



MASTER THESIS

ASTROPHYSICS INSTITUT COLOGNE

**“Modelling turbulent gases with Finite Volume
and Discontinuous Galerkin methods ”**

submitted by

JOHANNES MARKERT

Köln - June 6, 2017

Contents

1	Introduction	1
2	Theory	2
2.1	Governing equations	2
2.1.1	Compressible Euler Equations	2
2.1.2	Weak Formulation	4
2.2	Turbulence Statistics	5
2.2.1	Energy Cascade & Powerspectrum	7
2.2.2	Probability Distribution Functions (PDF)	9
2.3	Supersonic Shocks	10
2.3.1	Shock Formation	10
2.3.2	Method of Characteristics	10
2.3.3	Entropy Increase	12
2.4	Finite Element Scheme	12
2.4.1	Galerkin Method	13
2.4.2	Flux Functions	17
2.4.3	Time Integration	19
3	Numerical Prerequisites	20
3.1	Computational Frameworks	20
3.2	Turbulent Forcing	22
3.3	Shock Capturing	24
3.4	Grid Spaces & Data Transformation	28
4	Results & Discussion	31
4.1	SOD Shock Tube Problem	31
4.1.1	Adiabatic & Isothermal Shock	32
4.1.2	Adiabatic & Isothermal Sod with Strong Shock	33
4.2	Driven Turbulence	35
4.2.1	Mach Number Evolution	36
4.2.2	Column Density & Velocity	37
4.2.3	FV-DG Mode Switching	44
4.2.4	Energy Dissipation	45
4.2.5	Density & Velocity Distributions	47
4.2.6	Energy & Velocity Powerspectra	51
4.2.7	Summary	55
4.3	Decaying Turbulence	55
4.3.1	Mach Number Evolution	56
4.3.2	Column Density & Velocity	56
4.3.3	FV-DG Mode Switching	66
4.3.4	Energy Dissipation	67
4.3.5	Density & Velocity Distributions	69
4.3.6	Energy & Velocity Powerspectra	71
4.3.7	Summary	73
5	Conclusion & Outlook	74

1 Introduction

Astrophysical turbulence simulations of interstellar gases play vital role in the understanding and modelling of star formation. A multitude of numerical schemes and computational fluid dynamics software has been developed each with their own merits and drawbacks.

Finding the best numerical solver for a specific problem domain is a vibrant field of research as of today even though the first pioneers began to conduct simple physics simulations many decades ago. A lot has changed since these days. The present explosion in computing power paralleled by a striking inflation of operational costs passes on a mighty tool to the astrophysics community which opens the door to numerical experiments unfeasible ten years ago.

Alongside the advent of new technical possibilities modern numerical schemes appear which are better suited for utilizing latest CPU architectures to gain better performance. Especially, the aeronautics and car industries push forward the development of new simulation software. They are interested in modelling flight characteristics for new airplane designs, the aerodynamic drag of cars or want to increase the efficiency of combustion engines. Consequently, innovative numerical schemes should be highly flexible with regards to meshing complex geometries, very accurate, massively parallelizable and blazingly fast. The demands for underlying physical model are moderate. The Navier-Stokes equations suffice on most occasions. One of the latest specimen of their kind is the high-order accurate CFD software FLEXI developed by a team lead by Prof. Claus-Dieter Munz in the Aeronautics Institut in Stuttgart, Germany. The basic idea is to introduce piecewise polynomial functions of order higher order. The scheme is called Discontinuous Galerkin method.

In contrast to aeronautical models the demands for turbulence simulations of astrophysical gases are quite the opposite. In free space there are no complex geometries at least not in proximity of massive objects like black holes. The physical model become extremely complex when electro-magnetic fields, radiation, chemistry and gravity are introduced to the governing equations. Furthermore a huge range of scales in time, space, energy, density and pressure must be covered, too. As if this is not enough high Mach numbers in the gas give rise to strong shock conditions and discontinuities which challenge the stability and accuracy of every numerical solver. Up to now the astrophysics community considered finite-volume schemes of first/second order to be the only viable method for conducting their simulations. FLASH originally developed by researchers at the University of Chicago is the name for a well established nuclear and astrophysics simulation software with finite-volumes as their foundational scheme.

This thesis is a first attempt to introduce higher order Galerkin methods to astrophysical turbulence simulations. The statistical evaluation of simple turbulence setups run by FLASH and FLEXI are expected to give an insight into the potential Galerkin schemes might offer to the astrophysics community.

2 Theory

2.1 Governing equations

2.1.1 Compressible Euler Equations

The compressible Euler Equations are hyperbolic conservation equations which model perfect fluids without interaction of their constituents. We assume no heat conduction ($\mathbb{T}^{i0} = \mathbb{T}^{0i} = 0$), no viscosity ($\mathbb{T}^{ij} = p\mathbb{I}$, $\mu_d = 0$) and no gravity $g = 0$. Within in the comoving frame the *stress-energy tensor* \mathbb{T} reads:

$$\mathbb{T}^{\alpha\beta} = \text{diag}(\rho c^2, p, p, p) = \left(\rho + \frac{p}{c^2} \right) u^\alpha u^\beta + p \mathbb{G}^{\alpha\beta} \quad (1)$$

In the flat spacetime the *metric tensor* is set to $\mathbb{G} = \text{diag}(-1, 1, 1, 1)$. The total energy and the number of particles are conserved.

$$\partial_\nu \mathbb{T}^{\mu\nu} = 0 \quad (2)$$

$$\partial_\mu (n u^\mu) = 0 \quad (3)$$

Taking the non-relativistic limit, we arrive at the in conservative form of the Euler equations.

$$\partial_t \rho + \nabla \cdot (\rho \underline{u}) = 0 \quad \text{mass conservation} \quad (4)$$

$$\partial_t (\rho \underline{u}) + \nabla \cdot (\rho \underline{u} \underline{u}^T) + \nabla p = \underline{F} \quad \text{momentum conservation} \quad (5)$$

$$\partial_t E + \nabla \cdot (\underline{u} (E + p)) = 0, \quad \text{energy conservation} \quad (6)$$

where the total energy E is composed of the internal energy \mathcal{I} and the kinetic energy \mathcal{K} .

$$E = \mathcal{I} + \mathcal{K} = \frac{p}{\gamma - 1} + \frac{\rho}{2} \underline{u}^2, \quad (7)$$

with γ being the *adiabatic constant*.

The source term \underline{F} allows us to perpetually inject a force field which gets important in the discussion of driven turbulence later on. See section 3.2.

Equation of State If not stated otherwise all simulations follow the *ideal gas law*.

$$p = \frac{c^2}{\gamma} \rho = R T \rho = \frac{R}{c_v} \mathcal{I} = (\gamma - 1) \mathcal{I}, \quad (8)$$

where R is the specific ideal gas constant, T is the gas temperature and $c_v = \frac{\gamma-1}{R}$ is the specific heat capacity at constant volume.

The γ is set to

$$\gamma = \frac{c_p}{c_v} := \frac{5}{3}, \quad (9)$$

which represents a mono-atomic gas without interacting forces.

The speed of sound c is a direct consequence of the ideal gas equation.

$$c^2 = \gamma \frac{p}{\rho} := C_P = \text{const.} \quad (10)$$

During the numerical simulation the equation of state is enforced via the *polytropic process* (also called *polytropic cooling*) at every timestep.

$$p = C_P \rho^\Gamma, \quad (11)$$

where the *polytropic exponent* is set to $\Gamma := 1$ which is equivalent to an isothermal process. A thorough derivation can be found in [?], p.2-7.

Dimensionless Euler Equations We want to show that the Euler equations are invariant to changes of units. This discussion is useful since most numerical frameworks do not support physical units and rescaled physical quantities avoid truncation errors due to the limits of floating point operations. For this, we choose a characteristic length l_r , a characteristic velocity u_r and a characteristic density ρ_r . Multiplying suitable combinations of these constants with the Euler equations yields

$$[\partial_t \rho + \nabla \cdot (\rho \underline{u})] \cdot \frac{l_r}{\rho_r u_r} = 0 \quad (12)$$

$$[\partial_t (\rho \underline{u}) + \nabla \cdot (\rho \underline{u} \underline{u}^T) + \nabla p - \underline{F}] \cdot \frac{l_r}{\rho_r u_r^2} = 0 \quad (13)$$

$$[\partial_t E + \nabla \cdot (\underline{u} (E + p))] \cdot \frac{l_r}{\rho_r u_r^3} = 0 \quad (14)$$

We simplify and get

$$\partial_{\tilde{t}} \tilde{\rho} + \tilde{\nabla} \cdot (\tilde{\rho} \tilde{\underline{u}}) = 0 \quad (15)$$

$$\partial_{\tilde{t}} (\tilde{\rho} \tilde{\underline{u}}) + \tilde{\nabla} \cdot (\tilde{\rho} \tilde{\underline{u}} \tilde{\underline{u}}^T) + \tilde{\nabla} \tilde{p} - \tilde{\underline{F}} = 0 \quad (16)$$

$$\partial_{\tilde{t}} \tilde{E} + \tilde{\nabla} \cdot (\tilde{\underline{u}} (\tilde{E} + \tilde{p})) = 0, \quad (17)$$

where $t_r = \frac{l_r}{u_r}$ (characteristic time) and

$$\tilde{t} = \frac{t}{t_r}, \quad \tilde{\rho} = \frac{\rho}{\rho_r}, \quad \tilde{\underline{u}} = \frac{\underline{u}}{u_r}, \quad \tilde{\nabla} = l_r \nabla, \quad \tilde{E} = \frac{E}{\rho_r u_r^2}, \quad \tilde{p} = \frac{p}{\rho_r u_r^2}, \quad \tilde{\underline{F}} = \underline{F} \frac{l_r}{\rho_r u_r^2}. \quad (18)$$

Consequently, the dimensionless Euler equations do not change under unit transformation. If not stated otherwise we drop the tilde sign ($\tilde{\cdot}$) and assume always dimensionless quantities from now on.

Choice of parameters One consequence of dimensionless units is the free choice of parameters. We want to use this feature in choose a sensible set of parameters. Considering the Euler equations in conservative form, eqn. (4), their functions of space and time

$$\rho = \rho(t, x, y, z), \quad (\rho \underline{u}) = (\rho \underline{u})(t, x, y, z), \quad E = E(t, x, y, z) \quad (19)$$

are completed with

$$\gamma := 5/3, \quad R := 1, \langle \rho \rangle := 1, \langle c \rangle := 1, \quad (20)$$

From that we derive

$$C_P = \frac{c_0^2}{\gamma} = 3/5 = 0.6, \quad \langle p \rangle = C_P \cdot \langle \rho \rangle = 0.6, \quad \langle E \rangle = \frac{\langle p \rangle}{\gamma - 1} = 0.9, \quad \langle T \rangle = \frac{\langle c \rangle^2}{\gamma R} = 0.6, \quad (21)$$

where $\langle \cdot \rangle$ is the *volume-weighted average* or *mean value* over the domain Ω

$$\langle q \rangle = \frac{\int_{\Omega} q d\Omega}{\int_{\Omega} d\Omega} \quad (22)$$

If not state otherwise, these set of parameters define the global state at all times.

2.1.2 Weak Formulation

A natural way to define a generalized solution of the Euler equations that does not require differentiability is, to go back to the integral form of the conservation law,

The basic idea is to take the PDE, multiply it by a smooth *test function*, integrate one or more times over some domain, and then use integration by parts to move derivatives off the function q and onto the smooth test function. The result is an equation involving fewer derivatives on q , and hence requiring less smoothness.

In this section we want to derive the *weak formulation* of the governing equations. This establishes the basis for the polynomial formulation which is the core idea of all DG methods. First, the Euler equations get split up into terms resembling the independent one temporal and three spatial dimensions with respect to the linear differential operator.

$$\partial_t \underline{U} + \partial_x \underline{F}(\underline{U}) + \partial_y \underline{G}(\underline{U}) + \partial_z \underline{H}(\underline{U}) + \underline{S} = 0, \quad (23)$$

where

$$\underline{U} = (\rho, \rho u_1, \rho u_2, \rho u_3, E)^T \quad (24)$$

$$\underline{F}(\underline{U}) = (\rho u_1, \rho u_1^2 + p, \rho u_1 u_2, \rho u_1 u_3, u_1(E + p))^T \quad (25)$$

$$\underline{G}(\underline{U}) = (\rho u_2, \rho u_2 u_1, \rho u_2^2 + p, \rho u_2 u_3, u_2(E + p))^T \quad (26)$$

$$\underline{H}(\underline{U}) = (\rho u_3, \rho u_3 u_1, \rho u_3 u_2, \rho u_3^2 + p, u_3(E + p))^T \quad (27)$$

$$\underline{S} = (0, -f_1, -f_2, -f_z, 0)^T \quad (28)$$

Defining a vector-valued test function $\phi = (0, \dots, 0, \phi_i, 0, \dots, 0)^T$ ($i \in 1, \dots, 5$), multiplying component-wise with above equation and integrating over the domain Ω we get

$$\int_{\Omega} \left(\partial_t U_i \phi^i + \partial_x F_i(\underline{U}) \phi^i + \partial_y G_i(\underline{U}) \phi^i + \partial_z H_i(\underline{U}) \phi^i + S_i \phi^i \right) d^3x = 0 \quad (29)$$

Integration-by-parts rearranges the integral into a *source term*, *volume term* and *surface term*.

$$\int_{\Omega} \partial_t U_i \psi(x, y, z)^i d^3x + \int_{\Omega} S_i \psi(x, y, z)^i d^3x = \quad (30)$$

$$\begin{aligned} & \int_{\partial\Omega} \left(F_i(\underline{U}) \psi(x, y, z)^i n_x + G_i(\underline{U}) \psi(x, y, z)^i n_z + H_i(\underline{U}) \psi(x, y, z)^i n_z \right) d^2x, \\ & - \int_{\Omega} \left(F_i(\underline{U}) \partial_x \psi(x, y, z)^i + G_i(\underline{U}) \partial_y \psi(x, y, z)^i + H_i(\underline{U}) \partial_z \psi(x, y, z)^i \right) d^3x \end{aligned} \quad (31)$$

where $\underline{n} = (n_x, n_y, n_z)^T$ is the outward surface normal to $\partial\Omega$.

Unfortunately, weak solutions are not unique, and so an additional problem is to identify which weak solution is the physically correct vanishing-viscosity solution. Again, one would like to avoid working with the viscous equation directly, but it turns out that there are other conditions one can impose on weak solutions that are easier to check and will also pick out the correct solution. These are called *entropy conditions* by analogy with the gas dynamics case, where a discontinuity is physically realistic only if the entropy of the gas increases as it crosses the shock.

2.2 Turbulence Statistics

Turbulences are a common phenomena in nature. It names a flow regime in fluid dynamics characterized by chaotic changes in pressure and flow velocity and contrasts *laminar* flow, which occurs when a fluid flows in parallel layers, with no disruption between them.

In astrophysics turbulence is suspected to play a major role in star formation within interstellar media (ISM). Hence, a theoretical understanding of the underlying mechanics is crucial in order correctly model turbulences in numerical simulations. While the problem for incompressible media has been thoroughly studied in the past (Kolmogorov scaling), the additional dynamics introduced by compressibility are still an active field of research.

It is important not to forget that real ISM turbulence is neither isothermal, nor polytropic. The real ISM has a local temperature that is not a simple function of density but results from the evolution of the thermal energy. Gravity takes over control over the most dense regions. Magnetic fields lead to effects like frozen filaments called *Alfvén's frozen in theorem*.

Reynolds Number An important dimensionless quantity in fluid mechanics used to help predict flow patterns in different fluid flow situations is the *Reynolds number* Re . It is the ratio between *inertial* and *viscous* forces and can be calculated by

$$Re = \frac{\text{inertial force}}{\text{viscous force}} = \frac{\rho_0 u_0 L}{\mu_{vis}}, \quad (32)$$

where ρ_0 , u_0 , L , μ_{vis} are the fluid density, fluid velocity, characteristic length and dynamic viscosity, respectively. Since the governing equations are defined to be the inviscid Euler equations (cf. section 2.1), in theory, no viscosity is anywhere present.

$$\mu_{vis} \equiv 0 \implies Re = \infty \quad (33)$$

This has far-reaching consequences because the Reynolds number predicts the transition from *laminar* to *turbulent* flow. Laminar flow becomes unstable when the inertial forces dominate

over the viscous forces: $Re \gg 1$. Consequently, the Euler equations lead to turbulence over all time and length scales. We come back to this issue later.

Reynolds Decomposition Turbulences is characterized by a random fluctuation of flow variables in time measured at a fixed point in space. Their time evolution can be described with the *Reynolds decomposition*.

$$q(t) = \langle q \rangle_t + \tilde{q}(t), \quad \langle \tilde{q} \rangle \equiv 0 \quad (34)$$

where

$$\langle q \rangle_t = \frac{\int_{t_0}^{t_1} q(t) dt}{t_1 - t_0} \quad (35)$$

is the time-average of the flow property $q(t)$. A turbulent flow can be globally described in terms of the mean values of the flow properties like density, velocity and pressure.

As the Reynolds decomposition suggests it does not make sense to talk about a turbulence at one specific point in time or space. Thus, we define important *mean values* which help to quantify the current state of the turbulent system as a whole.

Root-mean-square Velocity u_{rms} is the mass-weighted average squared velocity of the system.

$$u_{\text{rms}} = \sqrt{\frac{\int_{\Omega} \rho \underline{u}^2 d\Omega}{\int_{\Omega} \rho d\Omega}} \quad (36)$$

Remark We introduced the mass-weighting since the conducted turbulence simulations are highly compressible.

Sonic Mach Number \mathcal{M} is directly related to u_{rms} via

$$\mathcal{M} = \frac{u_{\text{rms}}}{\langle c \rangle}. \quad (37)$$

From $\langle c \rangle = 1$, according to eqn. (??), follows that Mach number and root-mean-square velocity are equivalent and, if not stated otherwise, used interchangeably.

Turning time T_{turn} Under section 2.1 are characteristic time scale was introduced $t_r = \frac{l_r}{u_r}$ in order to proof that the Euler equations are invariant under unit transformations. Now it gets a practical meaning since we define a *turning time* that tells how long it takes for a turbulence vortex to traverse the total length L of the physical domain. The turning time is also often called *crossing time* or *dynamic time scale*.

$$T_{\text{turn}} = \frac{L}{u_{\text{rms}}} \quad (38)$$

Dynamic time t_d is the scaled *physical* time t .

$$t_d = \frac{t}{T_{\text{turn}}} \quad (39)$$

All time evolution plots presented in this thesis are scaled to dynamic time which sets the time axis in relation to the observed turbulence.

Bulk Motion $\underline{\mathcal{P}}$ is the integrated momentum over the whole domain.

$$\underline{\mathcal{P}} = \int_{\Omega} \rho \underline{u} d\Omega \quad (40)$$

When discussing about turbulences in closed systems say a periodic box (three-dimensional torus) present bulk motion distorts the measurement of mean values like the sonic Mach number. In the worst case scenario the whole fluid would coherently move in one direction pretending to be a turbulence of certain Mach number. Correction of bulk motion is an important subject in turbulence simulations.

Total Kinetic Energy \mathcal{K} is important in the discussion of energy accounting, especially in closed systems where the total energy $E = \mathcal{I} + \mathcal{K}$ is conserved.

$$\mathcal{K} = \int_{\Omega} \frac{\rho}{2} \underline{u}^2 d\Omega \quad (41)$$

Mean Kinetic Energy Dissipation ϵ is the negated time derivative of the total kinetic energy \mathcal{K}

$$\epsilon = -\frac{d\mathcal{K}}{dt} \quad (42)$$

It is a measure of how dissipative a numerical scheme is at a specific point in time. The next section explains this further.

2.2.1 Energy Cascade & Powerspectrum

Turbulent flows always show a three-dimensional character and they lead to rotational flow structures, called turbulent eddies, with a wide range of length and energy scales.

Fluid particles which were initially separated by a large distance can be brought close together by eddying motions. Consequently, mass, heat and momentum are very effectively exchanged. It is an established fact that this property has a profound influence in birth of stars, solar systems and cosmic structures.

The largest turbulent eddies interact and transfer energy from the mean flow. Since large eddies are of the same order as the characteristic length and velocity scale the flow is inviscid and their dynamics are dominated by inertial effects. They transfer kinetic energy down to smaller eddies via *vortex stretching*. This way an energy cascade emerges from largest to smallest scales.

The spatial wavenumber k of an eddy of wavelength λ is defined as

$$k = \frac{2\pi}{\lambda} \quad (43)$$

and the *spectral energy* $E(k)$ is a function of k .

see Kolmogorov-Burgers Model for Star-forming Turbulence - inertial range -> Kolmogorov scaling since large scales - dissipative range -> Burgers scale - theoretical ground

see SCALING RELATIONS OF SUPERSONIC TURBULENCE IN STAR-FORMING MOLECULAR CLOUDS - numerical validation of theory in above paper

Kolmogorov length scales Smallest scales in the spectrum that form the viscous sub-layer range. In this range, the energy input from nonlinear interactions and the energy drain from viscous dissipation are in exact balance. The small scales have high frequency, causing turbulence to be locally isotropic and homogeneous.

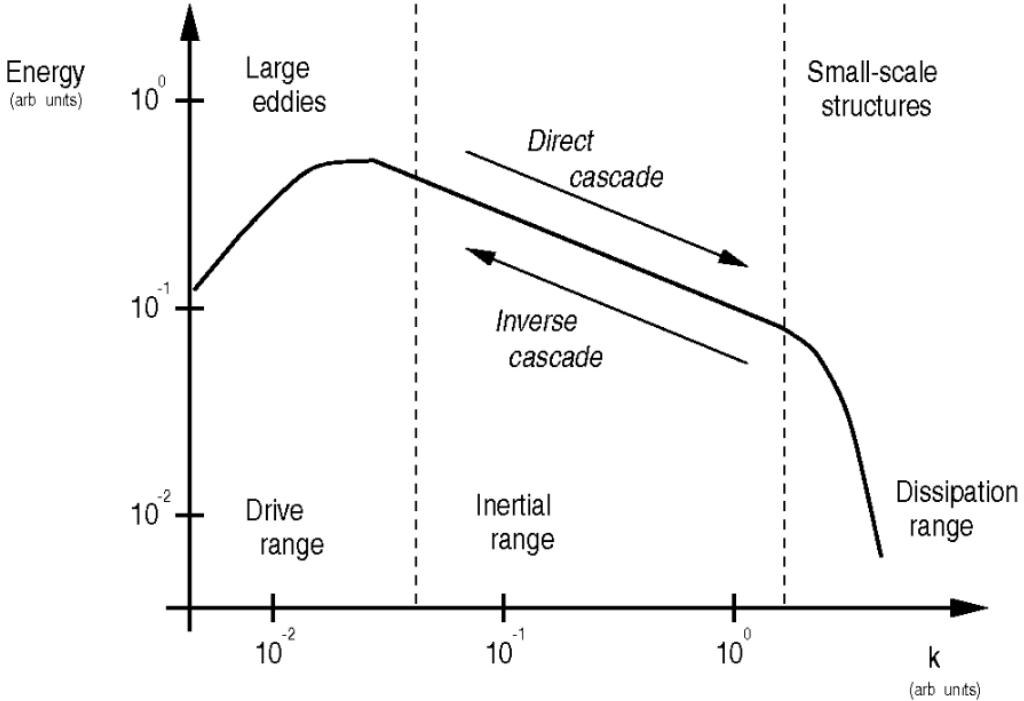


Figure 1: Schematic representation of the K41 picture of turbulence showing the spatial energy spectrum as an example. Source: [?], p6

The smallest length scales η of motion are dominated by viscous effects where their Reynolds number is equal to 1. $Re_\eta = u\eta/\mu = 1$, so inertia and viscous effects are of equal strength. The kinetic energy gets dissipated and converted into thermal energy. These so called KOLMOGOROV *microscales* are therefore a measure for the spatial resolution characteristics and the diffusive order of numerical schemes of equal h-refinement.

Grid point requirement

$$N \propto Re^{9/4} \quad (44)$$

can be relaxed since most dissipation takes place 5 to 15 times the Kolmogorov length scale η . See Moin and Mahesh, 1998

Velocity Powerspectrum The velocity power spectrum is calculated as follows. For each velocity component we take the Fourier transform of the velocity field $\underline{u} = (u_1, u_2, u_3)^T$. We denote these Fourier transforms as $\hat{\underline{u}} = (\hat{u}_1, \hat{u}_2, \hat{u}_3)^T$. Using these definitions the volume-weighted velocity power spectrum is defined as

$$P(\underline{k}) = \frac{1}{2} \hat{\underline{u}} \cdot \hat{\underline{u}}^\dagger, \quad (45)$$

where $\hat{\underline{u}}^\dagger = (\hat{u}_1^\dagger, \hat{u}_2^\dagger, \hat{u}_3^\dagger)^T$ is the complex conjugate of the transformed velocity and $\underline{k} = (k_1, k_2, k_3)^T$ is the spatial wave number vector. Taking the three-dimensional shell-average over $P(\underline{k})$ we get the familiar log-log-scale powerspectrum shown in fig. 1. It reads

$$\tilde{P}(k) dk = 4\pi k^2 \frac{1}{2} \hat{\underline{u}} \cdot \hat{\underline{u}}^\dagger dk, \quad (46)$$

where the pre-factor $4\pi k^2$ is a contribution from the differential volume of a thin sphere at radius $k = |\underline{k}| \geq 0$.

