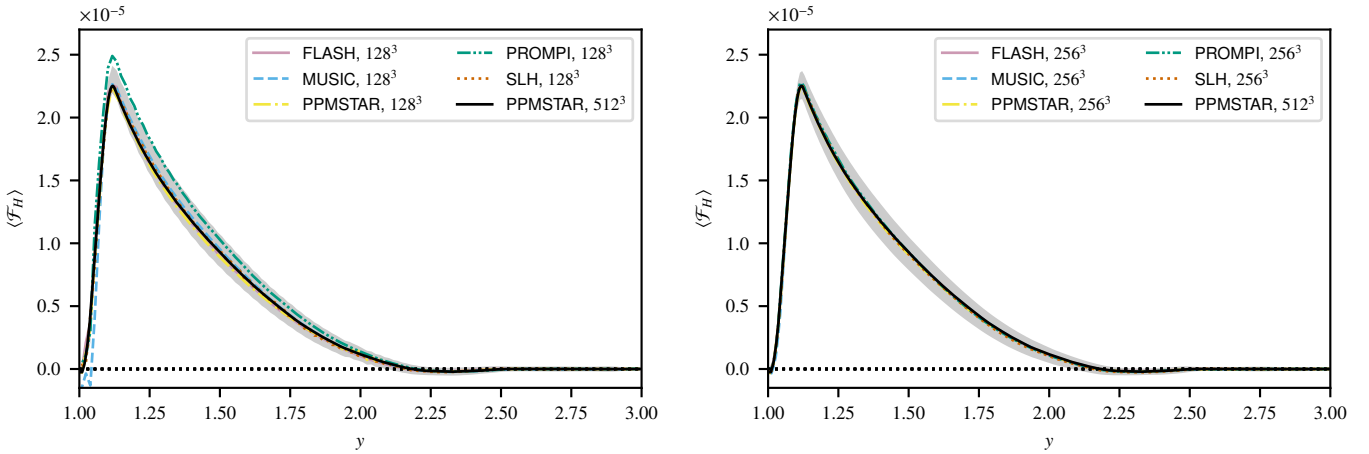


Fig. 14. Vertical profiles of the enthalpy flux averaged over the whole analysis time interval ($500 \leq t \leq 2000$, approximately $19\tau_{\text{conv}}$). The grey bands give an estimate of $\pm 3\sigma$ statistical variation in the averages due to stochasticity, see Sect. 3.1 for details.

although the differences are small already on the 128^3 grid. The slight code-to-code differences in the time-averaged position of the convective boundary (where $\langle \Delta \bar{X}_1 / \bar{X}_1 \rangle$ drops) decrease as the mass entrainment rate converges upon grid refinement, see Sect. 3.2.

We define the flux of enthalpy to be the quantity

$$\mathcal{F}_H = \overline{Hv_y} - \bar{H}\bar{v}_y, \quad (28)$$

where H is enthalpy per unit volume and v_y is the vertical component of velocity. The second term in Eq. (28) is included to remove the flux contribution of the thermal expansion and compression of the horizontally averaged stratification. We average the enthalpy flux over all of the simulation time after the initial transient ($500 \leq t \leq 2000$), ignoring the upward propagation of the convective boundary and focusing on the bulk of the convective layer instead. The profiles of $\mathcal{F}_H(y)$ shown in Fig. 14 agree well within the statistical-variation bands computed as described in Sect. 3.1, although our method seems to overestimate the statistical variation in this quantity as it is significantly larger than the code-to-code differences.

To explain this, we compute the autocorrelation function $R_{\mathcal{F}_H}(\Delta t)$ of $\mathcal{F}_H(y = 1.5)$ using the method described in Sect. 3.1. The shape of $R_{\mathcal{F}_H}(\Delta t)$ turns out to be rather different from the

autocorrelation function $R(\Delta t)$ of the bulk convective velocity; compare Fig. 15 with Fig. 4. First, the autocorrelation drops to zero on a timescale as short as $\approx 0.3\tau_{\text{conv}}$. This suggests that the width of the statistical variation bands should be reduced by the factor $0.3^{-\frac{1}{2}} = 1.8$, see Sect. 3.1. Moreover, all of our runs show some anticorrelation for $0.3\tau_{\text{conv}} \lesssim \Delta t \lesssim \tau_{\text{conv}}$ before the values of $R_{\mathcal{F}_H}(\Delta t)$ start to oscillate around zero for longer time shifts. The anticorrelation suppresses fluctuations in long-term time averages. From the physical point of view, the anticorrelation may reflect the fact that changes to the heat flux divergence result in local heating or cooling of the nearly isentropic stratification and the convective instability provides strong negative feedback, quickly driving the flux profile back to its quasi-stationary shape. In light of this, the $\approx 10\%$ overestimation of $\mathcal{F}_H(y)$ in the lower convection zone in the 128^3 PROMPI run may be statistically significant. However, the 256^3 PROMPI run agrees well with other codes run on the same grid as well as with the 512^3 PPMSTAR run. The small differences between the codes as well as the possible feedback mechanism suggest that \mathcal{F}_H is a poor indicator of code or simulation quality.