Kinetic Energy Powerspectrum The area under the shell-averaged transformed kinetic energy powerspectrum

$$\tilde{E}(k) = \frac{1}{V_\Omega} \int_0^\infty 4\pi k^2 \hat{\mathcal{K}} \cdot \hat{\mathcal{K}}^\dagger dk \quad (47)$$

is the total of the squared kinetic energy \mathcal{K} within the system in accordance with PARSEVAL's theorem

$$\int_{-\infty}^\infty |Y(x)|^2 dx = \frac{1}{2\pi} \int_{-\infty}^\infty |\hat{Y}(k)|^2 dk. \quad (48)$$

2.2.2 Probability Distribution Functions (PDF)

Using all the cells in our grid, we obtain the cumulative distributions, F , of the following quantities: logarithm of the density, the three velocity components v_i with $i = 1, 2, 3$, the logarithm of the trace free rate of strain, $|S|$, and the logarithm of vorticity, ω . To obtain these distributions, the corresponding quantities are binned linearly.

The share of a density range from ρ_A to ρ_B can then be calculated by

$$\Pr[\rho_A \leq \rho' \leq \rho_B] = \int_{\rho_A}^{\rho_B} \text{PDF}[\rho](\rho') d\rho' \quad (49)$$

According to various studies ... fully developed turbulence models yield a nearly log-normal density distribution.

The PDF is assumed to follow a lognormal analytical form

$$P_s(s) = \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left[-\frac{(s-s_0)^2}{2\sigma_s^2}\right], \quad (50)$$

where $s = \ln(\rho/\rho_0)$. Via the standard deviation σ_s of the distribution one can derive the mean sonic Mach number of the turbulence via this relation

$$(2\sigma_s)^2 = \ln(1 + b^2 \mathcal{M}^2) \quad (51)$$

Reordering yields an explicit expression for the mach number

$$\mathcal{M}_{PDF} = \frac{\sqrt{\exp((2\sigma_s)^2) - 1}}{b} \quad (52)$$

The proportionality constant b is dependent on the ratio of compressible to solenoidal forcing $\zeta \in [0, 1]$, see ..., and the number of spatial dimensions $D = 1, 2, 3$.

$$b = 1 + (D^{-1} - 1)\zeta \quad (53)$$

2.3 Supersonic Shocks

Shocks, or in more technical terms singular compression waves, are escalating highly localized spikes in density and pressure due to nonlinear dynamics encoded in the Euler equation. When a shock wave is emerging the velocity behind the wave front is higher than in front of it. The media gets highly compressed until an unphysical state is reached. The velocity characteristic begin to cross. Nature solves this dilemma by introducing additional physics like extreme heat radiation, explosions, bangs or detachment of media in case of surface waves. Either way, it involves an entropy increase in the system. A numerical solver has to capture this kind of physics in order to prevent unphysical solutions.

Astrophysical flows often involve shock waves. Thus, it is important that shocks are accurately described by the numerical code used for the simulation of the flow. The respective requirements on the code fall into two different categories. In the first category of problems the structure of the shock front itself is important, i.e., a high spatial resolution of the discontinuity is crucial. Typical examples for this category of problems are hydrodynamic flows with nuclear burning, where an insufficient spatial resolution can lead to quantitatively very inaccurate and in some cases to even qualitatively incorrect solutions (see Sect. 4). In the second category of problems the time scales of processes triggered by the shock wave are comparable or larger than the hydrodynamic time scale. Then mainly an accurate description of the two states on both sides of the shock is important, while the structure of the discontinuity matters less.

Solving Riemann problem. Due to the nonlinearity of the Euler equations

2.3.1 Shock Formation

For larger t the equation C.10) may not have a unique solution. This happens when the characteristics cross, as will eventually happen if $q_x(x,0)$ is negative at any point. At the time T_b where the characteristics first cross, the function $q(x, t)$ has an infinite slope — the wave "breaks" and a shock forms. Beyond this point there is no classical solution of the PDE, and the weak solution we wish to determine becomes discontinuous.

$$u_{\text{shock}} = \frac{u_L + u_R}{2} \quad (54)$$

is the shock speed, the speed at which the discontinuity travels.

When we set following initial condition:

$$U(0, x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } 0 \leq x < 1 \\ 2 & \text{if } 1 \leq x < 2 \\ 0 & \text{if } x > 2 \end{cases} \quad (55)$$

2.3.2 Method of Characteristics

The prototype for all ODEs of second order is the BURGER's equation. This is about the simplest model that includes the nonlinear and viscous effects of fluid dynamics.

$$q_t + q q_x = \epsilon q_{xx} \quad (56)$$

We set $\epsilon = 0$.

The characteristics satisfy

$$x'(t) = q(t, x(t)) \quad (57)$$

and along each characteristic q is constant, since

$$\frac{d}{dt}q(t, x(t)) = \partial_t q(t, x(t)) + \partial_x q(t, x(t))x'(t) \quad (58)$$

$$= q_t + q q_t \quad (59)$$

$$= 0. \quad (60)$$

Moreover, since q is constant on each characteristic, the slope $x'(t)$ is constant by eqn. (56) and so the characteristics are straight lines, determined by the initial data.

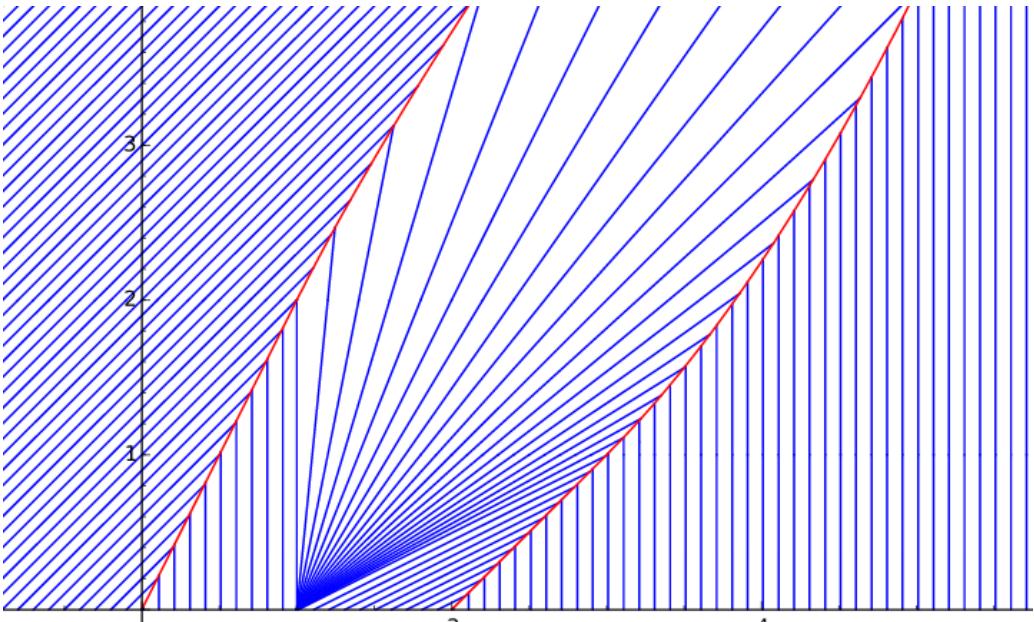


Figure 2: Source: <https://calculus7.org/2015/11/27/rarefaction-colliding-with-two-shocks/>

The picture above shows multiple interesting features of shocks.

For times beyond the breaking time some of the characteristics have crossed and so there are points x where there are three characteristics leading back to $t = 0$. One can view the "solution" q at such a time as a triple-valued function. However, the density of a gas cannot possibly be triple valued at a point.

The equation C.6) is a model of C.7) valid only for small ϵ and smooth q . When it breaks down, we must return to C.7). If the initial data is smooth and ϵ very small, then before the wave begins to break the ϵq_{xx} term is negligible compared to the other terms and the solutions to the two PDEs look nearly identical. However, as the wave begins to break, the second derivative term q_{xx} grows much faster than q_x , and at some point the ϵq_{xx} term is comparable to the other terms and begins to play a role. This term keeps the solution smooth for all time, preventing the breakdown of solutions that occurs for the hyperbolic problem. This smooth

solution has a steep transition zone where the viscous term is important. As $\epsilon \rightarrow 0$ this zone becomes sharper and approaches the discontinuous solution known as a shock. It is this vanishing-viscosity solution that we hope to capture by solving the inviscid equation.

Rarification wave is pulling both shocks together so they eventually collide

The speed of propagation can be determined by conservation. The relation between the shock speed s and the states q_i and q_r is called the Rankine-Hugoniot jump condition. For scalar problems:

$$u_{\text{shock}} (u_L - u_R) = f(u_L) - f(u_R) \quad (61)$$

For systems of equations, $q_i - q_r$ and $f(q_r) - f(q_i)$ are both vectors while s is still a scalar. Now we cannot always solve for s to make C.18 hold. Instead, only certain jumps $q_i - q_r$ are allowed, namely those for which the vectors $f(q_i) - f(q_r)$ and $q_i - q_r$ are linearly dependent. For a linear system with $f(q) = Aq$, C.18 gives

$A(q_i - q_r) = s(q_t - q_r)$, C.20 i.e., $q_i - q_r$ must be an eigenvector of the matrix A and s is the associated eigenvalue. For a linear system, these eigenvalues are the characteristic speeds of the system. Thus discontinuities can propagate only along characteristics, just as for the scalar advection equation.

2.3.3 Entropy Increase

The second law of thermodynamics requires that entropy must increase across a normal shock wave.

$$\Delta s = c_v \ln \left[\frac{p_2}{p_1} \left(\frac{\rho_1}{\rho_2} \right) \right] \quad (62)$$

This discussion is important on defining Flux functions.

2.4 Finite Element Scheme

Solving PDEs numerically means to discretize an original continuous problem. Most approaches consist of four major generally independent parts.

Meshing Divide the problem domain into adjacent self-contained sub-domains, called elements or cells. Depending on the scheme and requirements this step can happen periodically. Via *Adaptive Mesh Refinement* small scale phenomena within the simulation can be resolved where needed without degrading the overall performance disproportionately.

Reconstruction Approximate the exact solution in every element by a piecewise constant function or polynomial of order N_p .

Evolution Based on the current set of variables the conservation laws yield a new state which gets evolved one timestep into the future.

Averaging/Propagation Flux functions solve the RIEMANN problem and communicate the lately acquired state across boundaries and propagate the new information throughout the cell.

The last three actions are commonly grouped together under the term *REA algorithm* which is typical for GODUNOV-type methods.

Since following terms are mentioned on a regular basis throughout this document we give brief definitions.

Mesh A mesh consists of either cells or elements. The mesh can be structured or unstructured. It contains the necessary information where to find cells/elements and what their (spatial) relationship to neighbors are.

Grid Regular/Irregular, grid spaces, array of points/nodes.

Cell The atomic container type of a grid. They contain the actual data which can be a scalar, arrays of scalars, vectors, tensors, etc. What cells distinguish from points is that they have an expanse. Hence, one must specify if the data is defined in the cell-center, at their corners or at their faces.

Element Elements are spatially extended objects like cells. However, they group a list of points called *nodes* on which the data is pinned on. When an element interacts with the outside world it must extract the necessary values from these nodes via polynomial interpolation.

2.4.1 Galerkin Method

There are three distinct streams of numerical solution techniques: finite difference, finite element and spectral methods. Finite Volume Methods are a specialization of finite difference methods whereas DG methods belong to the family of finite element methods.

A numerical solving algorithm consists of three parts:

- Integration of the governing equations of fluid flow over the control volume
- Discretization of the integral into a system of algebraic equations
- iterative time stepping method

Advantage of polynomial methods: exact interpolation -> arbitrary high information density ... space savings

Since discontinuous Galerkin (DG) methods assume discontinuous approximate solutions, they can be considered as generalizations of finite volume methods.

Owing to their finite element nature, the DG methods have the following main advantages over classical finite volume and finite difference methods: — The actual order of accuracy of DG methods solely depends on the exact solution; DG methods of arbitrarily high formal order of accuracy can be obtained by suitably choosing the degree of the approximating polynomials. — DG methods are highly parallelizable. Since the elements are discontinuous, the mass matrix is block diagonal and since the size of the blocks is equal to the number of degrees of freedom inside the corresponding elements, the blocks can be inverted by hand (or by using a symbolic manipulator) once and for all. — DG methods are very well suited to handling complicated geometries and require an extremely simple treatment of the boundary conditions in order to achieve uniformly high-order accuracy. — DG methods can easily handle adaptivity strategies

since refinement or unrefinement of the grid can be achieved without taking into account the continuity restrictions typical of conforming finite element methods. Moreover, the degree of the approximating polynomial can be easily changed from one element to the other. Adaptivity is of particular importance in hyperbolic problems given the complexity of the structure of the discontinuities.

More information can be found in Bernardo Cockburn George E. Karniadakis Chi-Wang Shu(Eds.) Discontinuous Galerkin Methods Theory, Computation and Applications

They provide fast convergence, small diffusion and dispersion errors, easier implementation of the inf-sup condition for incompressible Navier-Stokes, better data volume-over-surface ratio for efficient parallel processing, and better input/output handling due to the smaller volume of data.

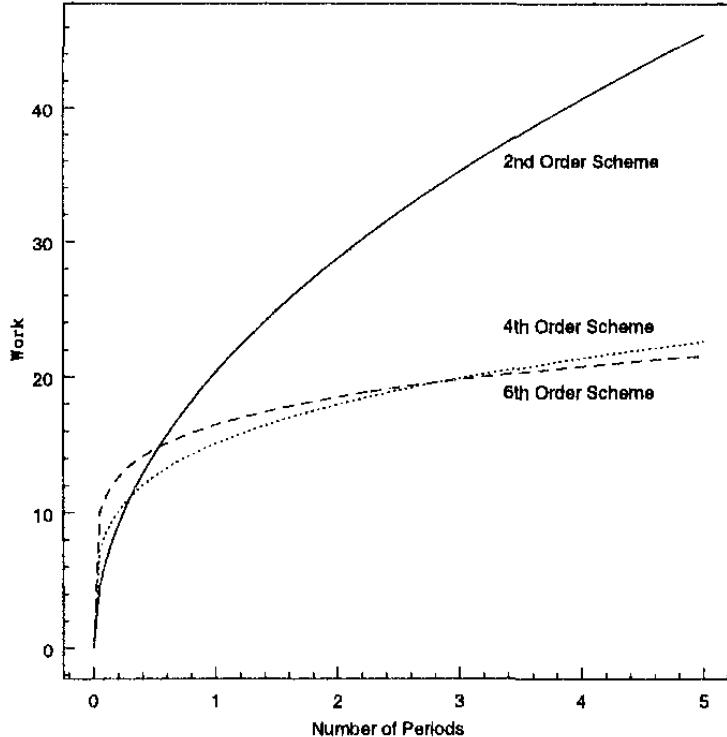


Figure 3: Computational work (number of floating-point operations) required to integrate a linear advection equation for M periods while maintaining a cumulative phase error of $e = 10\%$. Source:

[], p.10

hp Convergence The mathematical theory of finite elements in the 1970s has established rigorously the convergence of the h-version of the finite element. The error in the numerical solution decays algebraically by refining the mesh, that is, introducing more elements while keeping the (low) order of the interpolating polynomial fixed. An alternative approach is to keep the number of subdomains fixed and increase the order of the interpolating polynomials in order to reduce the error in the numerical solution. This is called p-type refinement and is typical of polynomial spectral methods [101]. For infinitely smooth solutions p-refinement usually leads to an exponential decay of the numerical error.

Source Spectral/hp Element Methods for CFD 1999

Briefly, we are going to take a closer look at the class of *Finite Elements* approach where the

Finite Volume and the *Discontinuous Galerkin* are concrete implementations are. At first the very physical domain Ω is divided into a *mesh* of adjunct self-contained sub-domains Ω_l ($l \in \mathbb{N}$) with concisely defined boundaries. For the rest of this text we omit the element index l . So Ω is now the domain within an element. The unknown solution \underline{U} is replaced by polynomials of order N_p constructed from linear combinations of orthogonal basis functions Ψ^j .

$$U_{\underline{i}}(t, x, y, z) \approx p_{\underline{i}}(t, x, y, z) = \sum_{j=0}^{N_p} U_{\underline{i}}^j(t) \Psi^j(x, y, z), \quad (63)$$

where \underline{i} and \underline{j} are three-dimensional *multi-indices*.

$$\underline{i} = (i, j, k)^T, \quad i, j, k \in \mathbb{N}_0 \quad (64)$$

Remark Usually, the polynomials of every element are transformed to a reference space $\hat{\Omega} = [-1, 1]^3$ where the actual interpolation takes place. This measure massively increases efficiency since the basis functions are equal among all elements. For the sake of simplicity this step is omitted here.

The following treatment is analog for all five conservative variables of the Euler equation hence we ignore the index i . Remembering the general weak formulation, eqn. (30), of the solution integral,

$$\begin{aligned} & \int_{\Omega} \left(\sum_{j=0}^{N_p} (\partial_t U^j(t)) \Psi^j(\underline{x}) \right) \phi(\underline{x}) d^3x + \int_{\Omega} S(t) \phi(\underline{x}) d^3x = \\ & \int_{\partial\Omega} [F(t) \phi(\underline{x}) n_x + G(t) \phi(\underline{x}) n_z + H(t) \phi(\underline{x}) n_z] d^2x, \\ & - \int_{\Omega} \left[\left(\sum_{j=0}^{N_p} F^j(t) \Psi^j(\underline{x}) \right) \partial_x \phi(\underline{x}) + \left(\sum_{j=0}^{N_p} G^j(t) \Psi^j(\underline{x}) \right) \partial_y \phi(\underline{x}) + \left(\sum_{j=0}^{N_p} H^j(t) \Psi^j(\underline{x}) \right) \partial_z \phi(\underline{x}) \right] d^3x, \end{aligned} \quad (65)$$

we put the time variable into the coefficients S, F, G, H and the space variables into the basis functions Ψ and test functions ϕ .

Lagrange Polynome If we associate the basis functions $\Psi^{\underline{l}} := L^{\underline{l}}$ and the test functions $\phi := L^{\underline{l}}$ with LAGRANGE polynomials of equal order N_p , we can formulate an interpolation and integration scheme (*collocation*) over the domain Ω . The polynomial in one dimension reads as follows

$$l_j(x) = \prod_{k=0, k \neq j}^p \frac{x - x_k}{x_j - x_k}, \quad j = 0, \dots, p, \quad (66)$$

with the KRONECKER property $l_j(x_i) = \delta_{ij}$.

For illustration purposes we begin with the simplest case: the one-dimensional LAGRANGE interpolation.

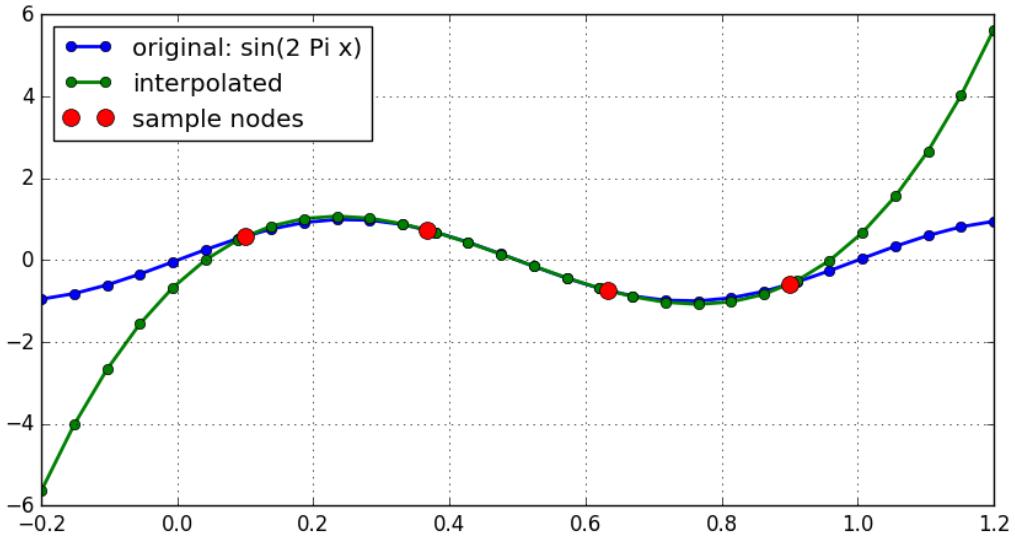


Figure 4: One-dimensinal Lagrange interpolation with four sample nodes.

The Lagrange polynome of third order needs four anchor nodes with their associated values in order to continuously interpolate all values in between. Unfortunately, Lagrange polynomes go to infinity going further away from the outer anchor nodes. This effect occurs in the Gauss-Quadrature where the outermost anchor nodes do not lie on the interval boundaries.

One gets to three-dimensional formulation via the *Tensor Product Ansatz*.

$$L_{\underline{i}}(\underline{x}) = L_{ijk}(x, y, z) = l_i(x) \cdot l_j(y) \cdot l_k(z) \quad (67)$$

We write the thee-dimensional polynome as:

$$P^{\underline{i}}(t, \underline{x}) = \sum_{\underline{i}=0}^{N_p} F_{\underline{i}}(t) L_{\underline{i}}(\underline{x}) = \sum_{i,j,k=0}^{N_p} f_{ijk}(t) \cdot l_i(x) \cdot l_j(y) \cdot l_k(z) \quad (68)$$

Note, that the time varying part of the polynome lives exclusively in the coefficients.

Galerkin Method Replacing Ψ^j and ϕ accordingly, we get an explicit ansatz, called *Galerkin* method.

$$\begin{aligned} & \int_{\Omega} \left(\sum_{j=0}^{N_p} \dot{U}_j(t) L_j(\underline{x}) \right) L_{\underline{i}}(\underline{x}) d^3x + \int_{\Omega} S(t, \underline{x}) L_{\underline{i}}(\underline{x}) d^3x = \\ & \int_{\partial\Omega} \left[F(t) n_x(\underline{x}) + G(t) n_y(\underline{x}) + H(t) n_z(\underline{x}) \right] L_{\underline{i}}(\underline{x}) d^2x \\ & - \int_{\Omega} \left[\left(\sum_{j=0}^{N_p} F_j(t) L_j(\underline{x}) \right) \partial_x L_{\underline{i}}(\underline{x}) + \left(\sum_{j=0}^{N_p} H_j(t) L_j(\underline{x}) \right) \partial_y L_{\underline{i}}(\underline{x}) + \left(\sum_{j=0}^{N_p} G_j(t) L_j(\underline{x}) \right) \partial_z L_{\underline{i}}(\underline{x}) \right] d^3x \end{aligned} \quad (69)$$

Evaluating above formula at discrete nodes $\underline{x}_{\underline{i}}$ and introducing *integration weights*

$$\omega_{\underline{i}} = \int_{\Omega} L_{\underline{i}}(\underline{x}) d^3x \quad (70)$$

we can discretize the continuous integrals.

$$\begin{aligned} & \sum_{\underline{k}=0}^{N_p} \left(\sum_{j=0}^{N_p} \dot{U}_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} + \sum_{\underline{k}=0}^{N_p} S(t, \underline{x}_{\underline{k}}) L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} = \\ & \left[F^*(t) + G^*(t) + H^*(t) \right] L_{\underline{i}}(\underline{x}) \\ & - \sum_{\underline{k}=0}^{N_p} \left[\left(\sum_{j=0}^{N_p} F_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}^{(x)}(\underline{x}_{\underline{k}}) + \left(\sum_{j=0}^{N_p} H_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}^{(y)}(\underline{x}_{\underline{k}}) + \left(\sum_{j=0}^{N_p} G_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}^{(z)}(\underline{x}_{\underline{k}}) \right] \omega_{\underline{k}} \end{aligned} \quad (71)$$

The partial differential ∂_x turns into a discrete linear operator.

$$L_{\underline{j}}^{(x)}(\underline{x}) = \partial_x L_{\underline{j}}(\underline{x}) = (\partial_x l_{j_1}(x)) \cdot l_{j_2}(y) \cdot l_{j_3}(z) \quad (72)$$

$$D_{\underline{i}\underline{j}}^{(x)} = D_{i_1 i_2 i_3 j_1 j_2 j_3}^{(x)} = l_{j_1}^{(x)}(x_{i_1}) \cdot l_{j_2}^{(x)}(y_{i_2}) \cdot l_{j_3}^{(x)}(z_{i_3}) \quad (73)$$

The operators for the y- and z-dimension are constructed in analog manner.

The surface term get replaced by flux functions where an exchange of mass, momentum and energy between element boundaries takes place. Following common tradition they get marked with a star: F^* .

Finally, we get to a semi-discrete weak formulation of the discontinuous Galerkin method.

$$\begin{aligned} & \sum_{\underline{k}=0}^{N_p} \left(\sum_{j=0}^{N_p} \dot{U}_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} + \sum_{\underline{k}=0}^{N_p} S(t, \underline{x}_{\underline{k}}) L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} = \\ & \left[F^*(t) + G^*(t) + H^*(t) \right] L_{\underline{i}}(\underline{x}) \\ & - \sum_{\underline{k}=0}^{N_p} \left[\left(\sum_{j=0}^{N_p} F_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) D_{\underline{k}\underline{i}}^{(x)} + \left(\sum_{j=0}^{N_p} H_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) D_{\underline{k}\underline{i}}^{(y)} + \left(\sum_{j=0}^{N_p} G_{\underline{j}}(t) L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) D_{\underline{k}\underline{i}}^{(z)} \right] \omega_{\underline{k}} \end{aligned} \quad (74)$$

Remark If the polynomial order is set to one $N_p = 1$ the formulation reduces to the first order FV method.