The flux of kinetic energy is defined in a way analogous to Eq. (28),

$$\mathcal{F}_k = \frac{1}{2} (\overline{\rho |v|^2 v_y} - \bar{\rho} \overline{|v|^2} \bar{v}_y). \quad (29)$$

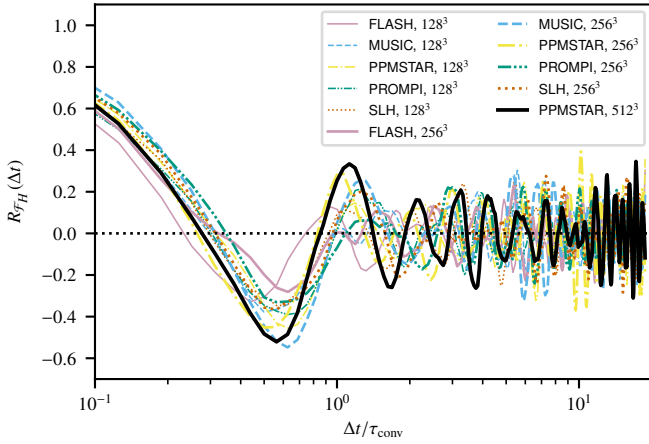


Fig. 15. Autocorrelation function $R_{\mathcal{F}_H}(\Delta t)$ of $\mathcal{F}_H(y=1.5)$ as a function of the time shift Δt in units of the convective turnover timescale τ_{conv} .

Figure 16 shows that the amplitude of \mathcal{F}_k is as much as ≈ 30 times smaller than that of \mathcal{F}_H in our simulations. Stochasticity introduces large statistical variation into the time-averaged profiles of this small quantity. Nevertheless, all of the 128^3 and 256^3 runs agree with each other as well as with the 512^3 PPMSTAR run on the profile of \mathcal{F}_k within the statistical-variation bands included in the figure.

Why is the magnitude of \mathcal{F}_k so much smaller than that of \mathcal{F}_H ? Using a simplified, two-stream model of convection, we show in Appendix B that the magnitude of \mathcal{F}_k depends on the degree of asymmetry between upflows and downflows. We characterise the asymmetry using a downflow filling factor f_d defined to be the relative horizontal area covered by flows with vertical velocity $v_y - \bar{v}_y < 0$. In the two-stream model, \mathcal{F}_k vanishes for $f_d = 0.5$, that is when there is perfect up-down symmetry. Figure 17 shows that f_d indeed is close to 0.5 in our simulations. In contrast to \mathcal{F}_k , \mathcal{F}_H remains a substantial fraction of the total flux even if there is perfect up-down symmetry because convection must transport heat across the convective layer. This explains why the amplitude of \mathcal{F}_k is much smaller than that of \mathcal{F}_H in our simulations. The two-stream model also predicts that \mathcal{F}_k is negative (i.e. directed downwards) for $f_d < 0.5$ and positive for $f_d > 0.5$. We observe the same trend in our simulations: compare Figs. 16 and 17 and see also Fig. B.2. The role of \mathcal{F}_k is much more important in strongly stratified, convective stellar envelopes, in which downflows tend to be substantially narrower than upflows and \mathcal{F}_k is large and negative (i.e. directed downwards). Viallet et al. (2013) show this difference directly in their Fig. 13, in which they compare their 3D hydrodynamic models of the convective envelope of a red giant and of an oxygen-burning shell in a massive star. The latter model is rather similar to our test setup and, not surprisingly, they also find that the amplitude of \mathcal{F}_k is ≈ 30 times smaller than that of \mathcal{F}_H in that model (compare their Figs. 13 and 15).

4. Summary and conclusions

In this work, we have defined a test problem involving adiabatic turbulent convection and mass entrainment from a stably stratified layer. The problem is directly relevant to current 3D simulation efforts to model convection in stellar interiors. The problem's physics and geometry are simplified to make it accessible to a wide range of 3D hydrodynamic codes but to retain the crucial processes we are investigating. Because turbulent flows are involved, this test problem is fundamentally different from

standard test problems. There is no single solution and the simulations must be compared in a statistical way, taking care not to overinterpret statistical fluctuations as real differences between codes or simulations. We compare simulations computed using the codes FLASH, MUSIC, PPMSTAR, PROMPI, and SLH, which we run for 25 convective turnover timescales to gather as much statistics as possible.

All of the simulations, computed on Cartesian grids ranging from 128^3 through 256^3 to 512^3 cells, show a turbulent convective layer dominated by large-scale flows with an rms Mach number of ≈ 0.04 . Upflows turning around at the upper convective boundary entrain mass from the bottom of the overlying stable layer and pull it into the convective layer. They also generate internal gravity waves, which are present throughout the stable layer. The bulk rms velocities as well as time- and space-averaged velocity profiles in both the convective and stable layer are statistically the same in all five codes even on the coarse 128^3 grid. They also agree with the profiles obtained on finer grids, although the 512^3 PPMSTAR run deviates slightly because of two long episodes of increased velocity, likely of statistical origin. All of the codes converge to the same kinetic energy spectrum upon grid refinement, which is consistent with Kolmogorov's $-\frac{5}{3}$ scaling. The five codes slightly differ in their dissipation rates close to the grid scale, which is reflected in slight differences in the shape of the kinetic energy spectrum in the dissipation range. The codes also give essentially the same spectrum in the stable layer for horizontal wavelengths $\gtrsim 26$ computational cells.

The convective boundary is somewhat under-resolved on the 128^3 grid, being as much as $\approx 50\%$ thicker than its converged thickness value as determined by the 256^3 and 512^3 runs. Nevertheless, the simulations agree on the total amount of mass entrained by the simulations' end within 9% already on the 128^3 grid. This improves to 4% on the 256^3 grid and the remaining spread seems to be dominated by stochasticity. The rapid convergence of the mass entrainment rate upon grid refinement is compatible with the idea of Spruit (2015) that the rate is set by the global energetics of the whole process as opposed to details of the small-scale instabilities occurring in the boundary layer. We also show that approximately 30–60 per cent of the entrainment rate can be attributed to a process in which mass entrainment is caused by the constant heating of the convective layer and the subsequent increase in its mean entropy.

Finally, all of the codes statistically agree on the time-averaged profiles of fluctuations in composition and entropy and on the profiles of the enthalpy- and kinetic-energy fluxes. This agreement is somewhat better on the 256^3 grid as compared to the 128^3 one. We give likely reasons for the rapid convergence of the time-averaged enthalpy flux, which makes it a rather poor indicator of code or simulation quality. The flux of kinetic energy is ≈ 30 times smaller than the flux of enthalpy in our test problem. We explain this disparity using a toy model showing that the flux of kinetic energy is very small indeed when the horizontal areas covered by upflows and downflows are close to being the same as observed in our simulations.

All in all, we find excellent code-to-code agreement on a rather complex turbulent-flow problem. The numerical schemes implemented in these codes differ in many aspects such as numerical flux functions, reconstruction and time-stepping methods, or the use (or not) of dimensional splitting. It would certainly be interesting to compare our results with those obtained using finite difference or spectral methods, which are not included in the present work. To facilitate future studies of this kind, we formulated the test problem in Cartesian geometry and such that the Mach number of the flows (≈ 0.04 rms) is easy

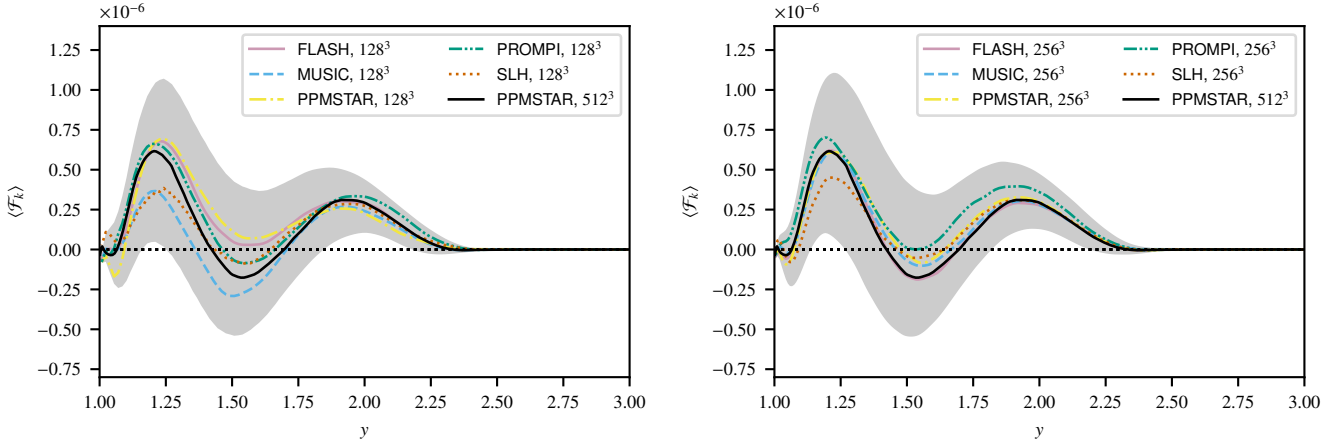


Fig. 16. Vertical profiles of the kinetic-energy flux averaged over the whole analysis time interval ($500 \leq t \leq 2000$, approximately $19\tau_{\text{conv}}$). The grey bands give an estimate of $\pm 3\sigma$ statistical variation in the averages due to stochasticity, see Sect. 3.1 for details.