2.4.2 Flux Functions

Defining consittent, accurate and stable flux functions is a huge numerical discipline of itself and an active field of research.

Riemann Problem Many of the methods are based on solving the RIEMANN problem between the states q_L and q_R in order to define the numerical flux $F^*(q_L, q_R)$. For one way to approach this, it is useful to view the data Q^n at time t_n as defining a piecewise constant function $q^n(t_n, x)$ which has the value Q_i^n for all x in the interval C_i . Suppose we could solve the conservation

law exactly over the time interval $[t_n, t_{n+1}]$ with initial data $q^n(t_n, x)$. We call the resulting function $q_n(t, x)$ for $t_n < t < t_{n+1}$. Then we might consider defining the numerical flux F_i^{*n} by

$$F_i^{*n} = \frac{1}{k} \int_{t_n}^{t_{n+1}} f(q^n(t, x_i)) dt \quad (75)$$

This integral is trivial to compute, at least provided the time step k is small enough, because of the fact that with piecewise constant initial data we can find the exact solution easily by simply piecing together the solutions to each Riemann problem defined by the jump at each interface.

The method above is known as GODUNOV's method, which was already mentioned at the beginning of the section, as an approach to solving the Euler equations of gas dynamics in the presence of shock waves.

A wide variety of approximate Riemann solvers have been proposed that can be applied much more cheaply than the exact Riemann solver and yet give results that in many cases are equally good when used in the Godunov or high-resolution methods. In brief, we will look at some possibilities.

All-Shock Solver In solving a nonlinear Riemann problem we must worry about whether the wave in each family should be a shock or rarefaction so that we know whether to use the Hugoniot locus or integral curve correspondingly. One simplification that can be made to the Riemann solver is to ignore the possibility of rarefaction waves and simply find a Riemann solution in which each pair of states is connected along the Hugoniot locus. The solution then consists entirely of discontinuities that satisfies the Rankine-Hugoniot conditions and is a weak solution of the conservation law. This approach is discussed by Colella [54] for gas dynamics. The all-shock solver is particularly valuable for problems where the Riemann problem is harder to solve, such as in problems where a more complicated equation of state than a gamma-law gas must be used. This occurs at high temperatures, or in relativistic flow, for example.

A potential problem with this approach, of course, is that by using a solution to the Riemann problem that does not satisfy the entropy condition, we might obtain a numerical solution which does not approximate the correct weak solution. Actually, however, in most cases (except for transonic rarefactions), Godunov's method with the Riemann solver will typically work well even if the correct solution involves rarefaction waves. This is because of the numerical dissipation that is introduced in every step of Godunov's method through the averaging process. Although the discontinuous solution used in the solution of a particular Riemann problem may not be correct, by averaging this solution over the grid cell at the end of the time step the discontinuity is smeared out and after many time steps a good approximation to the correct rarefaction wave will be computed. The all-shock approximation will typically be inadequate in the case of a transonic rarefaction. Such a discontinuity tends to persist and results in the computation of an entropy-violating weak solution. An *entropy fix* is often used to eliminate this problem.

HLLE (Harten-Lax-Entropy-fix) flux Another approach is to approximate the full Riemann solution by a single intermediate state bounded by two waves moving at speeds s_1 and s_2 . The wave speeds should be some approximations to the minimum and maximum wave speeds that would arise from this particular Riemann data and the intermediate state can then be calculated by the condition of conservation. This approach is developed by Harten, Lax, and van Leer originally [108] and improved by Einfeldt [78].

PPM Solver

Bouchut5 Solver

Higher-order Flux Regardless of what Riemann solver is used, Godunov's method will be at best first-order accurate on smooth solutions and generally gives very smeared approximations to shock waves or other discontinuities. The key is to use a better representation of the solution, say piecewise linear instead of the piecewise constant representation used in Godunov's method, but to form this reconstruction carefully by paying attention to how the data behaves nearby. In smooth regions the finite-difference approximation to the slope can be used to obtain better accuracy, but near a discontinuity the *slope* computed by subtracting two values of Q and dividing by h may be huge and meaningless. Using it blindly in a difference approximation will introduce oscillations into the numerical solution.

Limiter Function Near a discontinuity we may want to limit this slope, using a value that is smaller in magnitude in order to avoid oscillations. Methods based on this idea are known as slope-limiter methods. This approach was introduced by van Leer in a series of papers [246] through [248], where he developed the MUSCL scheme for nonlinear conservation laws (Monotonic Upstream-centered Scheme for Conservation Laws). The same idea in the context of flux limiting, reducing the magnitude of the numerical flux to avoid oscillations, was introduced in the flux-corrected transport (FCT) algorithms of Boris and Book [37]. We can view this as creating a hybrid algo- algorithm that is second order accurate in smooth regions but which reduces to a more robust first-order algorithm near discontinuities. This idea of hybridiza-hybridization was also used in early work of Harten and Zwas [110]. An enormous va- variety of methods based on these principles have been developed in the past two decades. One of the algorithms of this type that is best known in the astrophysics community is the piecewise parabolic method (PPM) of Woodward and Colella [59], which uses a piecewise quadratic reconstruction, with appropriate limiting.

2.4.3 Time Integration

At the end of section 2.4.1 in eqn. (74) we arrived at a semi-discrete weak formulation of the Euler equations. It resembles an ordinary differential equation of the form

$$\frac{d}{dt}y = f(t, y). \quad (76)$$

Defining initial values $y(t_0) = y_0$ this equation can be numerically solved in the most naive way via the EULER method. Chosing an appropiate timestep Δt we can explicitly integrate from the current state y^n to the future state y^{n+1} .

$$y^{n+1} = y^n + \Delta t \cdot f(t^n, y^n) \quad (77)$$

The Δt -convergence can be improved by introducing higher-order terms. A widely used class of higher-order time integration schemes are the Runge-Kutta methods (RK). The second-order RK, resp. *midpoint* method, reads

$$y^{n+1} = y^n + \Delta t \cdot f \left(t^n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2} \cdot f(t^n, y^n) \right), \quad (78)$$

evaluating the integrand two times per timestep. Adding again one timestep the integration becomes of third order.

$$y^{n+1} = y^n + \frac{1}{6} (k_1 + 4 k_2 + k_3), \quad (79)$$

where

$$k_1 = \Delta t \cdot f(t^n, y^n) \quad (80)$$

$$k_2 = \Delta t \cdot f\left(t^n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2} \cdot f(t^n, y^n)\right) \quad (81)$$

$$k_3 = \Delta t \cdot f(t^n + \Delta t, y^n - k_1 + 2 k_2). \quad (82)$$

The attentive reader might have recognised a pattern. RK-3 is equivalent to the Simpson's rule for integration.

Next to the spatial resolution of the polynomial ansatz, resp. h-p-convergence, we will compare the influence of the time integration in first, second and third order.

TODO: Stability Analysis (?)

Courant-Friedrichs-Lowy condition To keep a numerical algorithm stable the time step has to obey the *Courant-Friedrichs-Lowy condition* (CFL condition) which states that the domain of dependence of q_i^{n+1} of the algorithm at future time t^{n+1} should include the true domain of dependence at time $t = t^n$. Or in other words: nothing is allowed to flow more than 1 grid spacing within one time step. This means quantitatively

$$\Delta t \leq \frac{\Delta x}{u} \quad (83)$$

Given a CFL constant: $0 < C \leq 1$

$$\Delta t = C \cdot \min_x \left(\frac{\Delta x}{u(x)} \right) \quad (84)$$

The CFL condition is a necessary (but not sufficient) condition for the stability of any explicit differencing method.

In a three-dimensional orthogonal domain the timestep reads

$$\Delta t = C \cdot \min_{\underline{r}} \left(\frac{dx}{|v_x(\underline{r})| + c(\underline{r})}, \frac{dy}{|v_y(\underline{r})| + c(\underline{r})}, \frac{dz}{|v_z(\underline{r})| + c(\underline{r})} \right) \quad (85)$$

Remark For supersonic regimes the sound speed c can be neglected.

3 Numerical Prerequisites

3.1 Computational Frameworks

In this section a brief description of the simulation frameworks and what kind of turbulence setup is used. For the simulations two numerical frameworks, FLASH and FLEXI, specialized in computational fluid dynamics are compared.

FLASH4.3

The FLASH code, currently in its 4th version, is a publicly available high performance application code which has evolved into a modular, extensible software system from a collection of unconnected legacy codes. FLASH consists of inter-operable modules that can be combined to generate different applications. The FLASH architecture allows arbitrarily many alternative implementations of its components to co-exist and interchange with each other. A simple and elegant mechanism exists for customization of code functionality without the need to modify the core implementation of the source. A built-in unit test framework combined with regression tests that run nightly on multiple platforms verify the code.

The framework is firmly established within the astrophysics community and represents the trusted template for testing and comparing the young higher-order schemes provided by FLEXI.

FLASHalready consists of the modules necessary for the upcoming turbulence simulations. A complete set of source files and setup calls are provided in the digital appendix.

The demands for the grid/mesh module are minimal. Setting a uniform grid with periodic boundary conditions is sufficient. Like most grid implementations the domain is divided into blocks matching the number of computing units. Unfortunately, the uniform grid module of FLASHis quite unflexible with regard to possible block configurations since they are tightly coupled with the overall grid resolution or block size, respectively.

There is a wide variety of hydrodynamic and magnet-hydrodynamic solvers available, but we will focus on three split schemes which are intentionally constructed for supersonic turbulences: PPM, Bouchut3 and Bouchut5. They were already discussed briefly in section 2.4.2.

FLEXI

Flexi is developed by the team of the Numerics Research Group hosted at the Institute of Aero- and Gasdynamics at the University of Stuttgart. We are interested in efficient and robust numerical methods for applications in scale resolving CFD and we apply these methods to a variety of of large scale physical and industrial problems.

It provides an implementation of discontinuous Galerkin methods and was originally developed for aerospace applications in research and industries. Strong shocks under isothermal turbulence conditions were not part of the agenda in the first place.

In contrast to FLASHa number of necessary functionality was non-existent and had to be implemented beforehand. Consequently, missing modules were either ported from FLASHand adapted accordingly to fit into FLEXI's infrastructure or written from scratch. Following additions have been made.

polytropic cooling (see section 2.1 *Equation of State*) Trivial to implement.

bulk motion correction (see section 2.1 *Mean Values / Bulk Motion*)

turbulent forcing (see section 3.2)

shock capturing (see section 3.4)

Furthermore analysis and transformation tools were developed since checkpoint files had to be transferred between incompatible frameworks. See section 3.4/Transferring Data.

3.2 Turbulent Forcing

In equation eqn. (4) a *source* respectively *forcing* term was introduced into the Euler equations. Perpetually, at each time step a varying force field $\underline{F}(t, x, y, z)$ in time and space injects kinetic energy over the whole physical domain. Inducing natural looking turbulence with the desired properties is not a trivial task. One commonly used method is by exploiting the intermittent behaviour of random walks, in particular the ORNSTEIN-UHLENBECK PROCESS. Since we deploy pseudo-random numbers the forcing is replicable.

Based on [?] we formulate

$$\hat{d}\mathbf{f}(\mathbf{k}, t) = \frac{3}{\sqrt{1 - 2\zeta + 3\zeta^2}} \left[-\hat{\mathbf{f}}(\mathbf{k}, t) \frac{dt}{T} + F_0 \left(\frac{2\sigma^2(\mathbf{k})}{T} \right)^{1/2} \mathbf{P}_\zeta(\mathbf{k}) \cdot d\mathbf{W}_t \right] \quad (86)$$

$$(P_{ij})(\mathbf{k}) = \zeta P_{ij}^\perp(\mathbf{k}) + (1 - \zeta)P_{ij}^\parallel = \zeta \delta_{ij} + (1 - 2\zeta) \frac{k_i k_j}{k^2} \quad (87)$$

$$(88)$$

This acceleration field in fourier space allows us to precisely specify at which spatial scales we want to apply the forcing as well as the ratio of compressive and solenoidal modes.

The projection parameter $\zeta \in [0, 1]$ sets the relative contribution of compressible and solenoidal injection rates. If not stated otherwise ζ is set to 0.5. Turning time T and base forcing F_0 depend on the general simulation setup.

The goal is a fully developed turbulence where the turbulent system has reached dynamical equilibrium between energy inflow, supplied by forcing and outflow caused by small-scale dissipation and shock capturing.

The turbulent stirring module *StirGirichids* by P. Girichids was ported to FLEXI and tested. The module can be configured by a multitude of parameters at runtime. The most important one are

rmsv Desired average *root-mean-square-velocity* of the turbulence. When the specified threshold is reached small but perpetual injections keep the turbulence in proximity of the *rmsv*.

kmin, kmax Range of modes where to apply forcing. Usually, the range is set from 1 to 3. Stirring on only the first mode can be imagined as a force field with distinct features half the size of the box. Higher modes divide the box further down accordingly. Limiting forcing to only first three modes avoids imprinting a factitious powerspectrum on the system.

zeta Parameter between 0 and 1 which sets the ratio of *compressive* and *solenoidal* forcing. Many studies have shown a universality of both stirring types ... In this work *zeta* is set to 0.5.

A depiction of the velocity field after energy injection for the first time on an initially constant state is shown in fig. ?? as well as its associated velocity powerspectrum in fig. ???. Obviously, the majority of kinetic energy is crowded on the first three modes: $k = [1, 3]$. Gradually, the energy moves up to higher ks and creates the desired small scale structures typical for turbulence simulations. A fully developed turbulence snapshot is shown in fig. 7.

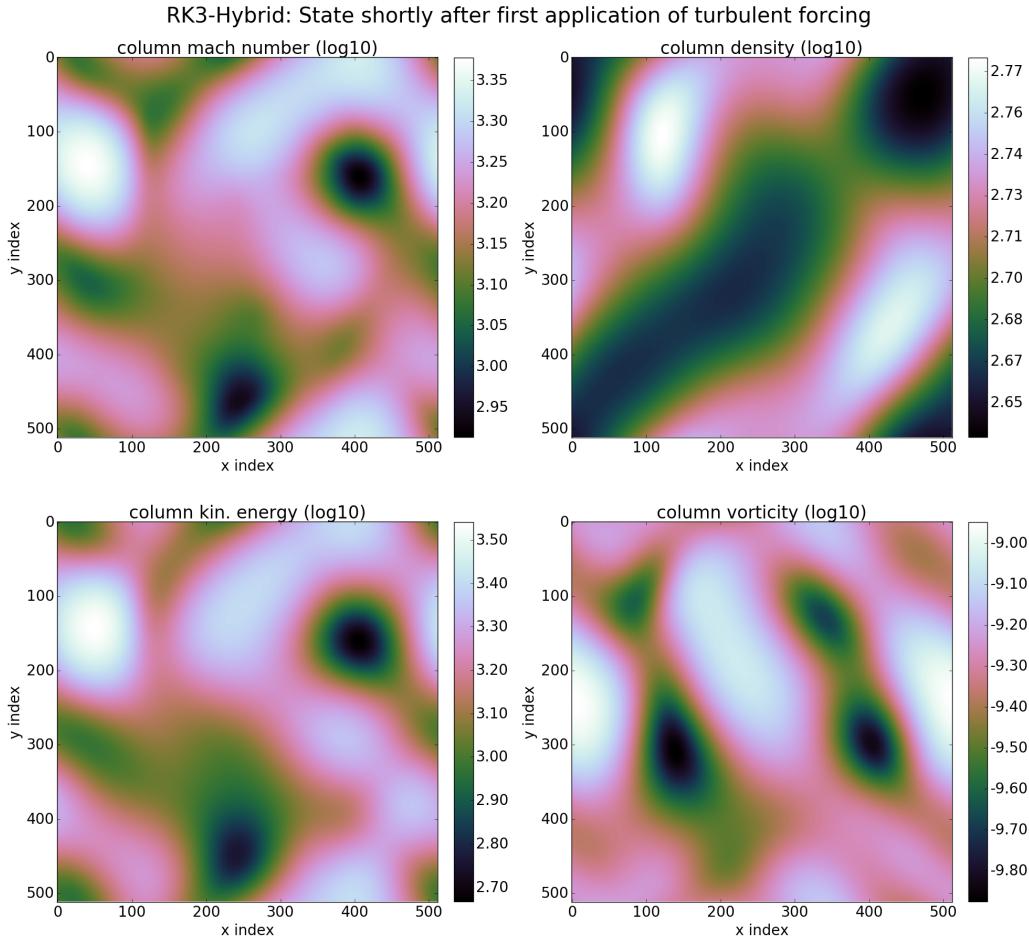


Figure 5: Early-stage snapshot taken immediately after first-time turbulent forcing on the first three modes.

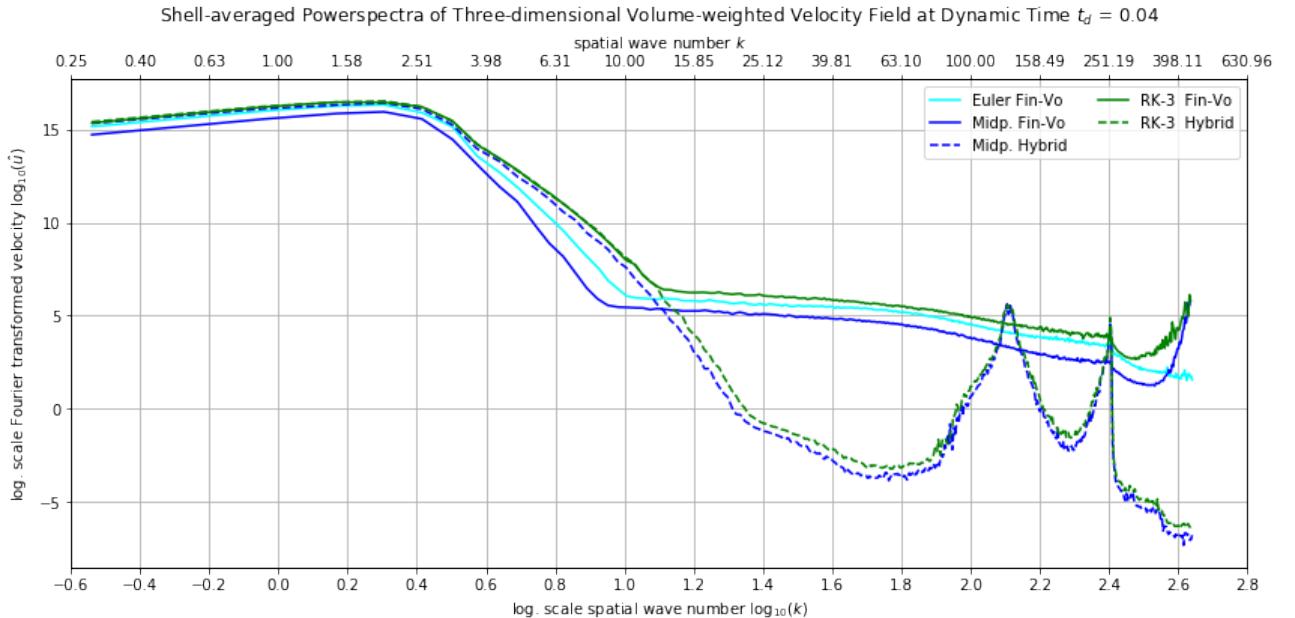


Figure 6: Shell-averaged three-dimensional velocity powerspectrum obtained from the velocity (or Mach in this case) field in fig. 5.

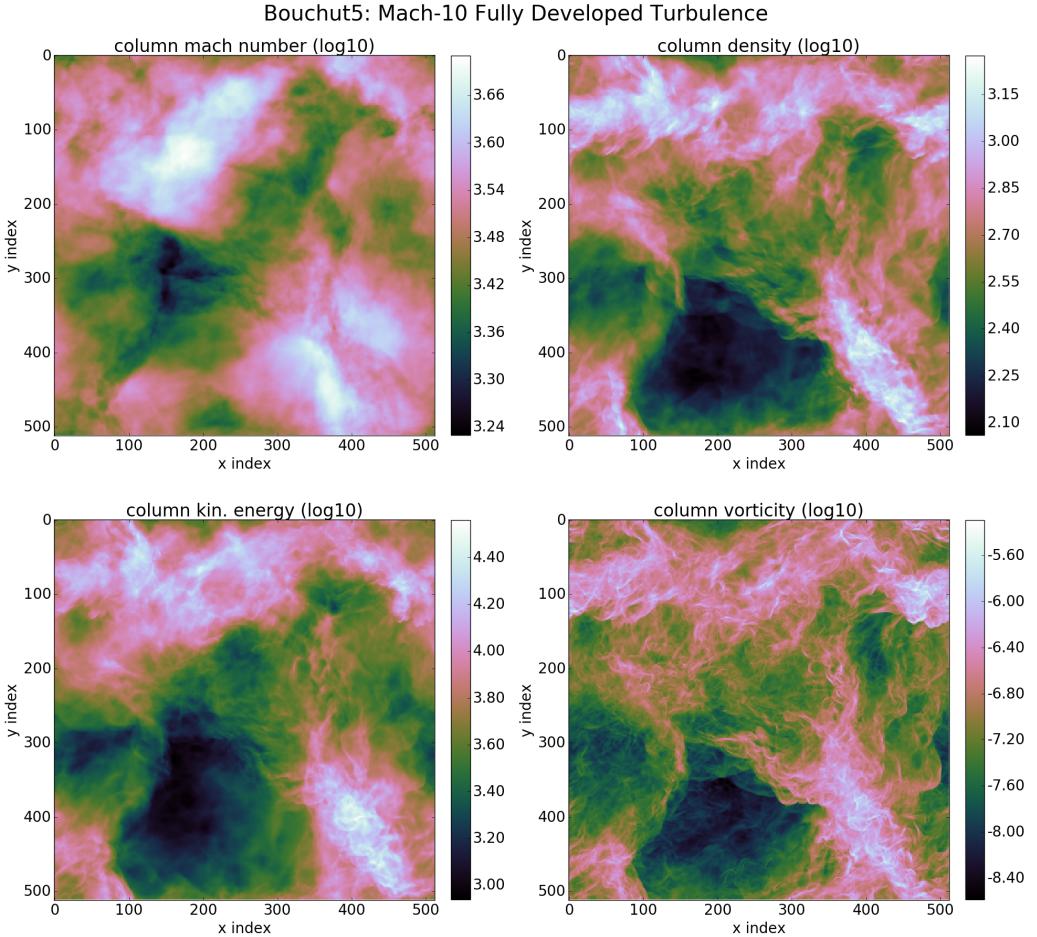


Figure 7: Fully developed Mach-10 turbulence snapshot simulated with Bouchut-5 and Girichidi's Stirred Turbulence Module. Forcing was applied on the first three modes and the picture was taken after four crossing times.

3.3 Shock Capturing

In hypersonic simulations the solver has to deal with strong shocks in an accurate and robust manner. We utilize shock capturing strategies in order to alleviate the destabilizing impact of discontinuities on DG schemes.

This endeavor, however, is far from trivial because of two main reasons. The first is that the exact solution of (nonlinear) purely convective problems develops discontinuities in finite time; the second is that these solutions might display a very rich and complicated structure near such discontinuities. Thus, when constructing numerical methods for these problems, it must be guaranteed that the discontinuities of the approximate solution are the physically relevant ones. Also, it must be ensured that the appearance of a discontinuity in the approximate solution does not induce spurious oscillations which spoil the quality of the approximation; on the other hand, while ensuring this, the method must remain sufficiently accurate near shocks in order to capture the possibly rich structure of the exact solution.

Within the scope of this work three intertwining concepts are developed. First we must detect (*sensoring*) a discontinuity and dampen (*artificial viscosity*) the appearing oscillations. In case this is not sufficient the element switches to FV mode (*switching*) and endures the troubling phase till it can safely revert to DG mode again.

Sensoring Based on the PERSSON indicator we develop a *smoothness* operator which measures the variance of the highest frequencies in modal space of the Galerkin polynome. In other words we build an oscillation detector.

First we express the solution of order N_p within each eleement in terms of an orthogonal basis as

$$u = \sum_{i=1}^{N_p} u_i \psi_i, \quad (89)$$

where N_p is the total number of terms in the expansion and ψ_i are the LEGENDRE basis functions. See fig. 8 for an illustrative depiction thereof.

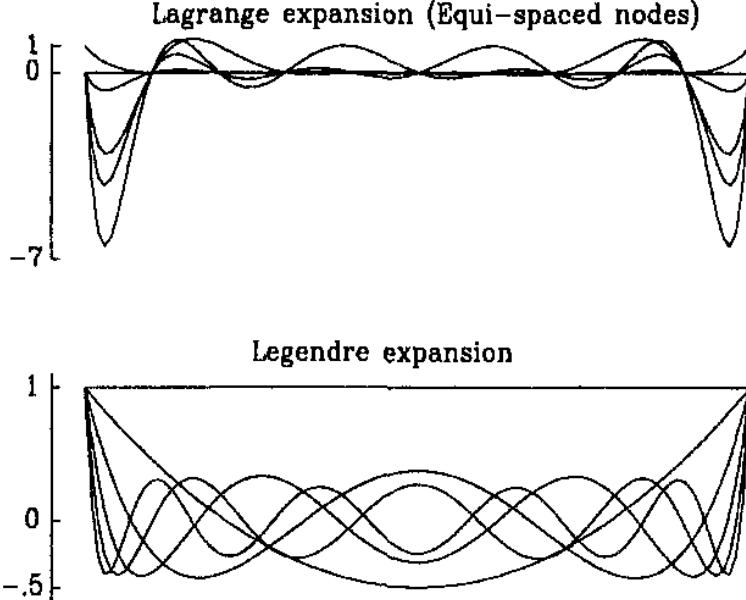


Figure 8: Expansion modes for the Lagrange and the Legendre Basis.