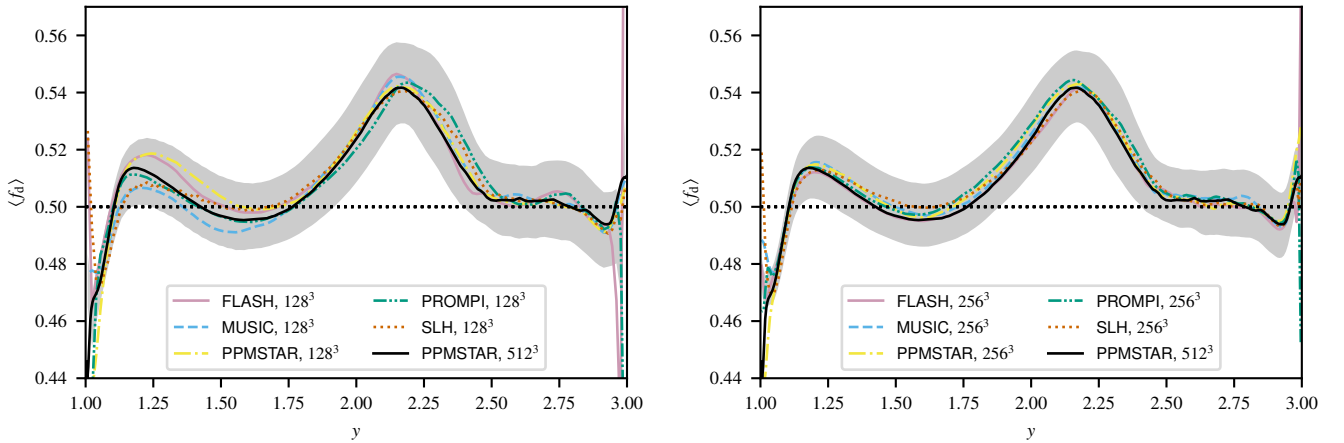


Fig. 17. Vertical profiles of the downflow filling factor averaged over the whole analysis time interval ($500 \leq t \leq 2000$, approximately $19\tau_{\text{conv}}$). The grey bands give an estimate of $\pm 3\sigma$ statistical variation in the averages due to stochasticity, see Sect. 3.1 for details.

to reach for a wide range of codes. We provide our data as well as all of the setup files and analysis scripts needed if anyone should be interested in performing and analysing simulations of the kind presented here in the future, see Appendix A. A useful, if quite expensive, future extension of this study could involve a series of simulations to see how weakly one can drive the convection and still maintain reasonable code-to-code agreement on affordable grids. With a lower driving luminosity, the mass entrainment rate would also decrease and allow us to gather statistics of the waves in the stable layer on long timescales, which would provide well-resolved temporal wave spectra and test the accuracy of the codes' asteroseismic predictions.

Acknowledgements. P. V. F. E. was supported by the US Department of Energy through the Los Alamos National Laboratory (LANL). LANL is operated by Triad National Security, LLC, for the National Nuclear Security Administration of the US Department of Energy (Contract No. 89233218CNA000001). This work has been assigned a document release number LA-UR-21-25840. R. A., J. H., L. H., G. L., and F. K. R. acknowledge support by the Klaus Tschira Foundation. The work of F. K. R. is supported by the German Research Foundation (DFG) through the grant RO 3676/3-1. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 – 390900948 (the Heidelberg STRUCTURES Excellence Cluster). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS (Jülich Supercomputing Centre 2019) at Jülich Supercomputing Centre (JSC). F. H.

acknowledges funding through an NSERC Discovery Grant. This work has benefited from and motivated in part by research performed as part of the JINA Center for the Evolution of the Elements (NSF Grant No. PHY-1430152). P. R. W. acknowledges support from NSF grants 1814181, 2032010, and PHY-1430152, as well as computing support through NSF's Frontera computing system at TACC. The software used in this work was in part developed by the DOE NNSA-ASC OASCR FLASH Center at the University of Chicago. This work is partly supported by the ERC grant No. 787361- COBOM and the consolidated STFC grant ST/R000395/1. The authors would like to acknowledge the use of the University of Exeter High-Performance Computing (HPC) facility ISCA. DiRAC Data Intensive service at Leicester, operated by the University of Leicester IT Services, which forms part of the STFC DiRAC HPC Facility. The equipment was funded by BEIS capital funding via STFC capital grants ST/K000373/1 and ST/R002363/1 and STFC DiRAC Operations grant ST/R001014/1. R. H. acknowledges support from the World Premier International Research Centre Initiative (WPI Initiative), MEXT, Japan and the IReNA AccelNet Network of Networks, supported by the National Science Foundation under Grant No. OISE-1927130. This article is based upon work from the ChETEC COST Action (CA16117), supported by COST (European Cooperation in Science and Technology). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101008324 (ChETEC-INFRA). This work used the DiRAC@Durham facility managed by the Institute for Computational Cosmology on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The equipment was funded by BEIS capital funding via STFC capital grants ST/P002293/1 and ST/R002371/1, Durham University, and STFC operations grant ST/R000832/1. This work also used the DiRAC Data Centric system at Durham University, operated by the Institute for Computational Cosmology on behalf of the STFC DiRAC HPC Facility. This equipment was funded by BIS National E Infrastructure capital grant ST/K00042X/1, STFC capital grants ST/H008519/1 and ST/K00087X/1, STFC DiRAC Operations

grant ST/K003267/1, and Durham University. DiRAC is part of the National E Infrastructure. S. W. C. acknowledges federal funding from the Australian Research Council through a Future Fellowship (FT160100046) and Discovery Project (DP190102431). This work was supported by computational resources provided by the Australian Government through NCI via the National Computational Merit Allocation Scheme (project ew6), and resources provided by the Pawsey Supercomputing Centre which is funded by Australian Government and the Government of Western Australia. We thank the anonymous referee for constructive comments that improved this paper.