Now we only consider the terms up to order $N_p - 1$, that is

$$u_{N_p-1} = \sum_{i=1}^{N_p-1} u_i \psi_i, \quad (90)$$

Whithin each element Ω we define the following *smoothness* sensor

$$s = \log_{10} \frac{\langle u - u_{N_p-1}, u - u_{N_p-1} \rangle}{\langle u, u \rangle}, \quad (91)$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product in $L_2(\Omega)$.

The smaller the indicator s , the smoother is the approximating solution.

Artificial Viscosity Artificial viscosity $0 < \epsilon \lll 1$ is introduced to the Euler equations as a very weakly interacting diffusion term.

$$\frac{\partial \underline{U}}{\partial t} + \nabla \cdot \underline{\underline{F}}(\underline{U}) = \nabla \cdot (\epsilon \nabla \underline{\underline{\tau}}), \quad (92)$$

with $\underline{\underline{\tau}}$ being the deviatoric stress tensor. The amount of viscosity varies for each element and depends on the current shock strength.

We have to consider two cases.

If the element is in FV mode ϵ is set quadratic proportional to the maximal root-mean-square velocity u_{rms} of the element.

$$\epsilon_{FV} = \epsilon_{FV,0} \cdot \max(u_{rms})^2 \quad (93)$$

with $\epsilon_{FV,0}$ being an arbitrary but sufficiently small constant. This measure should only have an effect on extreme velocity spikes otherwise the governing equations would get viscous and violate the timestep constraints.

In case of an element in DG mode the amount of ϵ is based on the *Persson Indicator* introduced above.

$$\epsilon_{DG} = \begin{cases} 0 & \text{if } s < s_0 - \kappa \\ \epsilon_{DG,0} & \text{if } s > s_0 + \kappa \\ \frac{\epsilon_0}{2} \left(1 + \sin \frac{\pi(s-s_0)}{2\kappa}\right) & \text{else} \end{cases} \quad (94)$$

The parameters $\epsilon_{DG,0}$ and κ are chosen empirically though $\epsilon_{DG,0}$ should be sufficiently small as well.

FV-DG Mode Switching By setting a specific threshold for the *smoothness indicator* s one can decide when to switch back and forth between DG and FV mode. This procedure is done at every timestep hence the elements in FV mode should follow along the shock waves throughout the domain. We perform a comparing study of two manifestations of the Persson indicator and clarify the advantages over the other.

Based on eqn. (??) the first variation, called *indicator A*, reads

$$s_A = \log_{10} \max \left(\frac{\langle u^2 - u_{N_p-1}^2 \rangle}{\langle u^2 \rangle}, \frac{\langle u_{N_p-1}^2 - u_{N_p-2}^2 \rangle}{\langle u^2 - u_{N_p-1}^2 \rangle} \right) \quad (95)$$

The second, called *indicator B*, reads

$$s_B = \log_{10} \max_q \frac{\langle u - u_{N_p-q} \rangle^2}{\langle u_{N_p-q} \rangle^2} \quad (96)$$

Fig. 9 shows a fully developed turbulence with shocks and their associated heat maps of indicator values calculated from the pressure. A normalized distribution (PDF) of indicator values s is plotted in fig. 10.

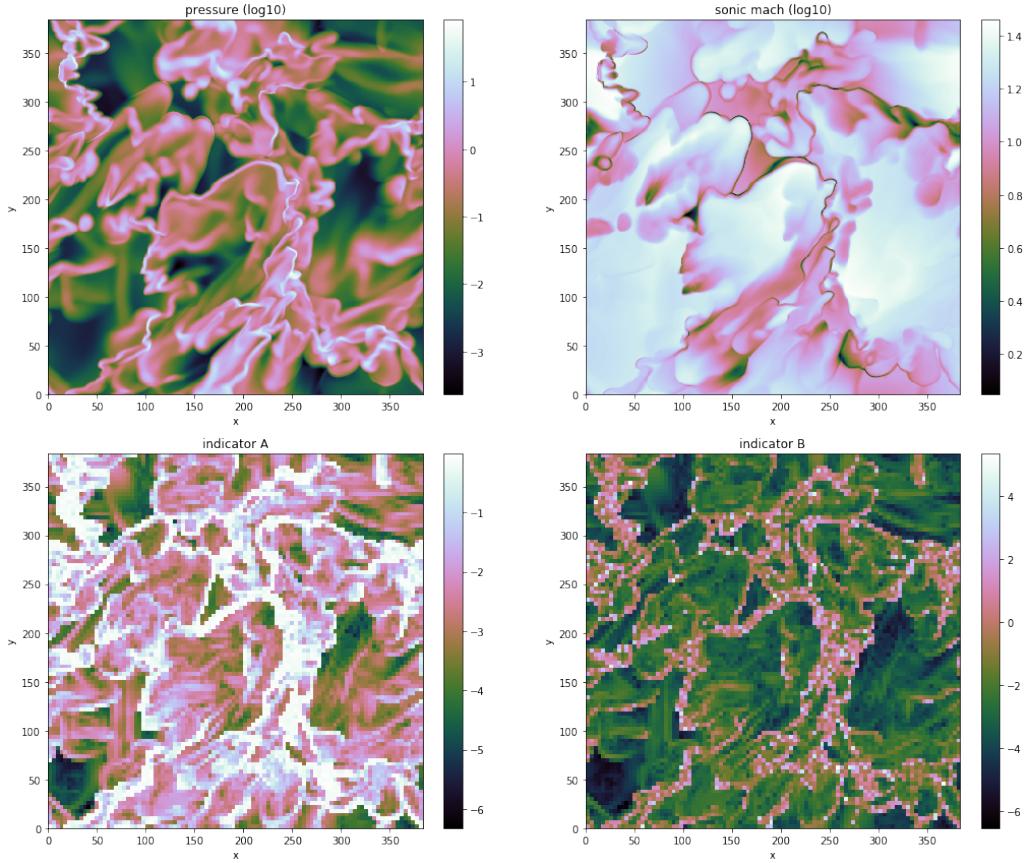


Figure 9: Slice with thickness of one element.

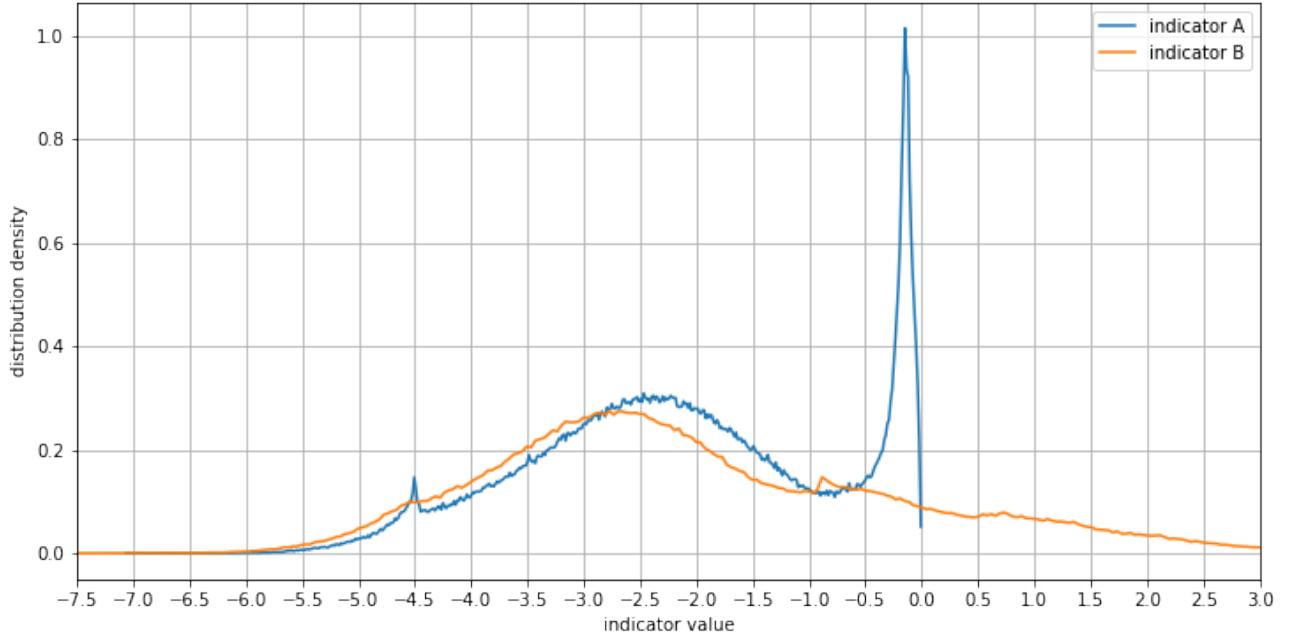


Figure 10: Probability Distribution of calculated indicator values.

The most prominent difference is the higher sensitivity of indicator A and its limitation to only negative values. The latter is a result of the normalizing fractions in eqn. (95). Both variants trail the shock fronts with sufficient accuracy. Considering the PDF in fig. 10, one can observe spikes in both curves in the interval between -1 and 0. They herald the realm of

strong shocks. The broad hill around -2.5 contains the majority of stressed elements which were affected by a recently pervading shock front. The little spike at -4.5 stems from a bias introduced by the indicator A. Numerous tests revealed an empirical threshold of $s_T = -4.5$ where the switching takes place. Clearly, indicator A unambiguously signalizes elements with unresolvable discontinuities. This opens the possibility to further strengthen the DG scheme for a broader coverage of the domain beyond $s_T = -4.5$.

3.4 Grid Spaces & Data Transformation

Grid Spaces The term *grid space* refers to the arrangement of nodes within an element. Four types of node configurations are of importance here: *Face-centered grid* (FCG), *body-centered grid* (BCG), *Gauss nodal grid* (GNG) and *Gauss-Lobatto nodal grid* (LNG). Finite Volumes live on a BCG while the Lagrange polynomials are anchored on either GNG or LNG (cf. section 2.4.1/Lagrange Polyonome). In order to compare and transfer data between the numerical schemes one has to translate between the different grids via Lagrange interpolation.

Although FLEXI offers curvilinear grids we only need the Euclidean geometry. Thus whe have a one-to-one relationship between the four grid spaces and are able to directly lay them on top of each other. Fig. 11 depicts a one-dimensional domain consisting of either eight cells (dashed lines) or two elements (thick lines) with four nodes. Four nodes imply a Lagrange polynomial of third order.

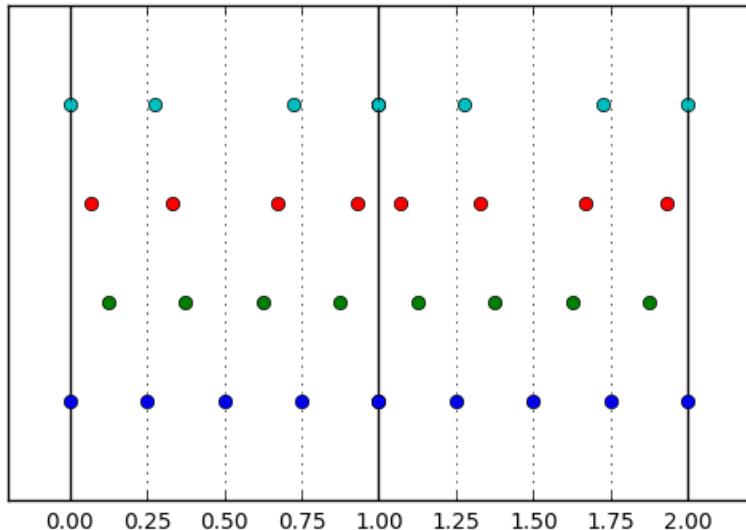


Figure 11: Nodes in a two element grid each consisting of four cells. From bottom to top: face-centered, body-centered, Gauss nodes ($n = 3$), Gauss-Lobatto nodes ($n = 3$).

When interpolating data between finite volumes and DG elements, cells get grouped together in numbers equal to the number of nodes in a superincumbent element. Interpolation happens in each group/element independently.

TODO: Refer to source code.

Interpolation Error The Lagrange interpolating polyonome $L(x)$ of order N is only exact for a polynomial function $f(x)$ of order $M \leq N$.

The remainder $R(x) = L(x) - f(x)$ can be expressed as

$$R(x) = L(x) f^{(N+1)}|_{\zeta}, \quad x_0 \leq \zeta \leq x_N, \quad (97)$$

using Taylor series analysis (cf. [?], p. 878). x_0, \dots, x_N being the interpolation nodes and $f^{(N+1)}(x)$ is the $(N+1)$ -th derivative of $f(x)$. Since $M \leq N$, $f^{(N+1)}(x) = 0 \Rightarrow R(x) = 0, \forall x$. \square

An example for an exact Lagrange interpolation in two dimensions is shown Fig. 12.

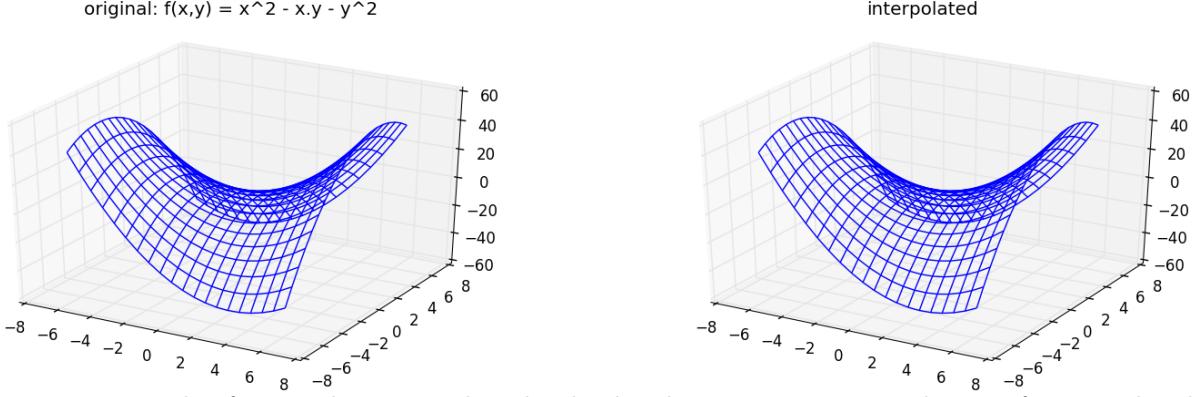


Figure 12: Example of a two-dimensional exact third-order Lagrange interpolation of a second-order polynomial. The relative error $\Delta_{\text{RMS}} = 0.000000\%$ (cf. eqn. (??)) is below machine precision.

Now instead of the remainder $R(x)$ we define a *root-means-square variance* as a kind of distance measure between an arbitrary function $f(x)$ and its interpolant $\tilde{f}(x)$.

$$\Delta_{\text{RMS}} = (f - \tilde{f})_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_i^N (f_i - \tilde{f}_i)^2}. \quad (98)$$

The relative error is calculated by

$$\Delta_{\text{RMS,rel.}} = \frac{\Delta_{\text{RMS}}}{f_{\text{RMS}}}, \quad (99)$$

where $f_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_i^N f_i^2}$.

Remark One has to ensure that f_i and \tilde{f}_i live in the same grid space.

Transferring Data from FLASH to FLEXI TODO: For the implementation details of data transferration we refer to the source code in Here only a brief sketch of the workflow is provided.

In order to give a sense of intuition we imagine a Cartesian grid with 512 body-centered points in each of the three dimensions. Each data point represents a finite-volume read in from a FLASH checkpoint file. We subdivide the contiguous block of 512^3 into elements of $4^3 = 64$ nodes. Within each element the data gets transformed via three-dimensional third-order Lagrange interpolation into Gauss or Gauss-Lobatta nodal space. Finally, the new data is written to a FLEXI checkpoint file. The reverse procedure is available, too. Which is useful for analysis and visualization.

Remark Before the interpolation step the data has to be translated from primitive to conservative variables (cf. section ??/FLEXI and section ??/Primitive and Conservative Variables).

Fig. 13 and fig. 14 show practical results of a data transfer from FLASH to FLEXI. They substantiate the applicability of the process within acceptable error margins. Unavoidable is the truncation of sharp spikes in the data. Evidently, critical information in the proximity of strong shocks gets lost. Considering these facts it is sensible to apply a Gaussian blur before the actual transfer. Consequently, almost identical initial states for FLASH and FLEXI can be ensured. Decaying turbulence simulations benefit from this as done in section 4.3.

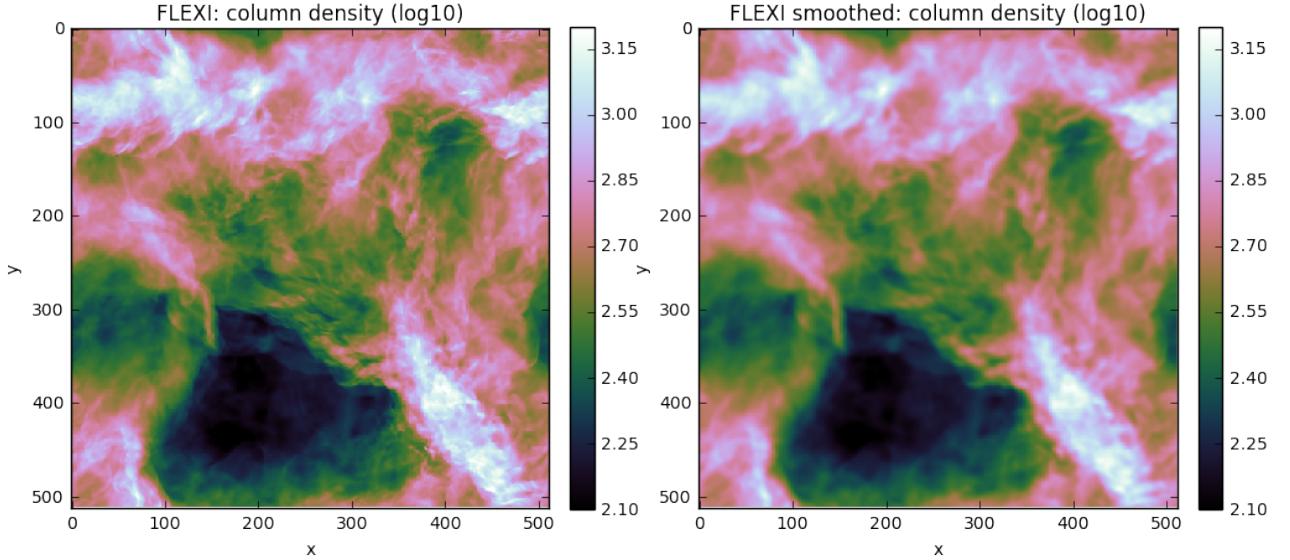


Figure 13: Column densities plots of interpolated hypersonic (Mach 10) turbulence (see fig. 7) from FLASH to FLEXI. Left side: The density is interpolated via a third-order (four nodes per element) Lagrange interpolation as described in the text. The relative interpolation error is estimated as $\Delta_{\text{RMS}} = 0.118763 \approx 12\%$ (cf. eqn. (??)). Right side: The interpolation can be improved by introducing a Gaussian blur before the interpolation step: $\Delta_{\text{RMS}} = 0.2808160 \approx 3\%$.

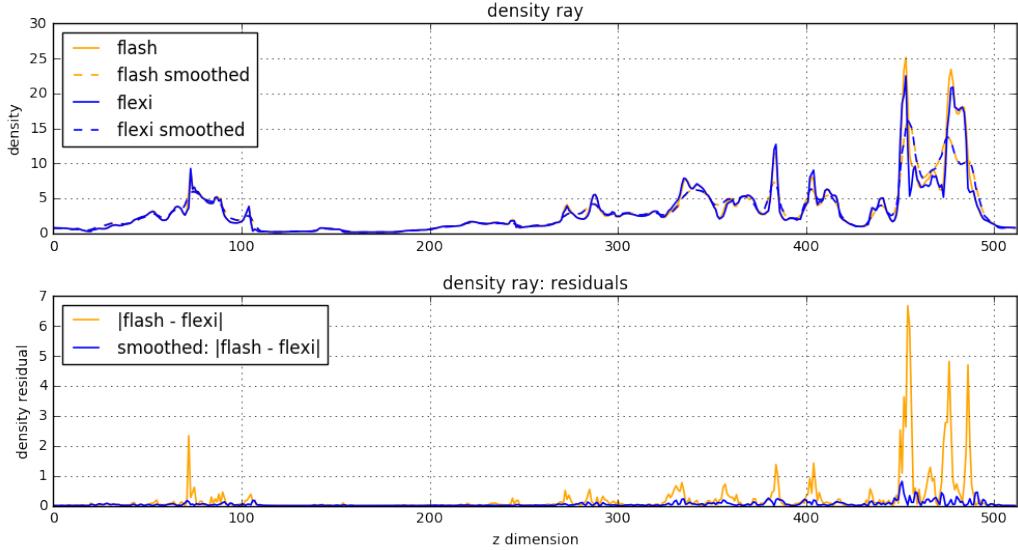


Figure 14: Comparison of one-dimensional density profiles and residuals through a hypersonic (Mach 10) turbulent box with original simulation data (flash) and transferred interpolated data (flexi). The density ray is extracted from same data already shown in fig. 7 and fig. 13. The smoothing beforehand the interpolation was done via a Gaussian blur which minimizes the residual considerably.

4 Results & Discussion

4.1 Sod Shock Tube Problem

Before we can reliably discuss the results of the turbulence simulations we need to validate the correctness of the solvers especially with regards to the modifications presented in the previous chapter. The standard test for the correct treatment of shocks is the SOD Shock Tube Problem in one dimension. At first, we look at the classical adiabatic test case, then introduce polytropic cooling (isothermal) and finally apply a strong shock situation by setting the initial left-side velocity to $u_L = 15$. All runs took place with a conservative CFL number of $CFL = 0.4$ where all runs do not crash. If not indicated otherwise the initial conditions were set like follows

Table 1: Sod-Shock: Initial Condition

Left Side			Right Side		
Name	Symbol	Value	Symbol	Value	
density	ρ_L	1.0	ρ_R	0.125	
velocity	u_L	0.0	u_R	0.1	
pressure	p_L	1.0	p_R	0.0	

We suggest a resolution of 512 elements for the whole domain $\Omega = [0, 1]$ since it resembles the same resolution applied on the turbulence simulations. This way we get a comparable insight on how well shocks can be resolved within the periodic box of the same length. Table table 2 outlines conducted experiments in tabular form.

Table 2: Overview of Conducted Runs for the Sod Shock Tube Problem

Setups	Adiabatic, Isothermal, Adiabatic w. Strong Shock, Isothermal w. Strong Shock
Solvers	Bouchut 3, Bouchut 5, PPM, Euler FV, Euler Hybrid ¹ , Midp. FV, Midp. Hybrid, RK3 FV, RK3 Hybrid
Resolution	512
CFL	0.4

4.1.1 Adiabatic & Isothermal Shock

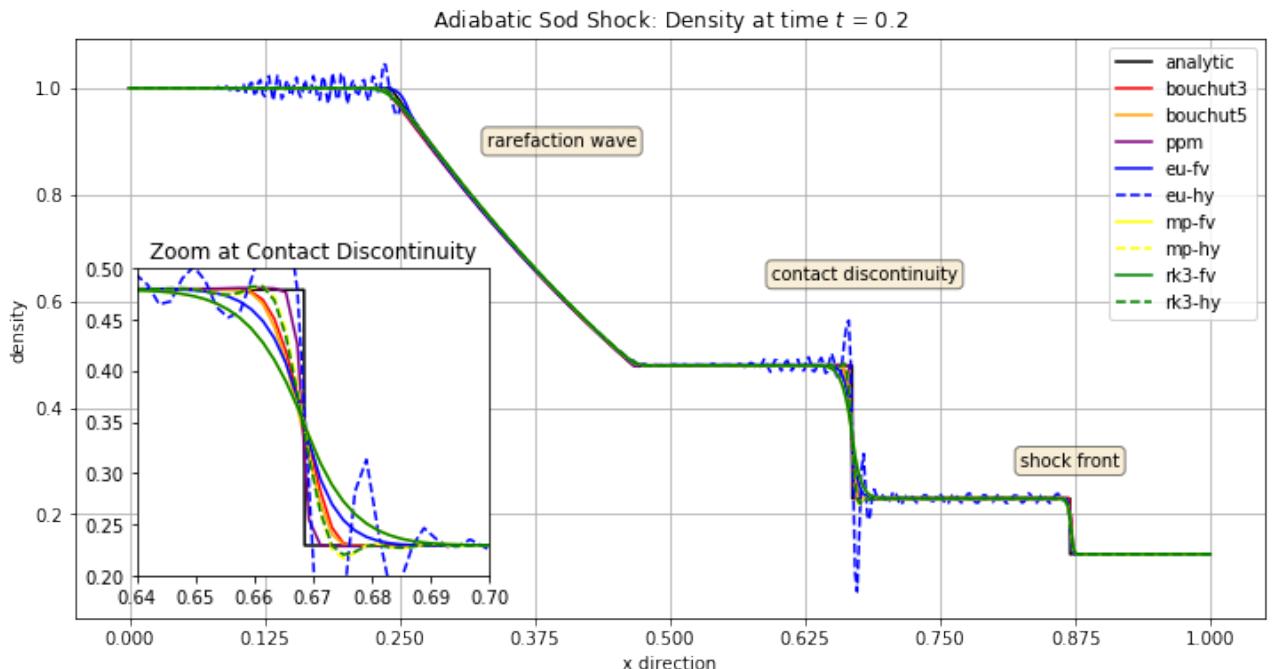


Figure 15: Canonical density profile at conventional time $t = 0.2$ with anticipated regions and discontinuities outlined in the plot.