References

- Aerts, C. 2021, *Rev. Mod. Phys.*, **93**, 015001
- Alvan, L., Brun, A. S., & Mathis, S. 2014, *A&A*, **565**, A42
- Andrassy, R., Herwig, F., Woodward, P., & Ritter, C. 2020, *MNRAS*, **491**, 972
- Arnett, D., Meakin, C., & Young, P. A. 2009, *ApJ*, **690**, 1715
- Baraffe, I., Pratt, J., Goffrey, T., et al. 2017, *ApJ*, **845**, L6
- Battino, U., Pignatari, M., Ritter, C., et al. 2016, *ApJ*, **827**, 30
- Beeck, B., Collet, R., Steffen, M., et al. 2012, *A&A*, **539**, A121
- Berberich, J. P., Chandrashekar, P., & Klingenberg, C. 2021, *Comput. Fluids*, **104858**
- Böhm-Vitense, E. 1958, *Z. Astrophys.*, **46**, 108
- Castro, N., Fossati, L., Langer, N., et al. 2014, *A&A*, **570**, L13
- Christensen-Dalsgaard, J., Silva Aguirre, V., Cassisi, S., et al. 2020, *A&A*, **635**, A165
- Claret, A., & Torres, G. 2019, *ApJ*, **876**, 134
- Colella, P., & Glaz, H. M. 1985, *J. Comput. Phys.*, **59**, 264
- Colella, P., & Woodward, P. R. 1984, *J. Comput. Phys.*, **54**, 174
- Constantino, T., Campbell, S. W., & Lattanzio, J. C. 2017, *MNRAS*, **472**, 4900
- Couch, S. M., & Ott, C. D. 2015, *ApJ*, **799**, 5
- Cristini, A., Hirschi, R., Meakin, C., et al. 2019, *MNRAS*, **484**, 4645
- Denissenkov, P. A., Herwig, F., Bildsten, L., & Paxton, B. 2013, *ApJ*, **762**, 8
- Denissenkov, P. A., Herwig, F., Woodward, P., et al. 2019, *MNRAS*, **488**, 4258
- Dimonte, G., Youngs, D. L., Dimits, A., et al. 2004, *Phys. Fluids*, **16**, 1668
- Dobler, W., Haugen, N. E., Yousef, T. A., & Brandenburg, A. 2003, *Phys. Rev. E*, **68**, 026304
- Doherty, C. L., Siess, L., Lattanzio, J. C., & Gil-Pons, P. 2010, *MNRAS*, **401**, 1453
- Edelmann, P. V. F. 2014, Dissertation, Technische Universität München, Germany
- Edelmann, P. V. F., & Röpke, F. K. 2016, in *JUQUEEN Extreme Scaling Workshop 2016*, eds. D. Brömmel, W. Frings, & B. J. N. Wylie, 63, JSC Internal Report No. FZJ-JSC-IB-2016-01
- Edelmann, P. V. F., Röpke, F. K., Hirschi, R., Georgy, C., & Jones, S. 2017, *A&A*, **604**, A25
- Edelmann, P. V. F., Ratnasingam, R. P., Pedersen, M. G., et al. 2019, *ApJ*, **876**, 4
- Edelmann, P. V. F., Horst, L., Berberich, J. P., et al. 2021, *A&A*, **652**, A53
- Ekström, S., Georgy, C., Eggenberger, P., et al. 2012, *A&A*, **537**, A146
- Falkovich, G. 1994, *Phys. Fluids*, **6**, 1411
- Federrath, C., Roman-Duval, J., Klessen, R. S., Schmidt, W., & Mac Low, M. M. 2010, *A&A*, **512**, A81
- Fleck, B., Carlsson, M., Khomenko, E., et al. 2021, *Phil. Trans. R. Soc. London, Ser. A*, **379**, 20200170
- Fryxell, B., Müller, E., & Arnett, D. 1989, in *Hydrodynamics and nuclear burning*, (Max-Planck-Inst. für Physik und Astrophysik), Max-Planck-Institut für Physik und Astrophysik München: MPA, 449
- Fryxell, B., Arnett, D., & Mueller, E. 1991, *ApJ*, **367**, 619
- Fryxell, B., Olson, K., Ricker, P., et al. 2000, *ApJS*, **131**, 273
- Geroux, C., Baraffe, I., Viallet, M., et al. 2016, *A&A*, **588**, A85
- Gilet, C., Almgren, A. S., Bell, J. B., et al. 2013, *ApJ*, **773**, 137
- Goffrey, T., Pratt, J., Viallet, M., et al. 2017, *A&A*, **600**, A7
- Hammer, N., Jamitzky, F., Satzger, H., et al. 2016, in *Parallel Computing: On the Road to Exascale, Proceedings of the International Conference on Parallel Computing, ParCo 2015, 1-4 September 2015, Edinburgh, Scotland, UK*, 827
- Herwig, F. 2000, *A&A*, **360**, 952
- Herwig, F., Woodward, P. R., Lin, P.-H., Knox, M., & Fryer, C. 2014, *ApJ*, **792**, L3
- Higl, J., Müller, E., & Weiss, A. 2021, *A&A*, **646**, A133
- Horst, L., Edelmann, P. V. F., Andrassy, R., et al. 2020, *A&A*, **641**, A18
- Horst, L., Hirschi, R., Edelmann, P. V. F., Andrassy, R., & Roepke, F. K. 2021, *A&A*, **653**, A55
- Joggerst, C. C., Nelson, A., Woodward, P., et al. 2014, *J. Comput. Phys.*, **275**, 154
- Jones, S., Andrassy, R., Sandalski, S., et al. 2017, *MNRAS*, **465**, 2991