In fig. 15 all utilized solvers satisfy the expected density profile of the classical Sod-Shock problem with varying precision. PPM yields the most accurate result while on the other hand RK3-Hybrid smears out the discontinuities considerably. Looking closely at the zoomed area all hybrid schemes (blue, yellow and green dashed lines) tend to oscillate. The unacceptable ringing of the Euler-Hybrid is a consequence of the instability of the Euler Time Integration within the DG operator. TODO: Reference. Consequently, Euler-Hybrid is disqualified and will be discarded in future discussion. Same assertions can be made for the isothermal case shown in fig. 16.

¹Euler Hybrid was only run with adiabatic and isothermal Sod the since it is considered inherently unstable.

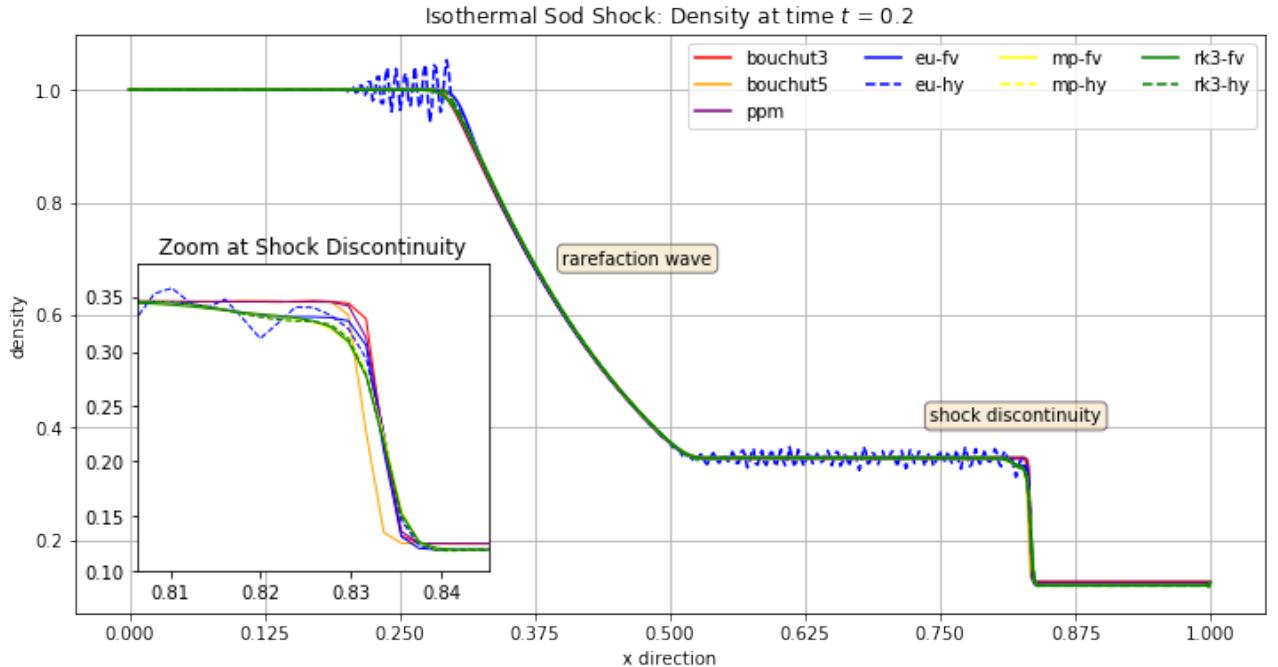


Figure 16: Isothermal density profile at conventional time $t = 0.2$ with expected regions and discontinuities outlined in the plot. The contact discontinuity gets suppressed due to polytropic cooling.

4.1.2 Adiabatic & Isothermal Sod with Strong Shock

Now we want to compare how the solvers cope with a strong shock situation ($u_L = 15$) first in the adiabatic and then in isothermal setting with polytropic cooling.

Considering fig. 17 all solutions yield a quite similar profile. The shock waves build up rapidly up to a certain height travel at a constant speed from left to right and widen with increasing time. One can recognize two distinct parts. A fast traveling pedestal moving at the speed of the initial velocity $u_{fast} = u_L = 15$ and a slow moving shot up rear at the speed of around two third of the fast wave $u_{slow} \approx 2/3 u_{fast} = 10$.

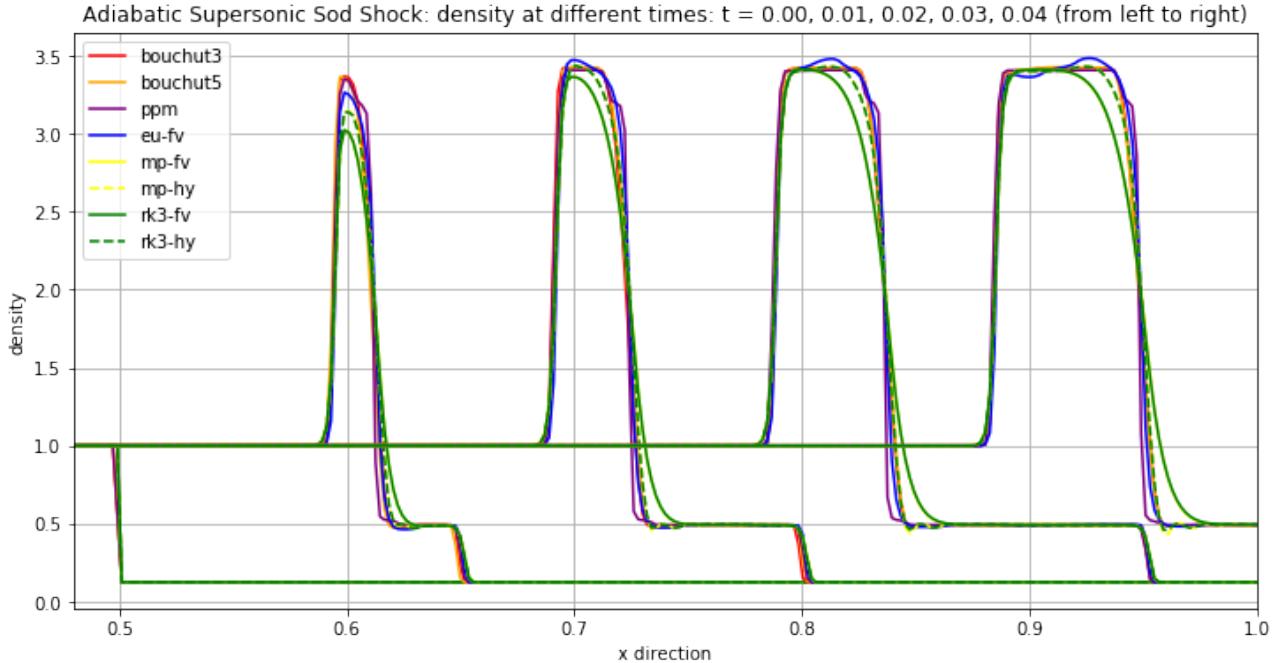


Figure 17: Adiabatic density snapshots depicting the movement of a strong shock wave from left to right. Note: Only the right half of the domain is shown. The pedestal of the rightmost wave ($t_d = 0.04$) has gone out of visible range.

The solutions for the strong shock setup with cooling, see fig. ??, are quite different. Most notably, the density peaks are of an order higher and the widths are considerably narrower than before. There are no recognizable partitions into a slow and fast moving wave. Additionally, the solvers do not agree on the shock wave speed with the Euler Fin-Vo being the slowest but resembling a very sharp high-rising peak on the other hand. We have no concise explanation for this but it should be safe to say that the mean free path of a shock wave in a turbulence simulation does not exceed half the size of the box. Hence, the influence of the velocity divergence is minimal.

It probably makes no sense commenting about the details of the curve shapes, see fig. 19, since at least around 16 cells (four elements of four nodes) are trying to resolve the shock wave. This is an unsufficient resolution compared to the adiabatic case where a shock wave is four to five times broader. Consequently, isothermal turbulence simulations are severely under-resolved.

Isothermal Supersonic Sod Shock: density at different times: $t = 0.00, 0.01, 0.02, 0.03, 0.04$ (from left to right)

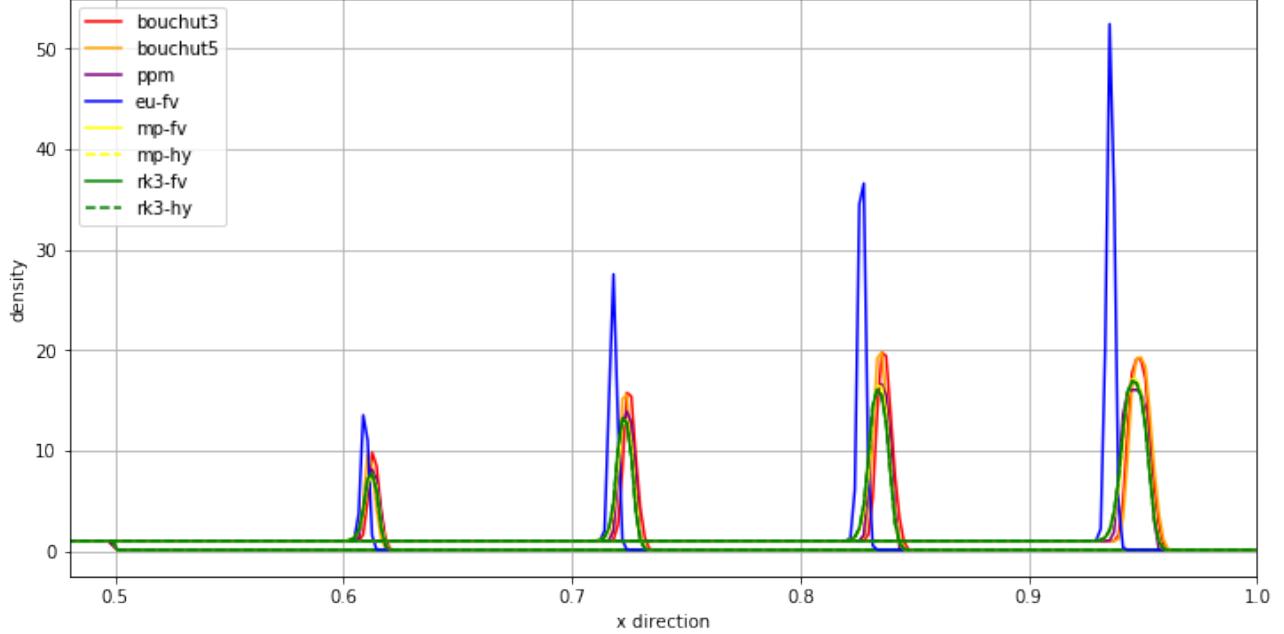


Figure 18: Isothermal density snapshots depicting the movement of a strong shock wave from left to right. Note: Only the right half of the domain is shown.

Isothermal Supersonic Sod Shock: Zoom at jump from time $t = 0.04$

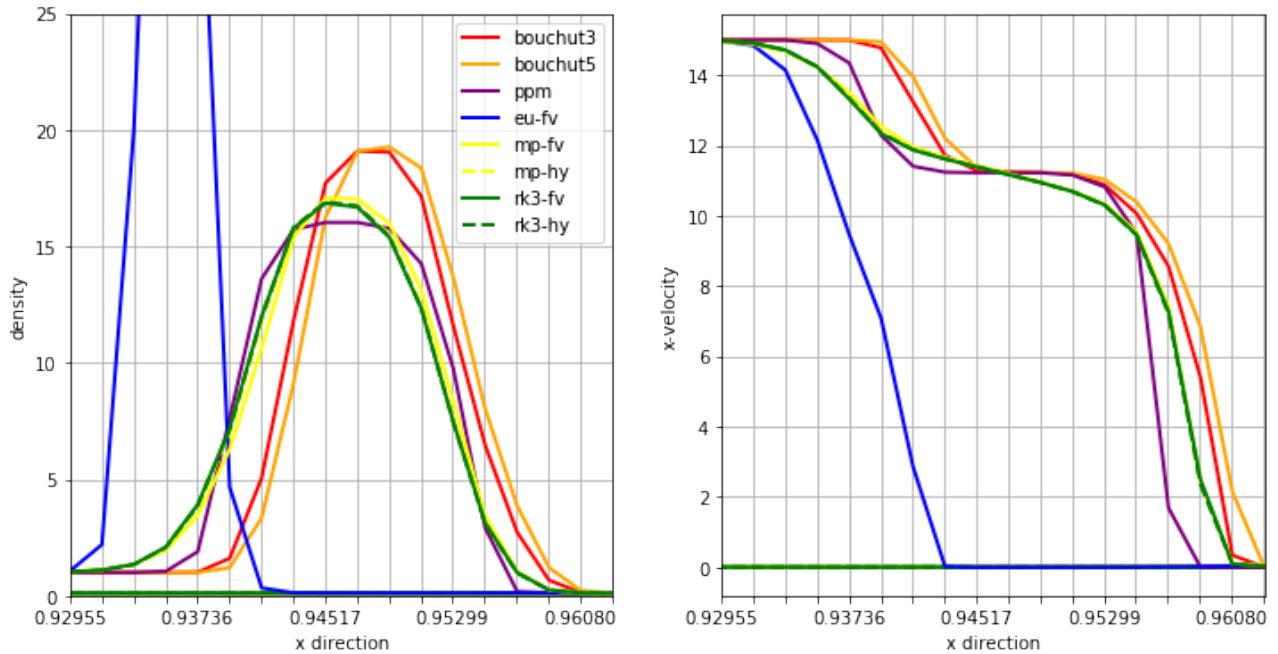


Figure 19: Zoom at the rightmost shock wave in fig. ???. The grid spacing of the x axis resemble the width of a finite volume.

4.2 Driven Turbulence

The isothermal driven turbulence simulation consists of two phases. At the beginning of the run turbulent forcing, explained in section 3.2, stirs the medium up to a certain root-mean-square velocity which is equivalent to the sonic Mach number $\mathcal{M}_{\text{forcing}}$ (cf. section ??/Choice

of Parameters). For now FLEXI solver can only bear driven Mach numbers up to Mach 3. Consequently we settled to $\mathcal{M}_{\text{forcing}} = 2.5$, retaining a small safety gap. One crossing time becomes $T_d = L/\mathcal{M} = 1/2.5 = 0.4$. When reaching $\mathcal{M} = 2.5$ as fast as possible but slow enough not to crash the solver the stirring module perpetually injects just enough energy to stay in proximity of $\mathcal{M}_{\text{forcing}}$. The turbulence is given four crossing times to fully develop after which forcing is deactivated. We will call this the *turbulent phase*. During the *decay phase* it takes another two turning times to let the turbulence drop below the supersonic barrier.

The turbulence simulations live within a three-dimensional periodic box of equal length in all dimensions. It means the state is a closed system where mass, momentum and energy are conserved. Via active forcing kinetic energy enters the medium on large scales and leaves it as internal energy via active cooling on small scales. This way a stationary flow of momentum gets pumped through the system. In conjunction with *bulk motion correction* hyper-sonic turbulence emerges which presents a stress test for every numerical scheme.

Based on considerations in section 2.1 we initialize the state as listed in table 3.

Table 3: Overview of initial values set for all turbulence simulations. While FLASH is using primitive variables FLEXI uses conservative variables.

FLASH			FLEXI		
Name	Symbol	Value	Name	Symbol	Value
density	ρ_0	1.0	density	ρ_0	1.0
x-velocity	$u_{x,0}$	0.0	x-momentum	$\mathcal{P}_{x,0}$	0.0
y-velocity	$u_{y,0}$	0.0	y-momentum	$\mathcal{P}_{y,0}$	0.0
z-velocity	$u_{z,0}$	0.0	z-momentum	$\mathcal{P}_{z,0}$	0.0
pressure	$p_{x,0}$	0.6	total energy	$E_{x,0}$	0.9

Remark Some numerical schemes actually solve the magneto-hydrodynamics equations. By setting the initial magnetic density flux field $B_0 = 0$, the MHD equations resemble the compressible Euler equations.

Next to the Mach number, the course of energy and energy dissipation are measured. Additionally, we look at selected density distributions and powerspectra in the turbulent and decay phase. Checkpoints for analysis and plotting are saved ten times per crossing time. Table 4 gives an overview of the conducted experiments.

Table 4: Outline of Driven Turbulence Simulations

Solver	B3	B5	PPM	EU-FV	MP-FV	MP-HY	RK3-FV	RK3-HY
CFL	0.8	0.8	0.8	0.4	0.8	0.6	0.9	1.2
$\mathcal{M}_{\text{forcing}}$	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
$t_{\text{forcing,stop}}$	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
$t_{\text{sim. end}}$	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
$N_{\text{checkpoints}}$	60	60	60	60	60	60	60	60

4.2.1 Mach Number Evolution

Fig. 20 clearly depicts the stir-up, turbulent and decay phases. After a steep stirring phase of roughly 0.1 turning times the forcing routine keeps the solvers from decaying below $\mathcal{M} = 2.5$.

The difference in rising times and sawtooth shapes in the zoomed picture (fig. 20) are mainly a result of unmatched timesteps (cf. fig. ??) and hence amount of injected energy per timestep. More meaningful is the rate of decay from $t_d = 4$ to 6 (decay phase) where the solvers notably diverge. The best results regarding retaining most kinetic energy, see also fig. 28, is the Euler FV followed by both RK3 solvers. PPM and Bouchut 3 and Bouchut 5 are somewhere in the middle. Both Midpoint solvers seem to be most dissipative and end up at roughly $\mathcal{M} = 0.75$.

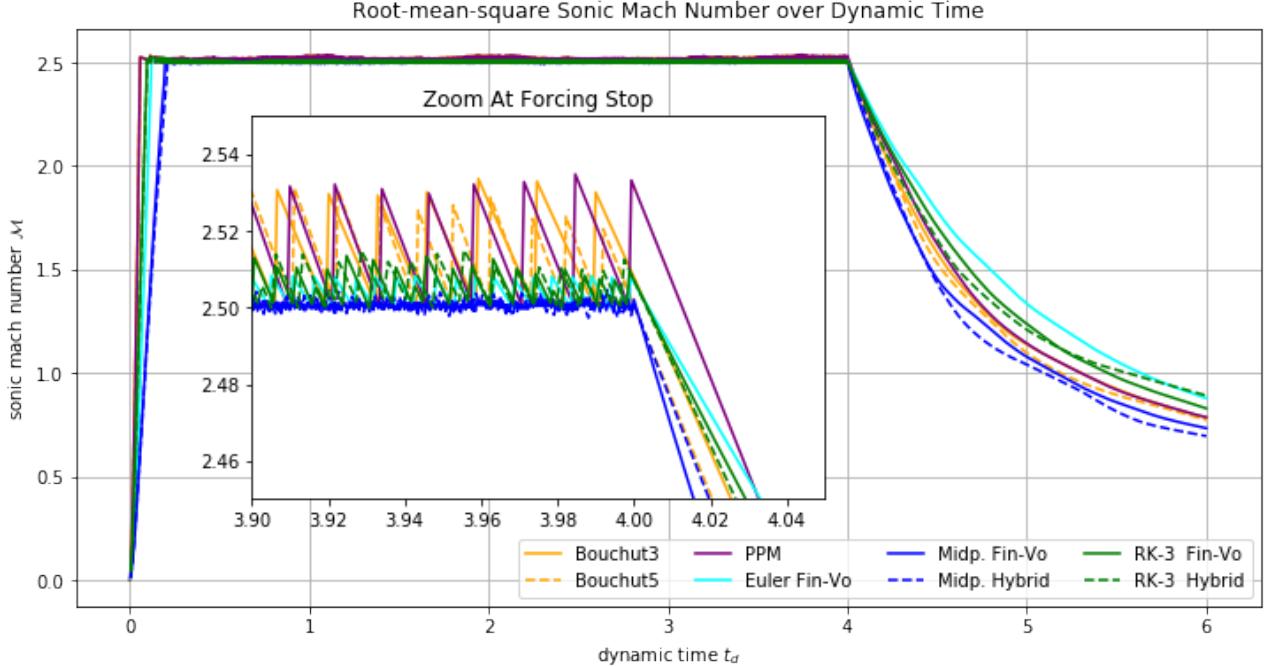


Figure 20: Time evolution of the root-mean-square Mach number \mathcal{M} . After a steep elevation to $\mathcal{M} = 2.5$ the forcing routines keep the solvers from decaying by perpetually injecting small amounts of kinetic energy. Taking a closer look (zoomed picture) a distinct sawtooth profile appears.

4.2.2 Column Density & Velocity

Following three figures show the column Mach number and the column density of all eight conducted runs after four crossing times when the forcing is stopped and the phase of turbulent decay begins. At this point in time all simulations are to be expected to show a fully developed isotropic supersonic turbulence with a variety of large and small scale features as well an established energy cascade.

In fig. 21 the three solvers from FLASH are grouped together in one page. Obviously, they fulfill the expectations and serve as reference for the solutions brought up by FLEXI.

First of all, the solutions from Euler FV and the RK3 schemes resemble an isotropic supersonic turbulence as well while the Midpoints seem to differ considerably. They show a cascade from large scale structures down to smaller scales which we will show later on. However there are some problems resolving structures on smallest scales. The solutions appear smoother than in fig. 21. In contrast to the others Midpoint FV (fig. 22) has a very distinct feature with a large blob of high mass concentration and contiguous regions of void swept empty by a strong velocity field. Apparently, the large scale modes of the driving force field (cf. section 3.2) at that time drags the majority of the medium into one big lump.

The solutions during the decay phase at $t_d = 5$ (fig. 24, fig. 25 and fig. 26) look much more

alike. Most strikingly, Euler FV and both RK3 schemes have the most energy left in the system proofed by the color intensity of the Mach number plots. All density plots reveal an isotropic turbulence with large and small scale features. As mentioned before FLEXI's solvers again show fewer features at smallest scales compared to FLASH.

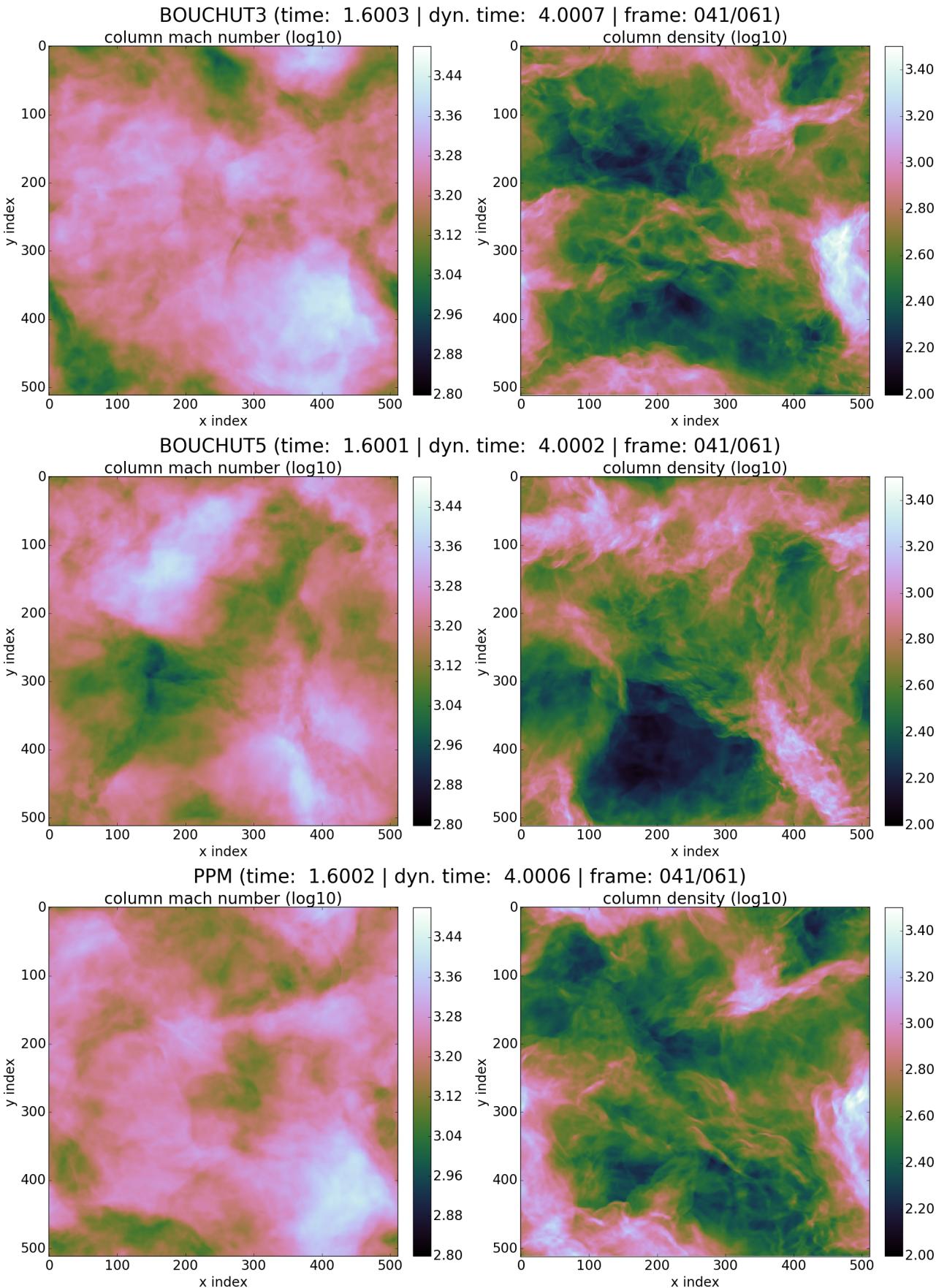


Figure 21: Column Sonic Mach Number and column density along z-axis at the time when forcing is stopped: $t_d = 4.0$.