- Jørgensen, A. C. S., Mosumgaard, J. R., Weiss, A., Silva Aguirre, V., & Christensen-Dalsgaard, J. 2018, *MNRAS*, **481**, L35
- Jülich Supercomputing Centre 2019, *J. Large-scale Res. Facil.*, **5**
- Kim, J.-H., Abel, T., Agertz, O., et al. 2014, *ApJS*, **210**, 14
- Kim, J.-H., Agertz, O., Teyssier, R., et al. 2016, *ApJ*, **833**, 202
- Lecoanet, D., McCourt, M., Quataert, E., et al. 2016, *MNRAS*, **455**, 4274
- Lecoanet, D., Cantiello, M., Anders, E. H., et al. 2021, *MNRAS*, **508**, 132
- Lee, D., & Deane, A. E. 2009, *J. Comput. Phys.*, **228**, 952
- Li, X.-S., & Gu, C.-W. 2008, *J. Comput. Phys.*, **227**, 5144
- Linden, P. F. 1975, *J. Fluid Mech.*, **71**, 385
- Liou, M.-S. 2006, *J. Comput. Phys.*, **214**, 137
- MacNeice, P., Olson, K. M., Mobarry, C., de Fainchtein, R., & Packer, C. 2000, *Comput. Phys. Commun.*, **126**, 330
- Maeder, A. 1976, *A&A*, **47**, 389
- Magic, Z., Weiss, A., & Asplund, M. 2015, *A&A*, **573**, A89
- McNally, C. P., Lyra, W., & Passy, J.-C. 2012, *ApJS*, **201**, 18
- Meakin, C. A., & Arnett, D. 2007, *ApJ*, **667**, 448
- Miczek, F. 2013, Dissertation, Technische Universität München, Germany
- Miczek, F., Röpke, F. K., & Edelmann, P. V. F. 2015, *A&A*, **576**, A50
- Mocák, M., Meakin, C., Campbell, S. W., & Arnett, W. D. 2018, *MNRAS*, **481**, 2918
- Plewa, T., Calder, A. C., & Lamb, D. Q. 2004, *ApJ*, **612**, L37
- Prandtl, L. 1925, *Zeitschrift Angewandte Mathematik und Mechanik*, **5**, 136
- Pratt, J., Baraffe, I., Goffrey, T., et al. 2017, *A&A*, **604**, A125
- Pratt, J., Baraffe, I., Goffrey, T., et al. 2020, *A&A*, **638**, A15
- Ramaprabhu, P., Dimonte, G., Woodward, P., et al. 2012, *Phys. Fluids*, **24**, 074107
- Rosvick, J. M., & Vandenberg, D. A. 1998, *AJ*, **115**, 1516
- Scannapieco, E., & Brüggem, M. 2015, *ApJ*, **805**, 158
- Scott, L. J. A., Hirschi, R., Georgy, C., et al. 2021, *MNRAS*, **503**, 4208
- Silva Aguirre, V., Christensen-Dalsgaard, J., Cassisi, S., et al. 2020, *A&A*, **635**, A164
- Sonoi, T., Ludwig, H. G., Dupret, M. A., et al. 2019, *A&A*, **621**, A84
- Spruit, H. C. 2015, *A&A*, **582**, L2
- Staritsin, E. I. 2013, *Astron. Rep.*, **57**, 380
- Stephens, D., Herwig, F., Woodward, P., et al. 2021, *MNRAS*, **504**, 744
- Sutherland, B. R. 2010, *Internal Gravity Waves* (Cambridge University Press)
- Sytine, I. V., Porter, D. H., Woodward, P. R., Hodson, S. W., & Winkler, K. H. 2000, *J. Comput. Phys.*, **158**, 225
- Timmes, F. X., & Swesty, F. D. 2000, *ApJS*, **126**, 501
- Toro, E. F. 2009, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* (Berlin Heidelberg: Springer)
- Trampedach, R., Stein, R. F., Christensen-Dalsgaard, J., Nordlund, Å., & Asplund, M. 2014, *MNRAS*, **442**, 805
- Van Leer, B. 1974, *J. Comput. Phys.*, **14**, 361
- Viallet, M., Meakin, C., Arnett, D., & Mocák, M. 2013, *ApJ*, **769**, 1
- Viallet, M., Goffrey, T., Baraffe, I., et al. 2016, *A&A*, **586**, A153
- Willcox, D. E., Townsley, D. M., Calder, A. C., Denissenkov, P. A., & Herwig, F. 2016, *ApJ*, **832**, 13
- Woodward, P. R. 1986, in *Astrophysical Radiation Hydrodynamics*, eds. K. H. A. Winkler, & M. L. Norman (Dordrecht: Springer), 188, 245, <https://www.lcse.umn.edu/PPMlogo>
- Woodward, P. R. 2007, in *Implicit Large Eddy Simulation, Computing Turbulent Fluid Dynamics*, eds. F. F. Grinstein, L. G. Margolin, & W. J. Rider (Cambridge: Cambridge University Press), 130, 2007
- Woodward, P., & Colella, P. 1981, in *Lecture Notes in Physics*, eds. W. C. Reynolds, & R. W. MacCormack (Berlin: Springer Verlag), 434
- Woodward, P., & Colella, P. 1984, *J. Comput. Phys.*, **54**, 115
- Woodward, P. R., Herwig, F., & Lin, P.-H. 2015, *ApJ*, **798**, 49
- Woodward, P. R., Lin, P.-H., Mao, H., Andrassy, R., & Herwig, F. 2019, *J. Phys.: Conf. Ser.*, **1225**, 012020