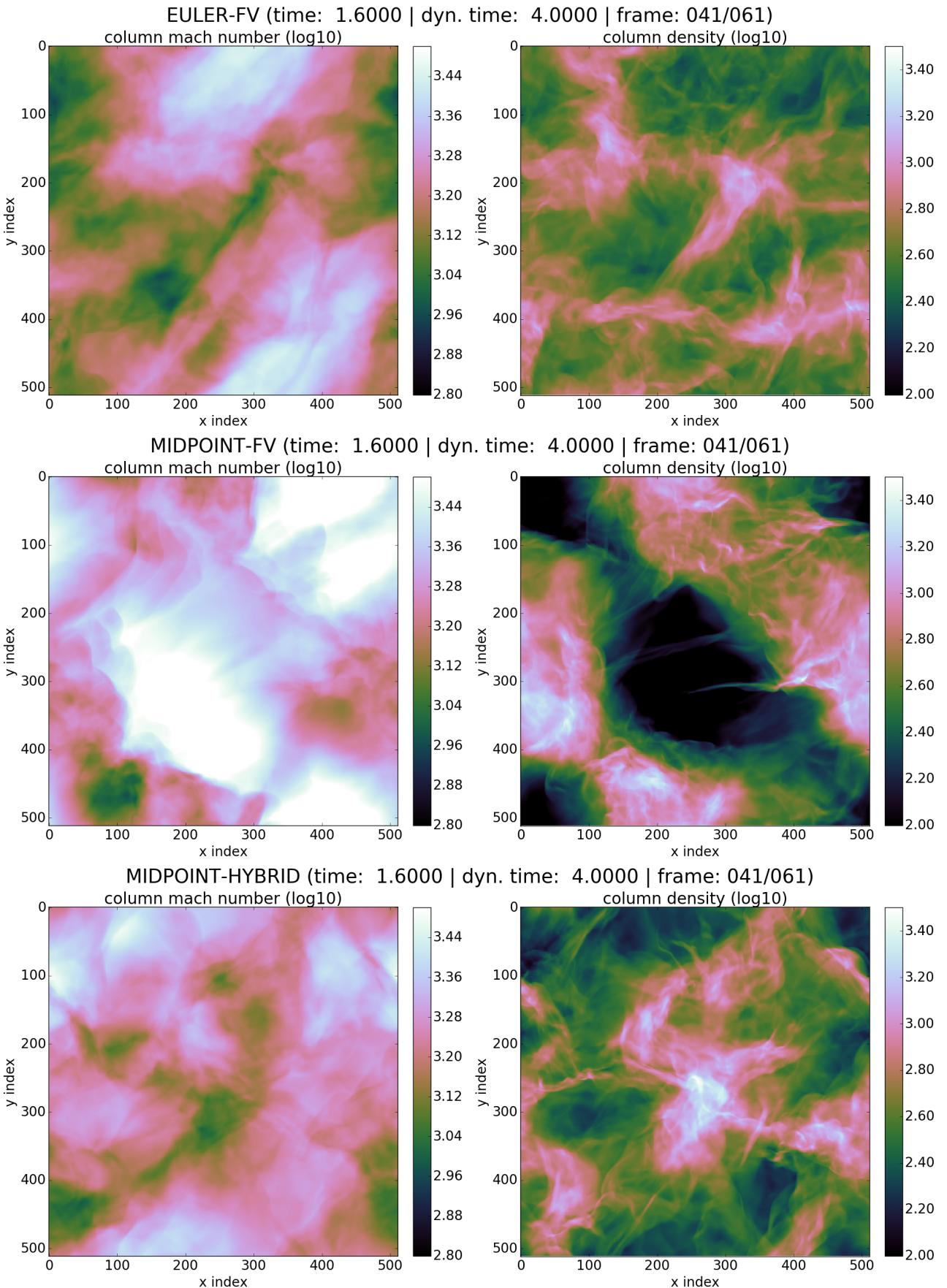


Figure 22: Column Sonic Mach Number and column density along z-axis at the time when forcing is stopped: $t_d = 4.0$.

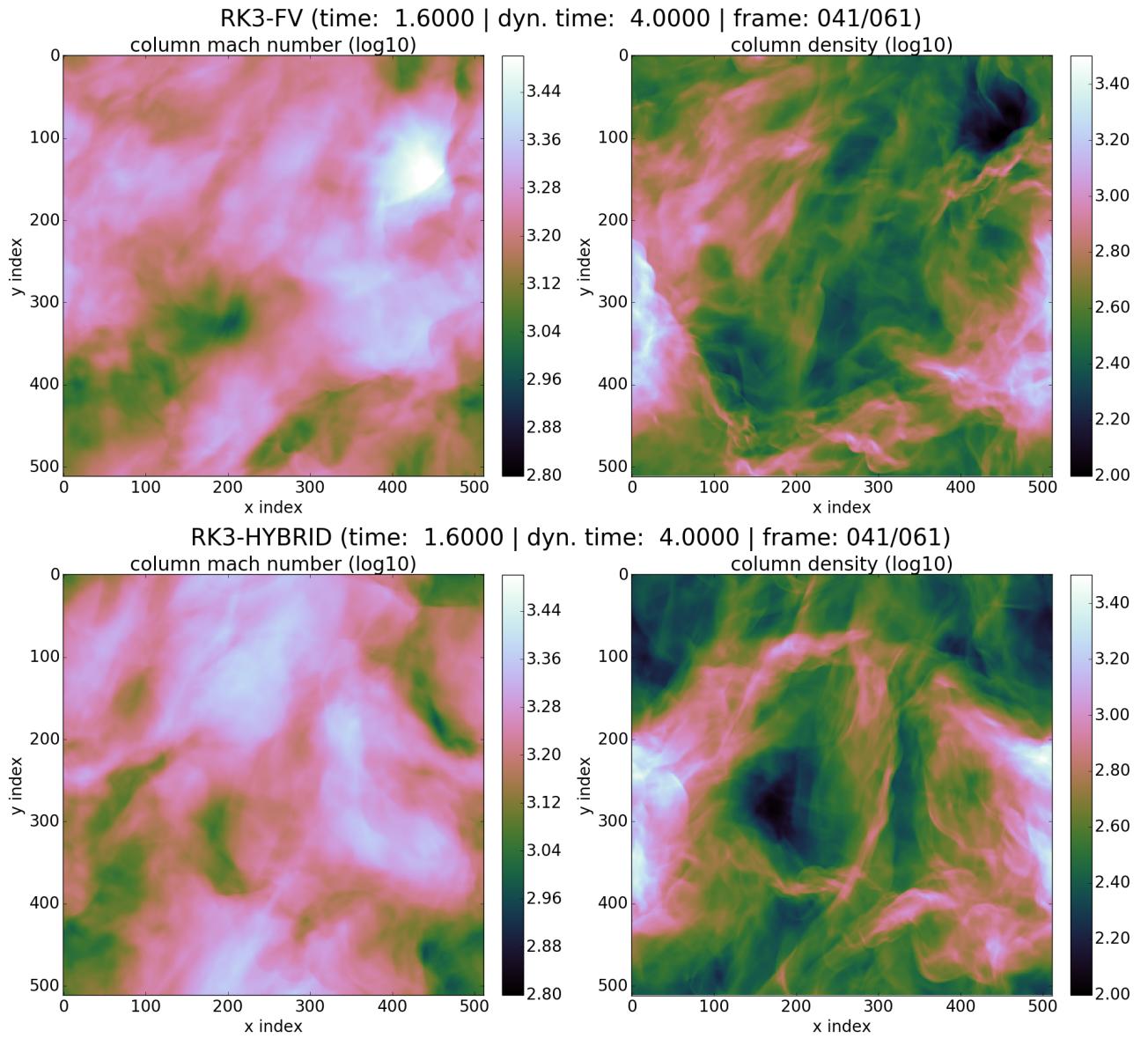


Figure 23: Column Sonic Mach Number and column density along z-axis at the time when forcing is stopped: $t_d = 4.0$.

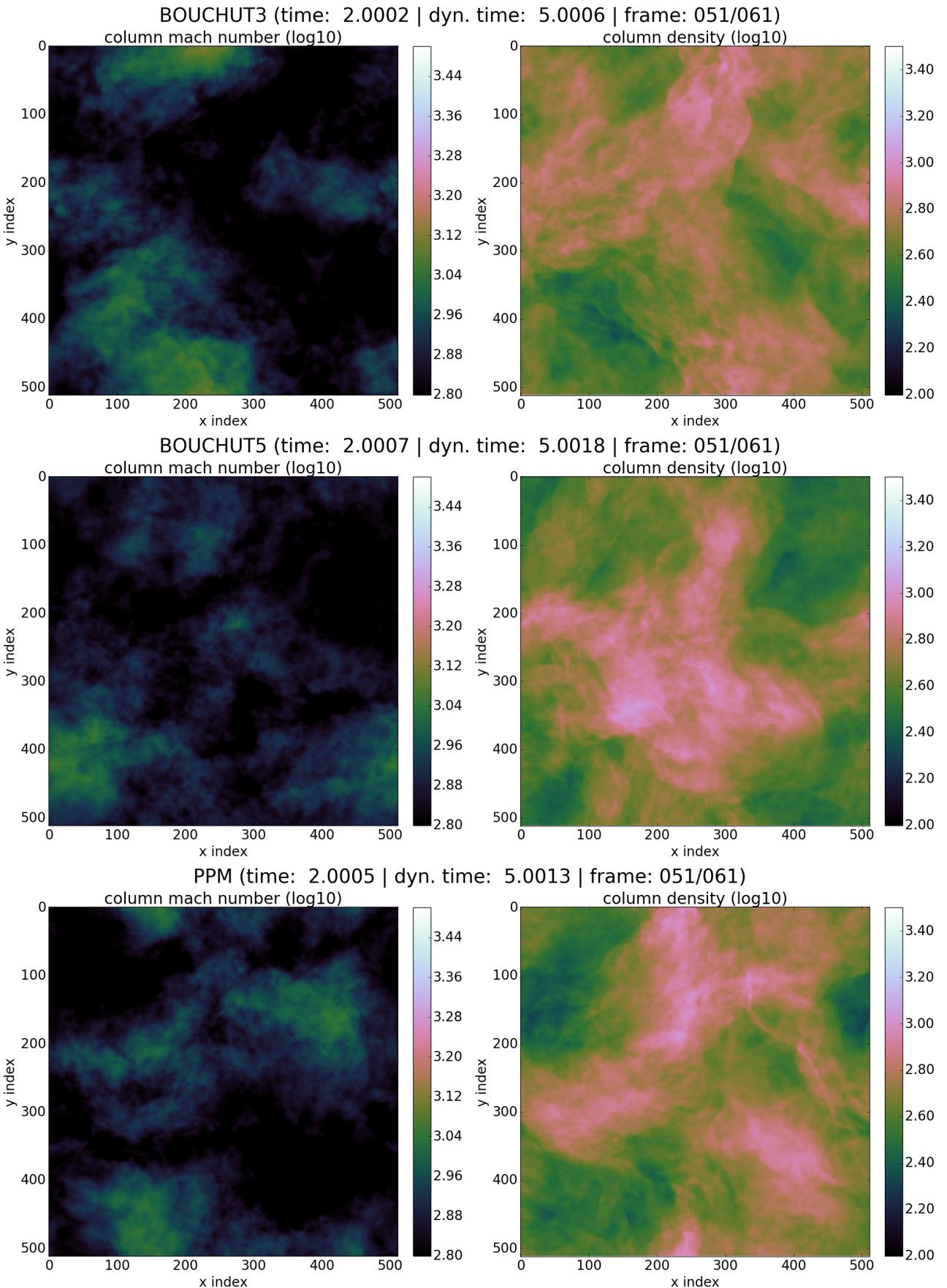


Figure 24: Column Sonic Mach Number and column density along z-axis during decay phase:
 $t_d = 5.0$.

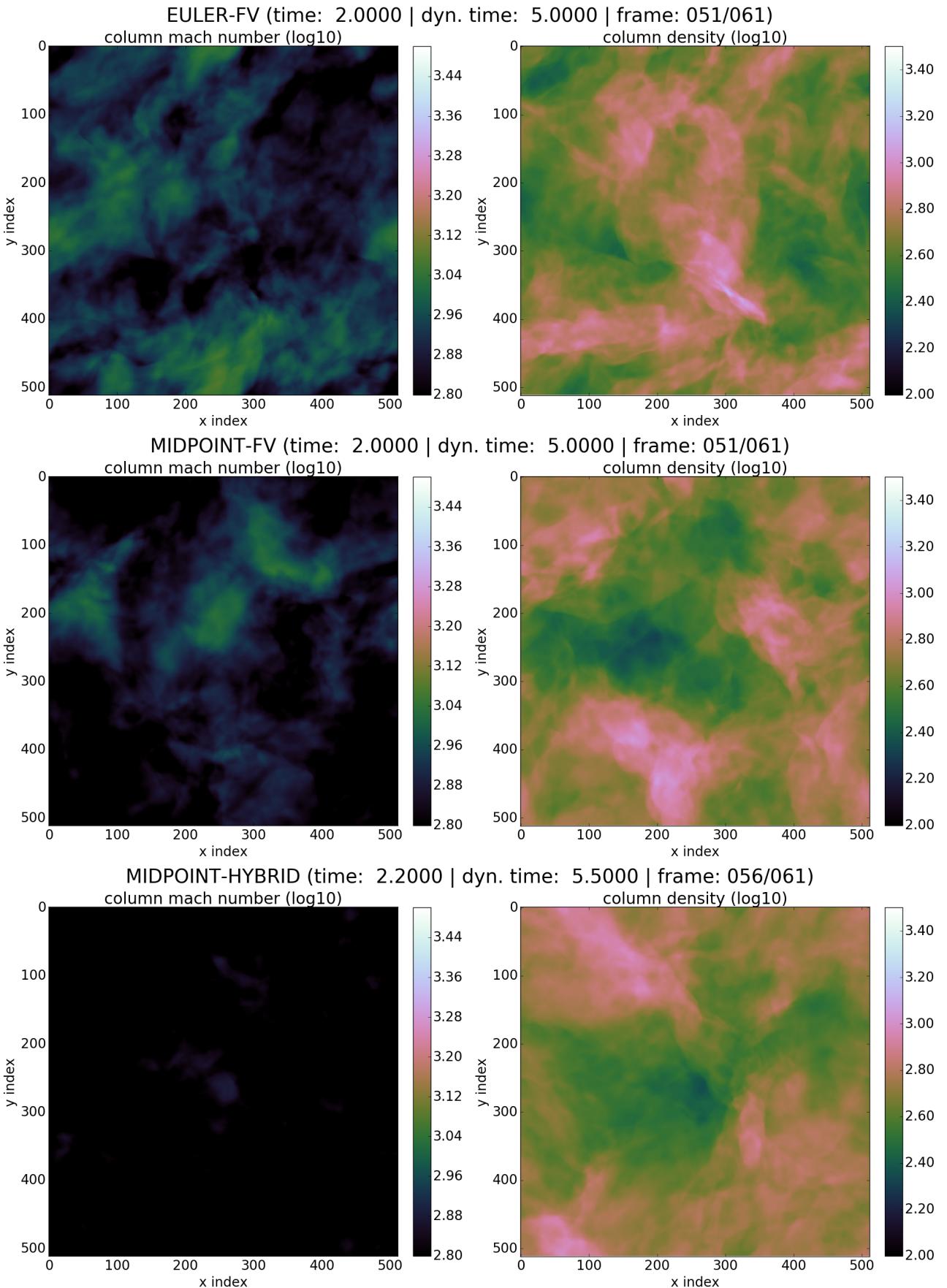


Figure 25: Column Sonic Mach Number and column density along z-axis during decay phase:
 $t_d = 5.0$.

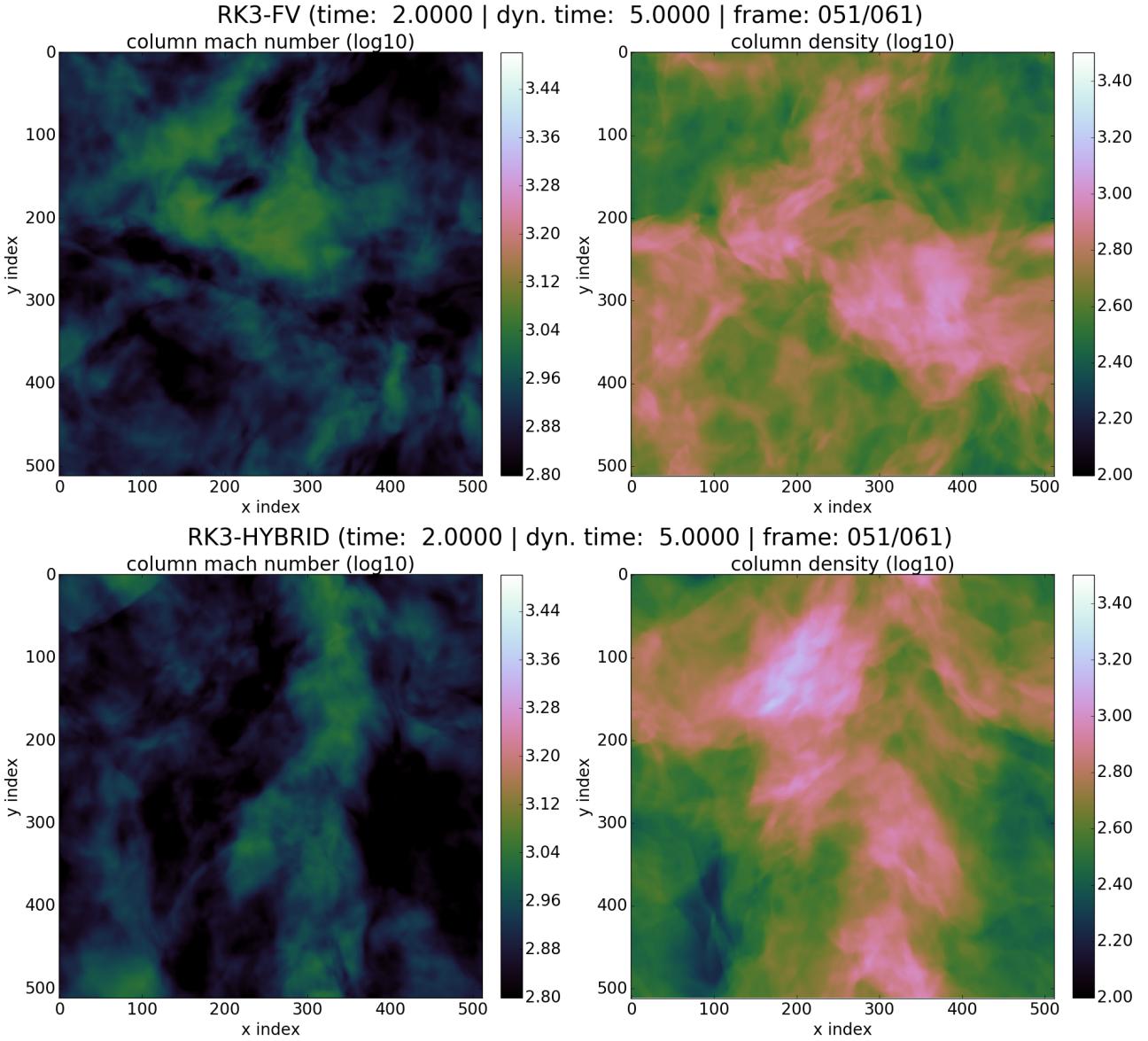


Figure 26: Column Sonic Mach Number and column density along z-axis during decay phase:
 $t_d = 5.0$.

4.2.3 FV-DG Mode Switching

Two of the conducted runs are hybrid schemes where shock capturing routines decide when a DG element must switch to FV mode in order to handle a strong shock condition correctly. Section ?? describes the procedure in detail. Fig. 27 shows the relative amount of FV elements over dynamic time. Similar to the Mach number plot (fig. 20) the three stages of turbulence are reflected in this plot. Initially, all elements are in DG mode. During the stir-up phase the FV elements quickly dominate and settle down in a range between 65% and 75% percent. After four crossing times when forcing is stopped the overall shock situation in the turbulence is relieved and fewer FV elements are needed. It finally reaches a balance of 50-50. If the simulation would go on the amount of FV elements is expected to reach 0% again.

The mode switching routines follow the Mach number with some latency. This make sense since the amount of FV elements are kind of an indicator for the intensity of the turbulence which inherently responds slowly to external changes. Consequently, even after stopping the

driver at $t_d = 4$ it takes at least half the crossing time till the ratio starts to noticeably decline.

Having only around 25% DG elements most of the time means their influence on the performance of the hybrid scheme is expected to be rather small. However fig. 27 proofs that the mode switching code operates as expected. Increasing the spatial resolution of supersonic turbulence simulations should yield even better ratios in favour of DG elements. The method is a proof-of-concept with lots of room for improvement.

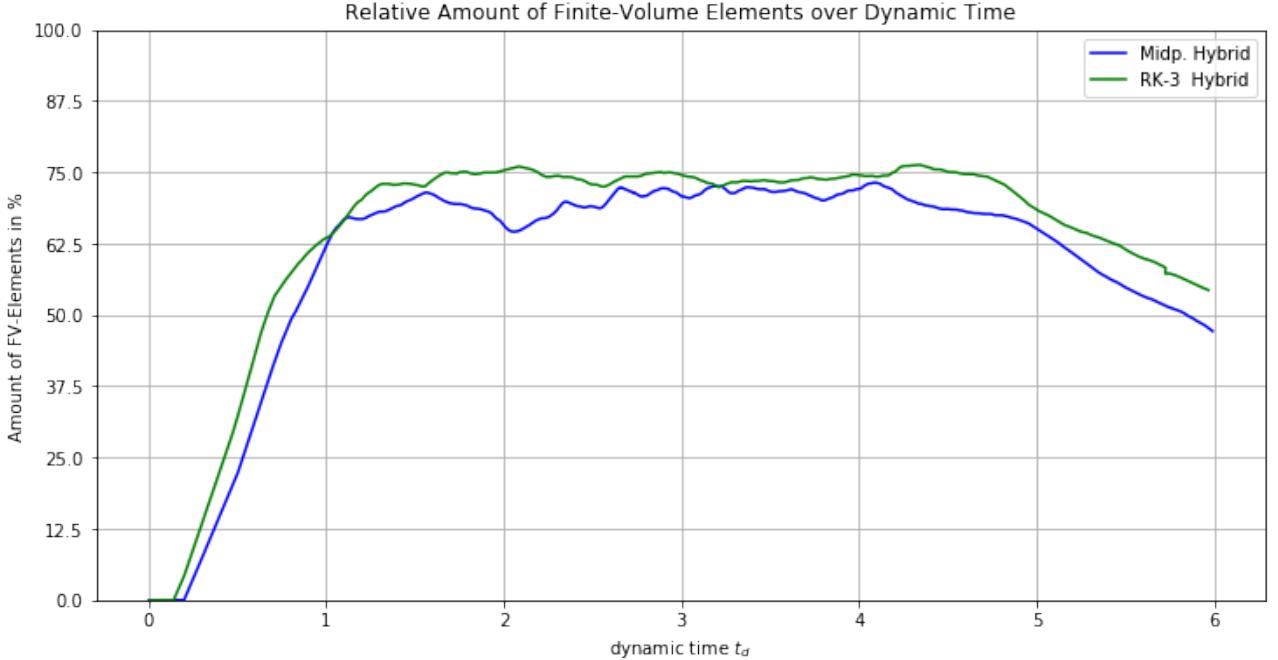


Figure 27: Time evolution of the ratio of Finite-Volume Elements to the total number of elements. The mode switching works both ways otherwise the plots would reach 100%. At $t_d = 4$ the driving stops and the amount of FV elements declines. *Remark* The other solvers would stay at 100% since they operate with Finite-Volumes only.

4.2.4 Energy Dissipation

In order to keep the temperature constant polytropic cooling deprives the system the energy which got transferred from the kinetic energy pool to the internal energy reservoir. The physical mechanics behind this have been explained in section 2.2/Energy Cascade.