Appendix A: Supplementary materials

To make this study easy to reproduce or extend, we provide our data products as well as setup and analysis scripts in electronic form. The 1D and 2D data, animations and spectra are available on Zenodo¹⁰. A set of Jupyter notebooks and Python scripts that can set up the test problem and reproduce our analysis as well as all plots shown in this work can be found on the CoCoPy GitHub repository¹¹, v1.1.0. The analysis makes use of the PyPPM toolkit¹², v2.0.1. To make the analysis even more approachable, we set up the CoCo Jupyter Hub¹³, which contains all of the data and scripts needed. The hub also contains the much more voluminous original 128³ and 256³ data sets. We will consider adding the 512³ runs upon request.

The hydrodynamic codes write 3D output into binary data files, the structure of which is code-dependent. We have created a Python interface for each of the codes. The interface makes sure that the 3D data are read and assigned to a set of arrays, which are then further processed in a code-independent way. The processing is performed in the system of units defined in Table 1 independently of what units the hydrodynamic codes use. As detailed below, we produce text files containing 1D horizontal averages and kinetic energy spectra as well as binary files containing 2D slices through the simulation box. This intermediate step speeds up data visualisation and facilitates data sharing.

The 1D horizontal averages are written into text-based `.rprof` files¹⁴ with one file per each output interval. The files contain a two-line header followed by a set of data tables containing the quantities summarised in Table A.1 such that the files are both easily human- and machine-readable.

The kinetic-energy spectra are computed in the $y = 1.7$ and $y = 2.7$ planes using Eq. 23. They are written into text-based `.spec` files with one file per each output interval. The files contain a one-line header followed by two data tables, the first for the $y = 1.7$ plane and the other for the $y = 2.7$ plane. The tables list the value of $\frac{1}{2} \|\Psi\|^2$ as a function of the wavenumber k ranging from $k = 0$ to the Nyquist wavenumber.

At each output interval, we also produce a set of NumPy's binary `.npy` files, each containing one variable in a 2D slice through the computation box. The slicing planes are $x = 0$, $y = 1.7$, $y = 2.7$, and $z = 0$ and the variables are the pseudo-entropy $A = p/\rho^\gamma$, density ρ , mass fraction X_1 of the μ_1 fluid, the Cartesian components of velocity v_x , v_y , and v_z , magnitude of vorticity $|\nabla \times \mathbf{v}|$, and velocity divergence $\nabla \cdot \mathbf{v}$.

Appendix B: Flux of kinetic energy in a two-stream model of convection

In this section, we derive a simple relation between the asymmetry between upflows and downflows, quantified by the downflow filling factor, and the flux of kinetic energy in a 1D, two-stream model of convection. Although purely kinematic in nature, the model shows how contributions from the upflows cancel those from the downflows, making the flux of kinetic energy vanish when the flow is close to perfect up-down symmetry.

We consider a 1D model of convection that consists of an upflow and a downflow such that the net mass flux through the whole horizontal plane vanishes. Both streams are assumed to have the same density ρ_0 for simplicity.¹⁵ Let $0 < f_d < 1$ be the geometrical filling factor (relative surface area) of the downflow. The net mass flux is then

$$\rho_0(1 - f_d)u_u + \rho_0 f_d u_d = 0, \quad (\text{B.1})$$

where $u_u > 0$ and $u_d < 0$ are the velocities of the upflow and downflow, respectively. The ratio of the velocity components is

$$\frac{u_d}{u_u} = -\frac{1 - f_d}{f_d} \quad (\text{B.2})$$

and the rms velocity is

$$u_{\text{rms}} = \sqrt{(1 - f_d)u_u^2 + f_d u_d^2}. \quad (\text{B.3})$$

Using Eq. B.2, we can express the individual velocities of the two streams in terms of u_{rms} :

$$u_u = \sqrt{\frac{f_d}{1 - f_d}} u_{\text{rms}}, \quad (\text{B.4})$$

$$u_d = -\sqrt{\frac{1 - f_d}{f_d}} u_{\text{rms}}. \quad (\text{B.5})$$

The net flux of kinetic energy is

$$\mathcal{F}_k = \frac{1}{2} \rho_0 [(1 - f_d)u_u^3 + f_d u_d^3], \quad (\text{B.6})$$

which can be expressed as a function of f_d and u_{rms} :

$$\mathcal{F}_k = \frac{1}{2} \rho_0 u_{\text{rms}}^3 f_d \left(\frac{1 - f_d}{f_d} \right)^{\frac{3}{2}} \left[\left(\frac{f_d}{1 - f_d} \right)^2 - 1 \right]. \quad (\text{B.7})$$

Figure B.1 shows the geometric factor $\mathcal{F}_k \left(\frac{1}{2} \rho_0 u_{\text{rms}}^3 \right)^{-1}$. Because $\rho_0 > 0$ and $u_{\text{rms}} > 0$, the flux is negative (i.e. directed downwards) for $f_d < 0.5$, positive for $f_d > 0.5$, and it vanishes when $f_d = 0.5$, that is when there is perfect up-down symmetry and the upflow and downflow contributions cancel each other in Eq. B.6.

We test the two-stream model by taking its input quantities from the 512³ PPMSTAR run and comparing the result with the actual profile of \mathcal{F}_k in the same run. In particular, we set $\rho_0 = \langle \bar{\rho} \rangle$, $f_d = \langle f_d \rangle$ and $u_{\text{rms}}^3 = \langle \bar{v}_{\text{rms}} \rangle^2 \langle \bar{v}_{y,\text{rms}} \rangle$ in Eq. B.7. The last expression is motivated by the fact that we are computing the flux of kinetic energy along the y axis. We use the whole analysis time interval ($500 \leq t \leq 2000$, approximately $19\tau_{\text{conv}}$) for the time averaging. Figure B.2 shows that the two-stream model closely reproduces the actual profile of \mathcal{F}_k both in magnitude and sign over most of the convective layer.