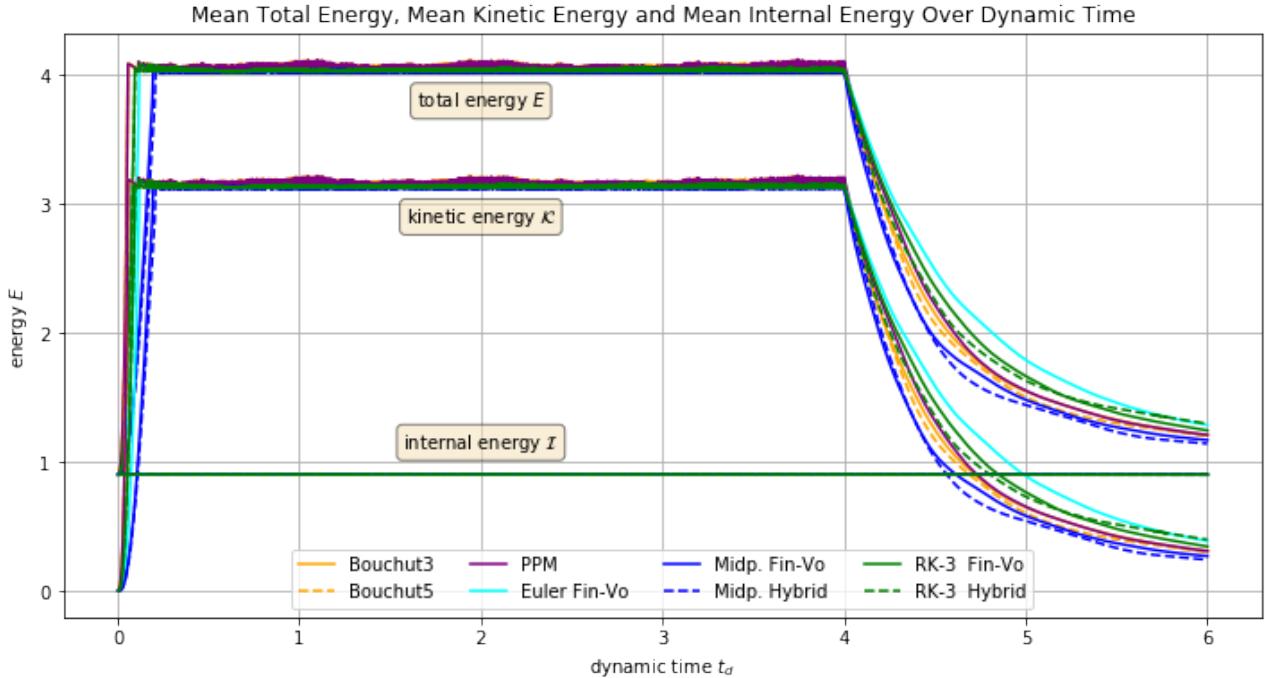


Figure 28: Time evolution of the energy in the system. The total energy E is the sum of the kinetic \mathcal{K} and internal energy \mathcal{I} : $E = \mathcal{I} + \mathcal{K}$. The internal energy is held constant via polytropic cooling (cf. section 3.1).

Now we are interested in how much kinetic energy is actually dissipated within a certain amount of time. A naive approach would be to take the numerical time derivative of the evolving kinetic energy (fig. 29) which would just yield sensible results for the decaying phase. Instead the energy loss right before and after polytropic cooling is recorded and divided by the current timestep. The resulting plot is shown in fig. 29. The rate of internal energy deprivation is equivalent to the rate of kinetic energy dissipation

$$-\frac{d\mathcal{I}}{dt_d} = -\frac{d\mathcal{K}}{dt_d} \quad (100)$$

since there is solely one energy source coming from turbulent forcing and one energy sink by cooling. During the transition from the initially constant state to turbulent state ($t_d \approx [0, 0.3]$) the dissipation rate increases rapidly and settles down to a mean rate of around $\left\langle -\frac{\mathcal{K}}{dt_d} \right\rangle = 6.5$ energy units per dynamic timestep. While most solvers follow a similar pattern up to 1.5 crossing times a change of quality seams to appear. This observation becomes even more clear in fig. 32. The conducted runs parts into two groups. The first group contains the Euler FV and both RK3 solvers. They show a balanced profile and happen to be the least dissipative ones. The second group, PPM and both Midpoints plus both Bouchuts, jump up and down in a wildly manner. Especially, the Midpoint solvers give rise to huge excesses which also reflects in the cumulative energy accounting shown in fig. 30. Furthermore, it seems that the situation would have gotten worse if the solvers would not have started to settle down due to driving stop at $t_d = 4$. During the decay phase the variations in dissipation rates are much smaller than before and even turn upside down at $t_d \approx 4.4$. An explanation of what causes the high pitched energy dissipation rates for the Midpoint methods is further investigated in the next sections.

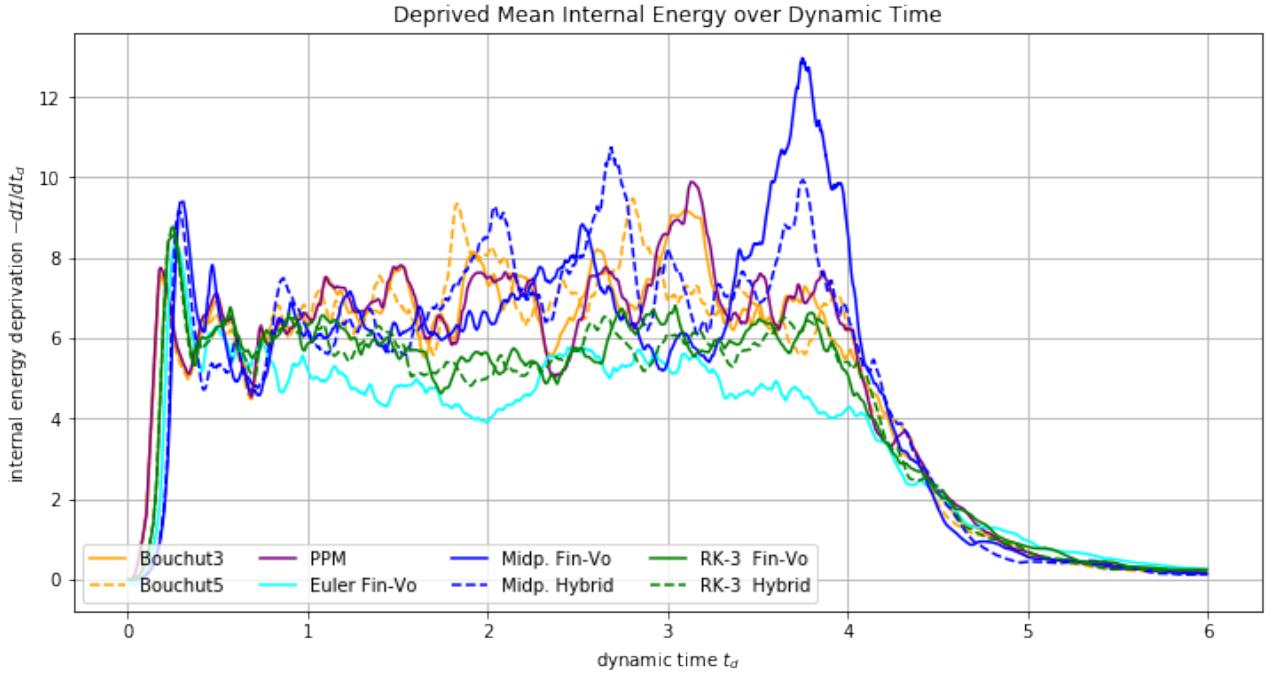


Figure 29: Time evolution of the internal energy deprivation by measuring the energy loss right before and after polytropic cooling and dividing by the current dynamic timestep.

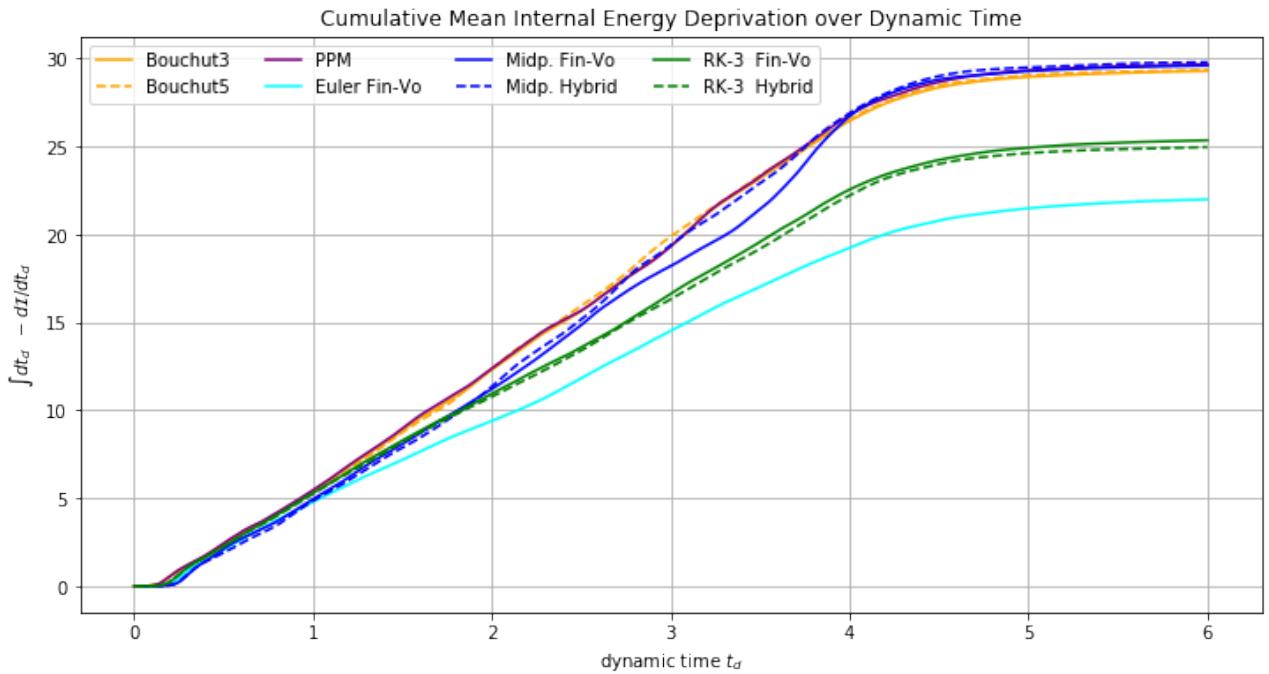


Figure 30: Cumulated energy deprivation collected by the polytropic cooling module. The plots represent the numerical integration (trapezoid rule) of the energy deprivation rates in fig. 29. By the end of the simulation circa 10 times the average energy (cf. fig. 28) was pumped through the system.

4.2.5 Density & Velocity Distributions

Fully developed supersonic turbulences are expected to reveal a lognormal density distribution (cf. section ??). Additionally, it gives an insight of the distribution of mass from smallest to largest scales. All log-log-scale density PDFs in fig. 31 show good indications for the presence

of turbulence. The dotted lines are a log-normal fits of eqn. (50) under section 2.2.2. The notorious underestimation of the fits on large scales is caused by the underresolution of the setup (cf. section ??). TODO: reference. While during the turbulent phase ($t_d = [2, 4]$) the solvers from FLASH have considerable more mass on small scales this is not the case anymore during the decay phase; exemplified by $t_d = 4.7$ and $t_d = 6.0$.

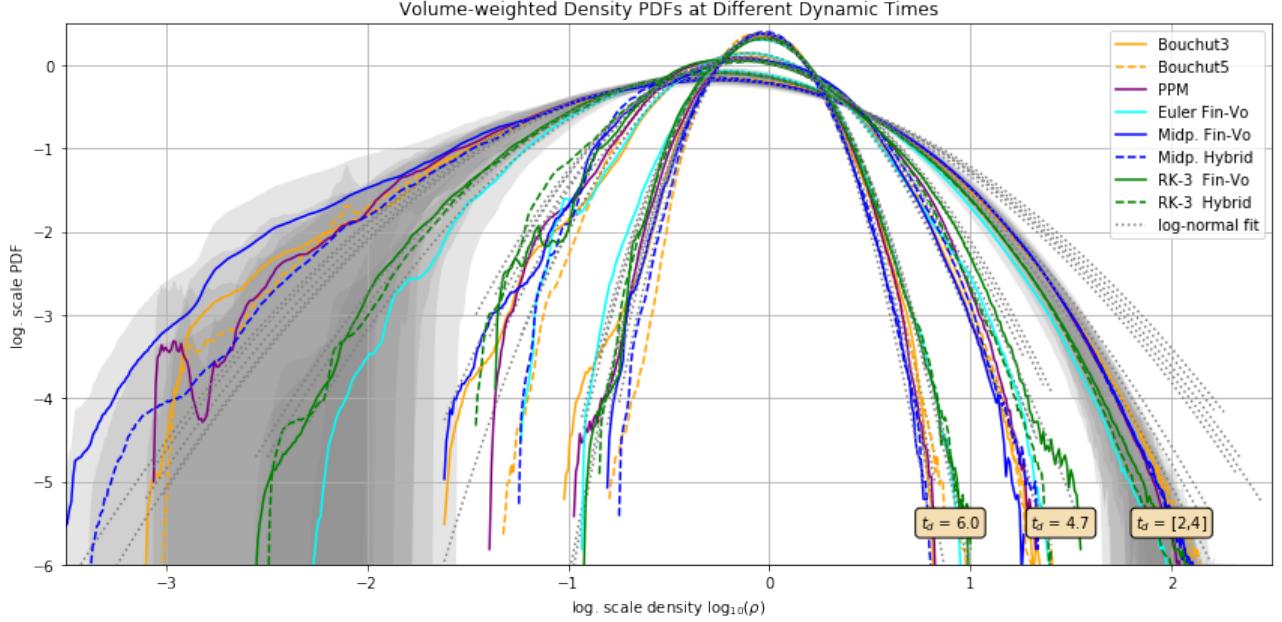


Figure 31: Log-log. scale volume-weighted density PDFs for the turbulent phase $t_d = [2, 4]$ (time-average) and two stages of the decaying phase: $t_d = 4.7$ and $t_d = 6.0$. As the decay progresses the width of the distributions decrease and the mean values return to the intial density: $\log_{10}(\rho_0) = \log_{10}(1) = 0$. The log-normal fit (dotted lines) were done with eqn. (52).

The width σ_s of the density distribution in fig. 31 are related to the sonic Mach number of the system (cf. section 2.2.2). Applying eqn. (52) and eqn. (??), where

$$b = 1 + (D^{-1} - 1)\zeta = 1 + (3^{-1} - 1)0.5 = 2/3, \quad (101)$$

we recover the Mach number of the turbulence. The result is shown in fig. 32. The plot is similar in shape to the energy dissipation in fig. 29. After two crossing times a change of quality seems to happen for some of the solvers accompanied by a lot of noise over time but better reconstruction of the true sonic Mach number of $\mathcal{M} = 2.5$. Again the Midpoint FV has a huge spike near the end of the turbulent phase ($t_d = 4$) which we take a closer look at in the next section. Euler FV and both RK3 schemes severely underestimate the Mach number by 20%. During the decay phase all solvers reveal slower values. It might be the case that during this phase the relation in eqn. (52) breaks down. Table 5 lists the fitting results of the time-averaged turbulent phase from 2 to 4 crossing times where the results seem most promising.

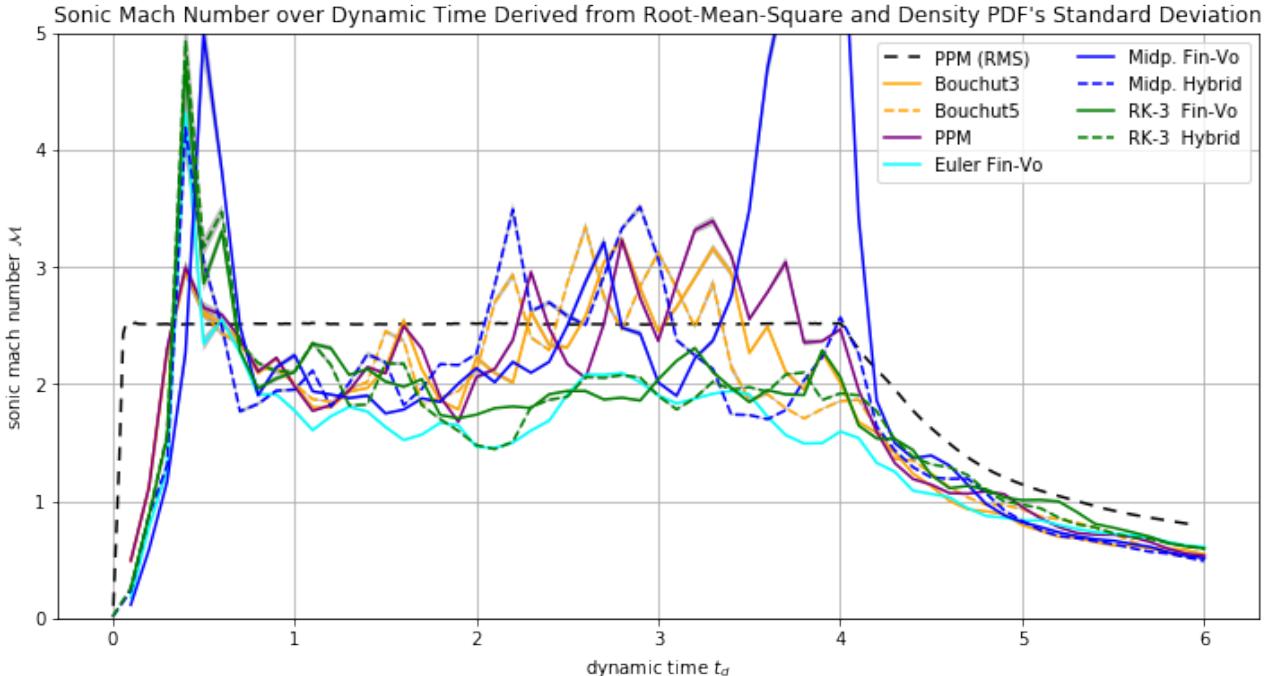


Figure 32: Time evolution of the sonic mach number over time derived from the width of the volume-weighted density PDFs of the turbulence. The dashed line represents the root-mean-square velocity of the PPM solver and serves as a reference. Due to the lognormal relationship the fit yielded very low error margins. Hence, the error bars are to small to see.

Table 5: Comparison of Gaussian Fitting Results of the Log-normal Density PDF averaged over the turbulent phase: $t_d = [2, 4]$. The PDF Mach number M_{PDF} of the Midpoint FV overshoots due to the influence of the runaway.
Compare fig. 32.

Solver	Mean s_0	Std. Deviation σ_s	RMS Mach M_{RMS}	PDF Mach M_{PDF}
Bouchut3	-0.31 ± 0.05	0.58 ± 0.06	2.63 ± 0.13	2.5 ± 0.4
Bouchut5	-0.29 ± 0.06	0.57 ± 0.08	2.59 ± 0.09	2.4 ± 0.6
PPM	-0.32 ± 0.05	0.59 ± 0.06	2.62 ± 0.12	2.6 ± 0.5
Euler Fin-Vo.	-0.24 ± 0.03	0.47 ± 0.04	2.56 ± 0.04	1.8 ± 0.2
Midp. Fin-Vo.	-0.31 ± 0.04	0.60 ± 0.09	2.80 ± 0.40	2.7 ± 0.8
Midp. Hybrid	-0.32 ± 0.05	0.56 ± 0.07	2.81 ± 0.18	2.4 ± 0.5
RK-3 Fin-Vo.	-0.26 ± 0.02	0.50 ± 0.03	2.61 ± 0.06	2.0 ± 0.2
RK-3 Hybrid	-0.26 ± 0.03	0.48 ± 0.04	2.55 ± 0.04	1.9 ± 0.3

Velocity and density distributions are an important tool in order to compare supersonic turbulences. An exceeding dispersion of velocity PDFs for example indicates strong bulk motions within the medium. Bulk motion distorts the turbulence by drawing off kinetic energy from the turbulence while at the same time the root-mean-square velocity of the system remains constant. In the worst case the energy cascades breaks down or does not even develop in the first place. Fig. 33 lacks any considerable dispersion which means the bulk motion correction (cf. section ??) is working correctly. More distinctive is the bulge of high velocities brought out by the Midpoint schemes with accompanied by huge error margins. This means that during the turbulent phase there must be extreme velocity fluctuations up to Mach 30 and beyond. Fig. 34 affirms this observation. Comparing this plot with the energy dissipation rates in fig. 29 there is a correlation between maximum velocity bursts and high pitched dissipation. However this applies only to the Midpoint schemes. The maximum velocity plot indicates only a singular

value which does not necessarily represents a prevalent phenonen in the system. In conjunction with the velocity PDF however this can be considered true for the Midpoint schemes.

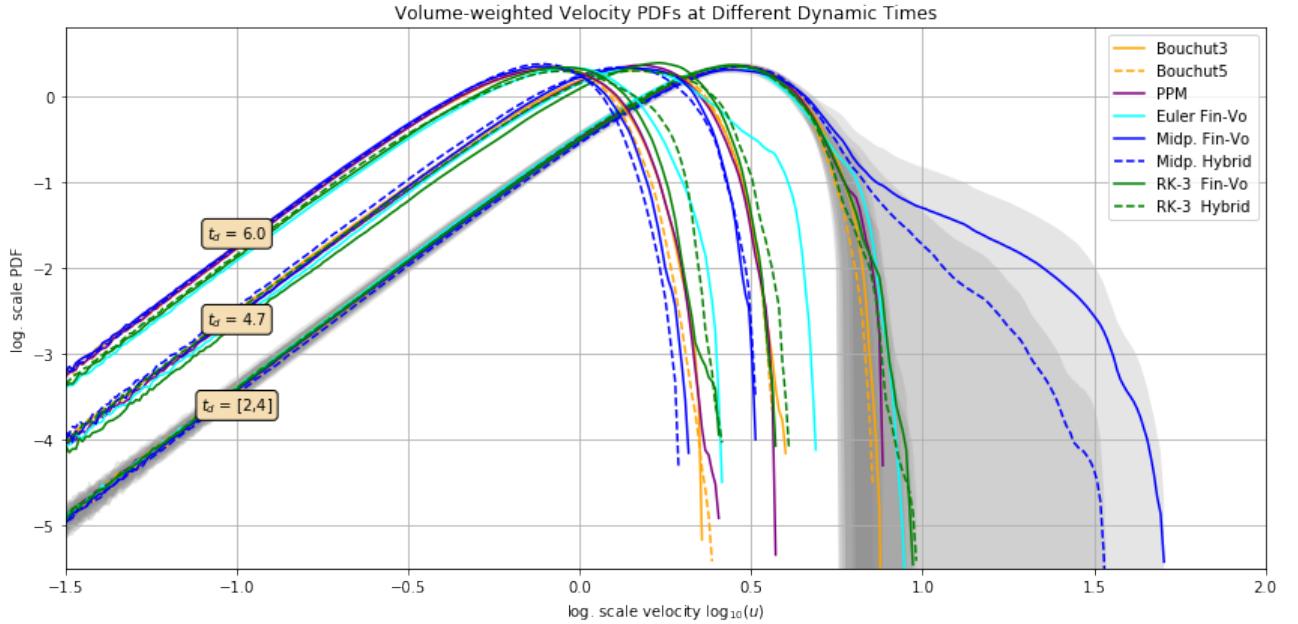


Figure 33: Volume-weighted velocity PDFs computed from the checkpoint files for the turbulent phase $t_d = [2, 4]$ (time-average) and two stages of the decaying phase: $t_d = 4.7$ and $t_d = 6.0$. The faint grey area marks the margin of error of the time-averaging. The Midpoint schemes yield strong fluctuation of high velocities during the turbulent phase (cf. fig. 34).

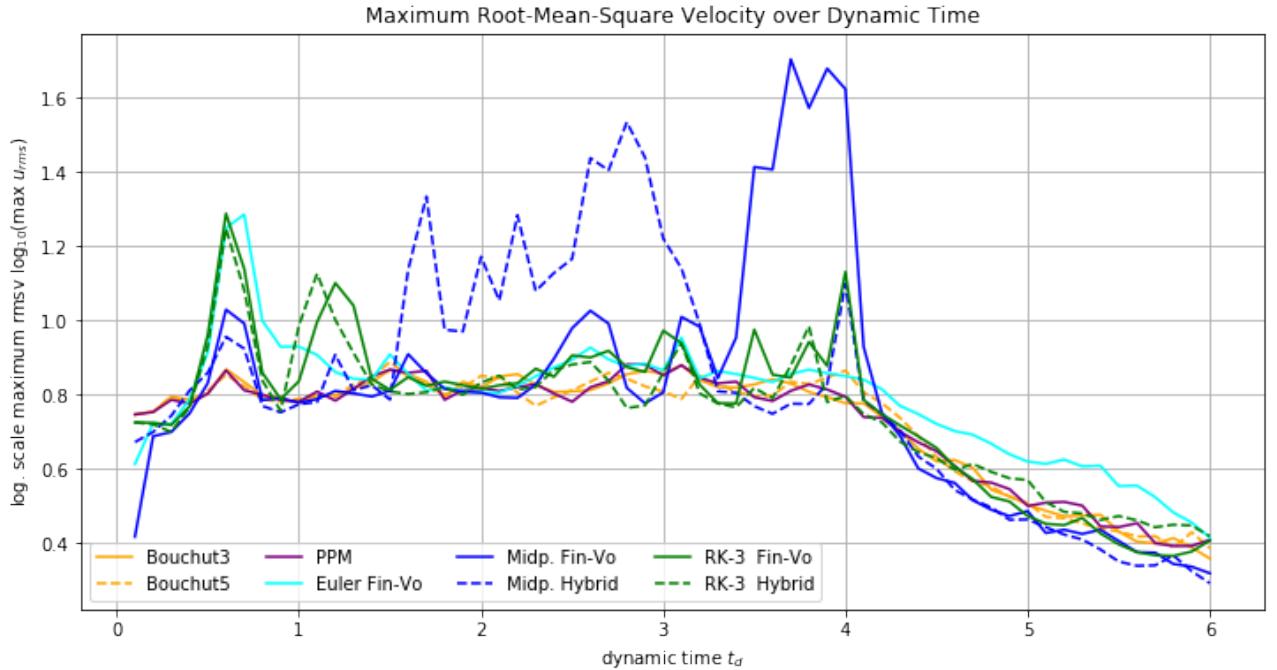


Figure 34: Maximum root-mean-square velocity over time retrieved from the checkpoint files. The Midpoint schemes show very strong maximum velocity bursts up to Mach 30 and beyond during the turbulent phase.