¹⁰ <https://doi.org/10.5281/zenodo.5796842>

¹¹ <https://github.com/robert-andrassy/CoCoPy>

¹² <https://github.com/PPMstar/PyPPM>

¹³ <https://www.ppmstar.org/coco>

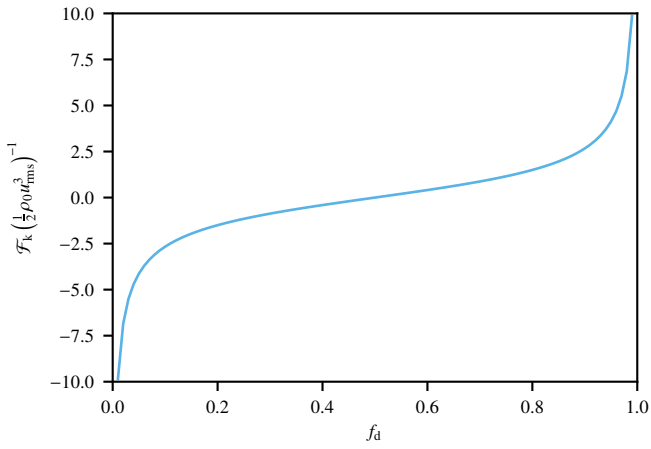
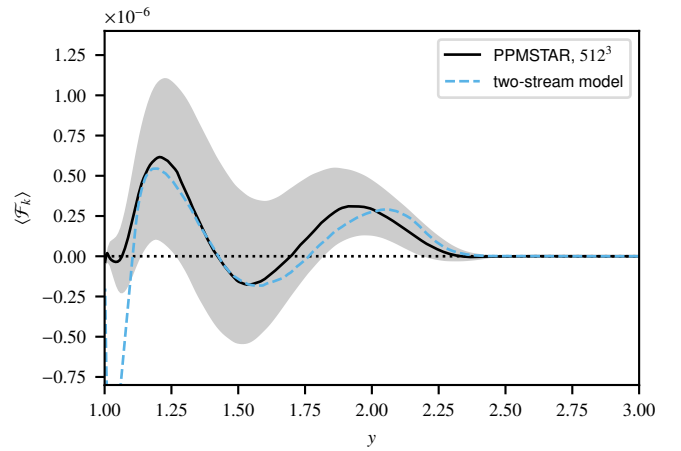
¹⁴ The extension `.rprof` is of historical origin: such files have mostly been used to store radial profiles of full-sphere PPMSTAR simulations of stellar convection.

¹⁵ We can neglect density fluctuations because (1) we do not model the dynamics of convection here and (2) the fluctuations are small.

Table A.1. Definitions of the horizontally averaged quantities available in the `.rprof` files.

Variable Name	Meaning	Definition	Statistics
Y	vertical coordinate	y	...
RHO	density	$\bar{\rho}$	full
P	pressure	\bar{p}	full
TEMP	temperature	\bar{T}	full
A	pseudo-entropy	\bar{A} ; $A = p/\rho^\gamma$	full
X1	mass fraction of the lighter fluid	\bar{X}_1	full
V	magnitude of velocity	$\widetilde{ \mathbf{v} }$	full
VX	x -component of velocity	$\widetilde{v_x}$	full
VY	y -component of velocity	$\widetilde{v_y}$	full
VZ	z -component of velocity	$\widetilde{v_z}$	full
VY	magnitude of v_y	$ \widetilde{v_y} $	average
VXZ	magnitude of horizontal velocity	$\widetilde{v_{xz}}$; $v_{xz} = \sqrt{v_x^2 + v_z^2}$	average
VORT	magnitude of vorticity	$\widetilde{\omega}$; $\omega = \nabla \times \mathbf{v} $	average
FK	flux of kinetic energy	$\mathcal{F}_k = \frac{1}{2} (\rho \mathbf{v} ^2 v_y - \rho \mathbf{v} ^2 \widetilde{v_y})$...
FH	flux of enthalpy	$\mathcal{F}_H = \overline{H v_y} - \overline{H} \widetilde{v_y}$...
FFD	downflow filling factor	fraction of the total horizontal area where $v_y - \widetilde{v_y} < 0$...

Notes. In the last column, ‘average’ means that only the average is computed and ‘full’ indicates that also the minimum, maximum, and standard deviation are computed. The names of the last three quantities then start with `MIN_`, `MAX_`, and `STDEV_`, respectively.


Fig. B.1. Geometric factor $\mathcal{F}_k \left(\frac{1}{2} \rho_0 u_{\text{rms}}^2 \right)^{-1}$ in Eq. B.7 as a function of the downflow filling factor f_d .

Fig. B.2. Profile of the kinetic-energy flux \mathcal{F}_k given by the two-stream model as compared with the actual profile of \mathcal{F}_k in the 512^3 PPMSTAR run. Inputs for the two-stream model are taken from the same PPMSTAR run. The time-averaging interval and statistical-variation bands are the same as those used in Fig. 16.