As already discussed under section 4.2.2 the Midpoint FV method shows a unique pattern of slow-moving mass lumps and regions of highly accelerated but very thin gas. When mass-

weighting the velocity distribution, see fig. 35, we get another picture. The bulge on the right side of fig. 33 completely disappears which supports the observation of highly accelerated thin gas in the Midpoint FV run. By contrast, Bouchut 3, Bouchut 5 and PPM show a richer structure in the slow moving range with a higher mass distribution.

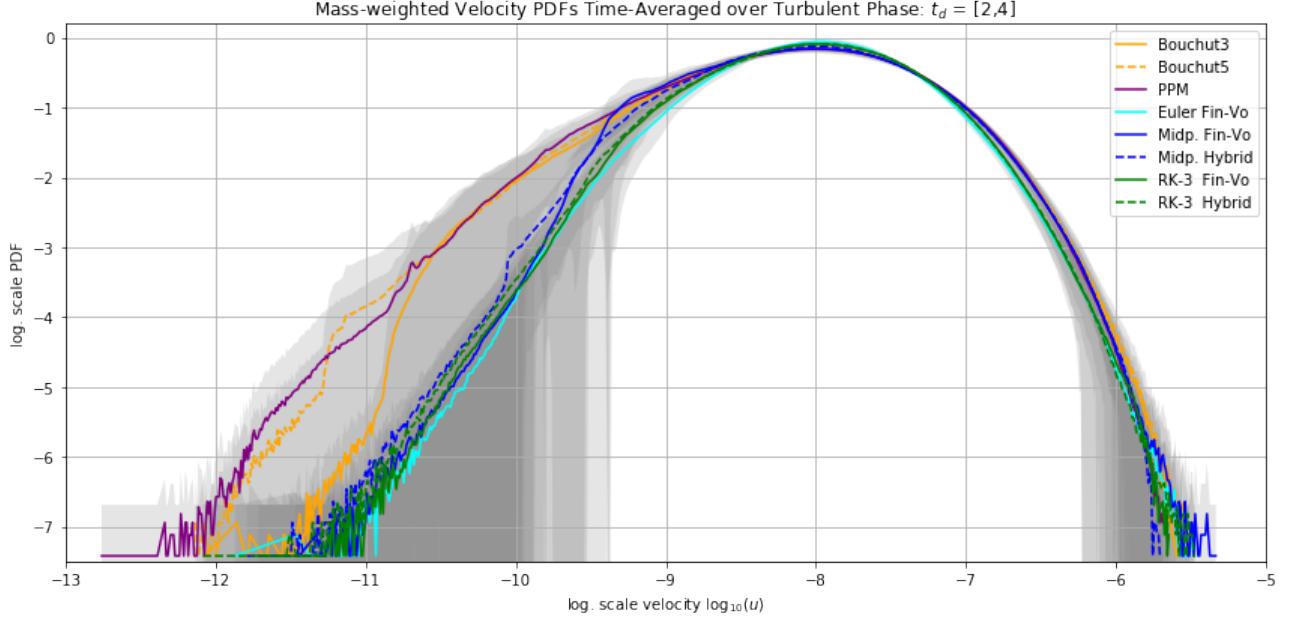


Figure 35: Log-log-scale mass-weighted velocity PDF time-averaged over the turbulent phase ($t_d = [2, 4]$). The gray areas mark the margin of error. The solvers from FLASH have a greater amount of slowly moving mass than the others. The bulge on the right side in fig. 33 is gone due to a small mass distribution in the high-velocity range.

4.2.6 Energy & Velocity Powerspectra

Powerspectra give an insight of the distribution of a physical quantity from large to small spatial scales. In this section we discuss the energy and velocity powerspectra which were introduced in section 2.2.1. The powerspectrum is divided in three ranges. In the driving or large scale range the energy injection from the turbulent driver takes place. It spans over the first three to four modes. From there the energy trickles down to smaller and smaller scales, called energy cascade, which takes place in the inertial range. Finally it reaches the dissipative range which begins in our case at roughly $k = 32$. Here the kinetic energy dissipates and gets transformed in to internal energy. In the powerspectrum of the kinetic energy, fig. 36, the boundaries of the mentioned ranges are drawn in. According to Parseval's theorem the normed area under the powerspectrum is equal to the total square of the real-valued quantity (cf. section 2.2.1). In fig. 36 it then be the total squared kinetic energy. Table 6 proofs via numerical integration that this is indeed true within the margins of error. In agreement with the discussion of column densities and the density distributions in the previous sections the solutions provided by FLASH's solvers contain more energy on small scales than those in FLEXI. To quantify this observation the area under the dissipative range was estimated and compared to the total squared kinetic energy. Consulting the right-most column in table 6 the relative amount of energy on small scales ranges from 2.4% (RK3 FV) to 5.7% (Bouchut 5).

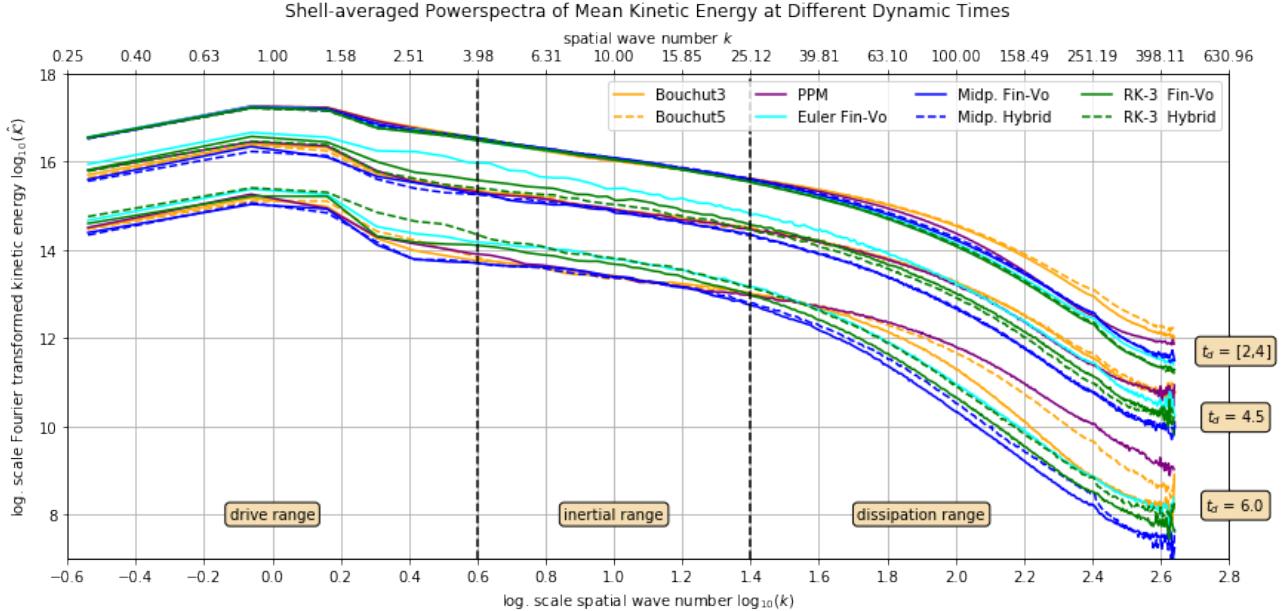


Figure 36: Powerspectra of the volume-weighted mean kinetic energy field shown for the turbulent phase $t_d = [2, 4]$ (time-average) and two stages of the decay phase: $t_d = 4.5$ and $t_d = 6.0$. When the total kinetic energy declines the powerspectrum shifts downwards.

Besides the lognormality of the density PDF (section ??) the slopes of the inertial range of velocity powerspectra are well-established indicators for the accurate modelling of turbulences. According to the theory (cf. section ??) we expect slopes of $m_{vw} = -2.1$ and $m_{mw} = -5/3$ for the volume-weighted and mass-weighted powerspectra of the three-dimensional velocity field, respectively. Fig. 37 and fig. 38 indeed reproduce given slopes for the time-averaged powerspectrum of the turbulent phase. The spectra at the end of the simulation when the turbulent decay as progressed considerably are plotted as well. Obviously, the inertial range has levelled down. This makes sense considering the fact that the supply of kinetic energy was cut off. Hence the large scale structures gradually dissolve and push their energy down the cascade. The change of slope can be seen in fig. 39 very clearly. In the beginning the inertial range is very steep. The turbulent driver is pumping energy into the system on first three modes. As the simulation continues the energy cascade builds up and is fully established after two crossing times. When entering the decay phase ($t_d = 4$) the slope flattens out. Table 6 lists the results of the linear fits of the inertial range as well as the numerically integrated areas under the time-averaged powerspectra of the turbulent phase.

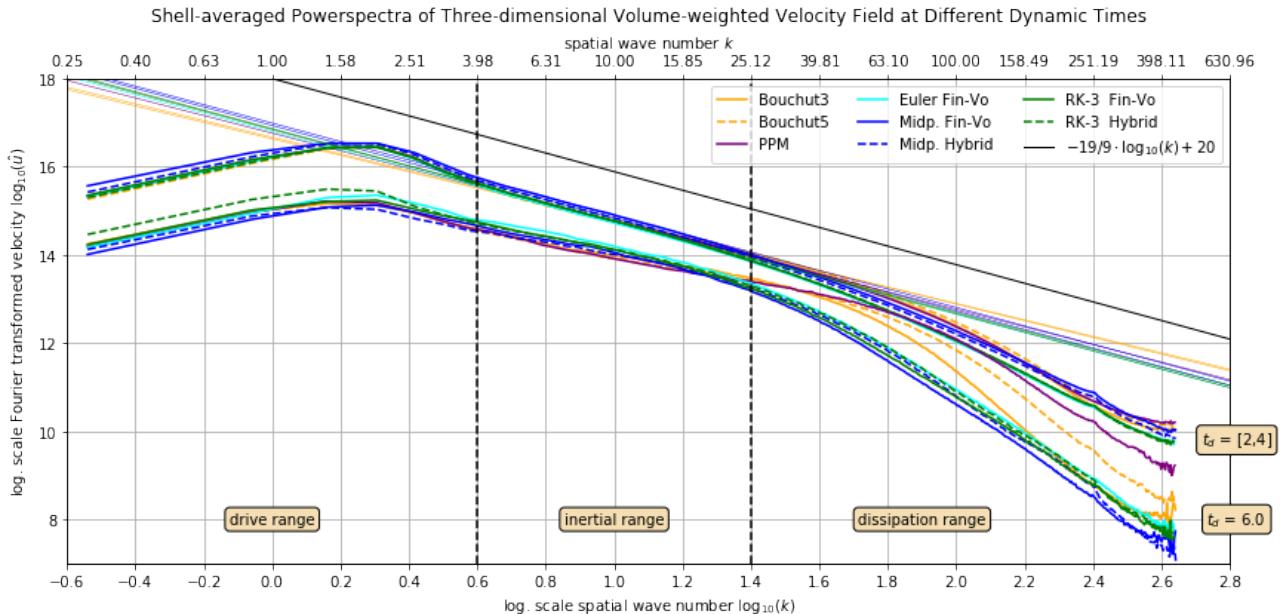


Figure 37: Powerspectra of the volume-weighted velocity field for the turbulent phase $t_d = [2, 4]$ (time-average) and at the end of the decaying phase $t_d = 6.0$.

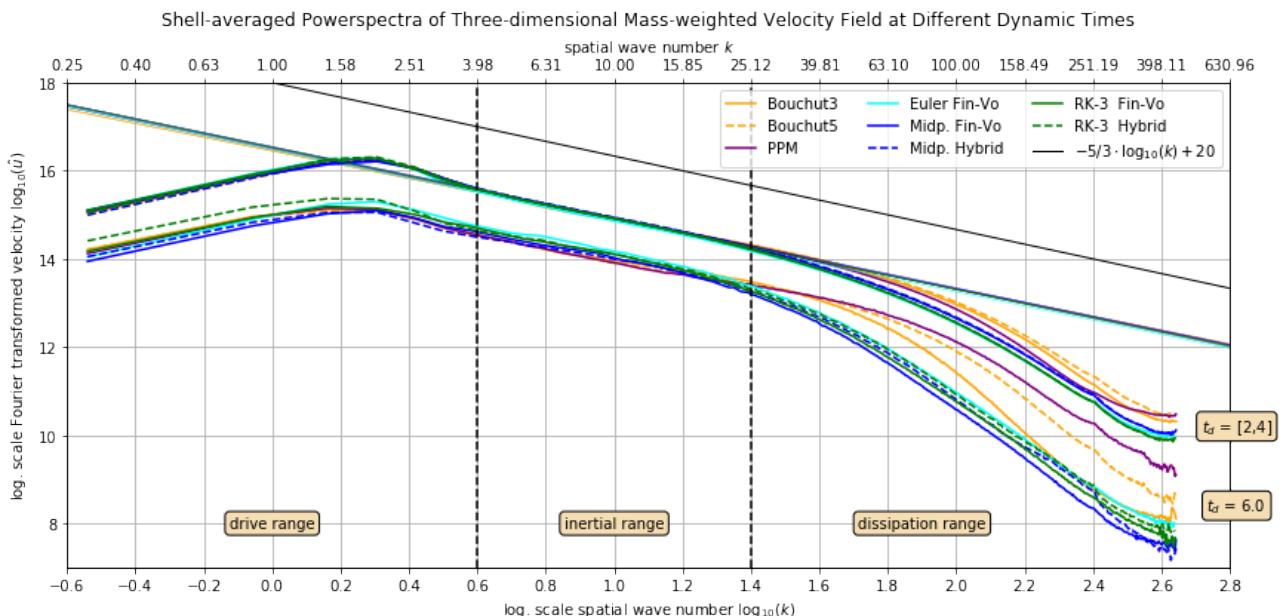


Figure 38: Powerspectra of the mass-weighted velocity field for the turbulent phase $t_d = [2, 4]$ (time-average) and at the end of the decaying phase $t_d = 6.0$.

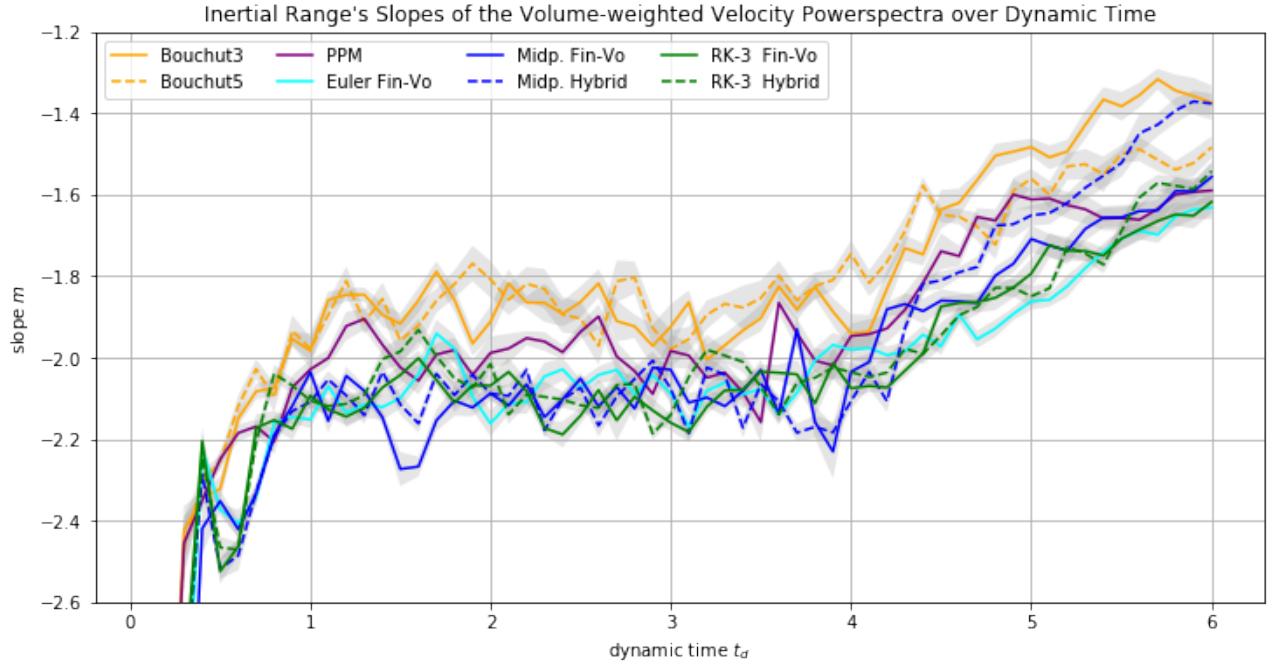


Figure 39: Time evolution of the inertial range's slope of the volume-weighted velocity powerspectra. See fig. 37.

Table 6: Comparison of the three types of powerspectra averaged over the turbulent phase: $t_d = [2, 4]$

Solver	Slope m	Offset n	Mean Squ. $\langle \cdot^2 \rangle$	Area A	Diss. Area A_{diss}	$A_{diss}/A [\%]$
Volume-weighted Mean Kinetic Energy						
Bouchut3	-1.16 ± 0.05	17.19 ± 0.24	39 ± 4	37 ± 8	2.00 ± 0.60	5.4 ± 1.7
Bouchut5	-1.19 ± 0.10	17.22 ± 0.30	40 ± 5	37 ± 8	2.10 ± 0.70	5.7 ± 1.8
PPM	-1.14 ± 0.07	17.19 ± 0.24	38 ± 4	36 ± 8	1.37 ± 0.33	3.8 ± 0.9
Euler Fin-Vo.	-1.06 ± 0.04	17.11 ± 0.30	35 ± 5	34 ± 9	0.88 ± 0.35	2.6 ± 1.0
Midp. Fin-Vo.	-1.10 ± 0.05	17.18 ± 0.27	37 ± 6	35 ± 9	1.00 ± 0.50	3.0 ± 1.3
Midp. Hybrid	-1.06 ± 0.02	17.15 ± 0.17	36 ± 4	34 ± 7	1.12 ± 0.33	3.3 ± 1.0
RK-3 Fin-Vo.	-1.07 ± 0.08	17.12 ± 0.31	33 ± 4	32 ± 8	0.75 ± 0.28	2.4 ± 0.9
RK-3 Hybrid	-1.17 ± 0.17	17.23 ± 0.34	33 ± 3	32 ± 6	0.81 ± 0.25	2.5 ± 0.8
Volume-weighted Velocity						
Bouchut3	-1.89 ± 0.13	16.67 ± 0.28	4.00 ± 0.40	4.0 ± 0.9	0.0152 ± 0.0034	0.38 ± 0.09
Bouchut5	-1.87 ± 0.11	16.63 ± 0.23	3.86 ± 0.26	3.8 ± 0.7	0.0179 ± 0.0035	0.47 ± 0.09
PPM	-2.00 ± 0.15	16.76 ± 0.31	4.00 ± 0.40	4.0 ± 0.9	0.0143 ± 0.0029	0.36 ± 0.07
Euler Fin-Vo.	-2.08 ± 0.10	16.80 ± 0.15	3.86 ± 0.16	3.8 ± 0.6	0.0068 ± 0.0012	0.18 ± 0.03
Midp. Fin-Vo.	-2.09 ± 0.05	17.00 ± 0.70	6.00 ± 4.00	6.0 ± 0.5	0.0150 ± 0.0130	0.24 ± 0.22
Midp. Hybrid	-2.11 ± 0.03	16.94 ± 0.21	4.80 ± 0.90	4.8 ± 1.4	0.0102 ± 0.0030	0.21 ± 0.06
RK-3 Fin-Vo.	-2.10 ± 0.08	16.87 ± 0.16	3.96 ± 0.17	3.9 ± 0.6	0.0074 ± 0.0015	0.19 ± 0.04
RK-3 Hybrid	-2.08 ± 0.04	16.85 ± 0.12	3.79 ± 0.14	3.9 ± 0.6	0.0073 ± 0.0015	0.19 ± 0.04
Mass-weighted Velocity						
Bouchut3	-1.56 ± 0.10	16.45 ± 0.19	3.158 ± 0.021	3.1 ± 0.4	0.054 ± 0.010	1.72 ± 0.33
Bouchut5	-1.57 ± 0.13	16.45 ± 0.22	3.158 ± 0.024	3.1 ± 0.4	0.060 ± 0.011	1.90 ± 0.40
PPM	-1.59 ± 0.07	16.49 ± 0.17	3.166 ± 0.020	3.1 ± 0.5	0.043 ± 0.007	1.38 ± 0.24
Euler Fin-Vo.	-1.61 ± 0.07	16.48 ± 0.15	3.134 ± 0.004	3.1 ± 0.4	0.021 ± 0.004	0.69 ± 0.13
Midp. Fin-Vo.	-1.60 ± 0.04	16.52 ± 0.15	3.128 ± 0.006	3.0 ± 0.4	0.028 ± 0.009	0.93 ± 0.29
Midp. Hybrid	-1.60 ± 0.03	16.55 ± 0.11	3.133 ± 0.006	3.2 ± 0.5	0.029 ± 0.005	0.92 ± 0.17
RK-3 Fin-Vo.	-1.61 ± 0.05	16.52 ± 0.12	3.137 ± 0.011	3.2 ± 0.4	0.022 ± 0.004	0.70 ± 0.13
RK-3 Hybrid	-1.62 ± 0.12	16.54 ± 0.19	3.138 ± 0.008	3.3 ± 0.5	0.022 ± 0.004	0.68 ± 0.13

4.2.7 Summary

In the driven supersonic turbulence setup we analyzed the various solutions of FLASH's and FLEXI's solver over the course of 6 crossing times. All runs consisted of three phases or stages. In the first phase the initially restful medium was stirred up by turbulent forcing to the root-mean-square velocity of $\mathcal{M} = 2.5$. After reaching this limit the driver kept the kinetic energy stable for 4 crossing times so that a turbulent energy cascade develops. Following this turbulent phase forcing was deactivated and the turbulence decayed for another 2 crossing times.

By analyzing density/velocity distributions and powerspectra it was confirmed that all solvers modelled a supersonic turbulence correctly. They confirmed a lognormal density profile and their inertial ranges descended with the correct slopes. However there are differences regarding energy dissipation rates and resolution of smallest scales revealing an inverse relationship. As cumulated energy budget in fig. 30 suggests Euler FV burnt the least amount of kinetic energy contrasting PPM, Bouchut 3 and Bouchut 5 who consumed roughly 15 to 20% more. On the other hand the latter yield more details on small scales which becomes clearly evident in the mass-weighted velocity distribution in fig. 35. Estimating the energy in the dissipative range, see table ??, also supports this observation.

The behaviour of the Midpoint schemes falls out of place and should be discussed separately. It seems that the turbulent driver creates a force field in such a unique way and over a longer period of time so that the majority of the mass is dragged into a lump surrounded by highly accelerated regions of very thin medium. See fig. 22 for a snapshot. The dissolution of those extremely high Mach regions drives up the energy dissipation substantially. See fig. 29. This remarkable example reveals the imprinting influence of the driver on the turbulence which is a problem in its own right.

In order to gain insight on how strong the influence of the turbulent driver might be we also conducted a decaying turbulence simulation with exact equal initial conditions. The results are described in the next section.

4.3 Decaying Turbulence

In the previous section we conducted a driven turbulence simulation where we identified a significant influence of the turbulent driver on the modelled turbulence. Consequently, we probe each solver with a decaying turbulence setup of exact equal initial conditions. A snapshot of a fully developed Mach-10 turbulence, generated by the Bouchut5solver provides the basis for the simulations. The initial turbulence setup is left to decay six turning times relative to $\mathcal{M} = 10 \Rightarrow T_{turn} = 1/10 = 0.1$. In order to avoid interpolation errors and unsolicited oscillations in the DG code the initial state was slightly smoothed beforehand via application of a Gaussian blur (cf. fig. 13). Apart from that the setup is identical to the driven turbulence setup detailed at the beginning of section 4.2. Table table 7 gives an overview of the conducted experiments.

Table 7: Outline of the Decaying Turbulence Setup. Note the small CFL number for PPM which turned out to be necessary in order not to crash. There is a slight variance among the initial Mach numbers/kinetic energies. They stem from the fact that these values were measured when the simulation started and are subject to minor numerical errors.

Solver	B3	B5	PPM	EU-FV	MP-FV	MP-HY	RK3-FV	RK3-HY
CFL number	0.8	0.8	0.1	0.4	0.8	0.9	0.9	1.2
initial mach number \mathcal{M}_0	9.877	9.877	9.877	9.875	9.872	9.873	9.871	9.871
initial kin. energy \mathcal{K}_0	48.78	48.78	48.78	48.76	48.72	48.73	48.72	48.72
sim. end $t_{\text{sim. end}}$	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
nr. checkpoints. $N_{\text{checkpoints}}$	30	30	30	30	30	30	30	30

Since the setup is very similar to the driven turbulence the discussion is kept short and is arranged in the same manner. The author advises to follow the the discussion in the previous section before moving on.

4.3.1 Mach Number Evolution

In contrast to the different decay rates in fig. 20 the declines of the Mach number over time (fig. 40) are unexpectedly alike. The decaying turbulences start at $\mathcal{M} = 10$ and take six crossing times to drop below the supersonic regime ($\mathcal{M} = 1$).

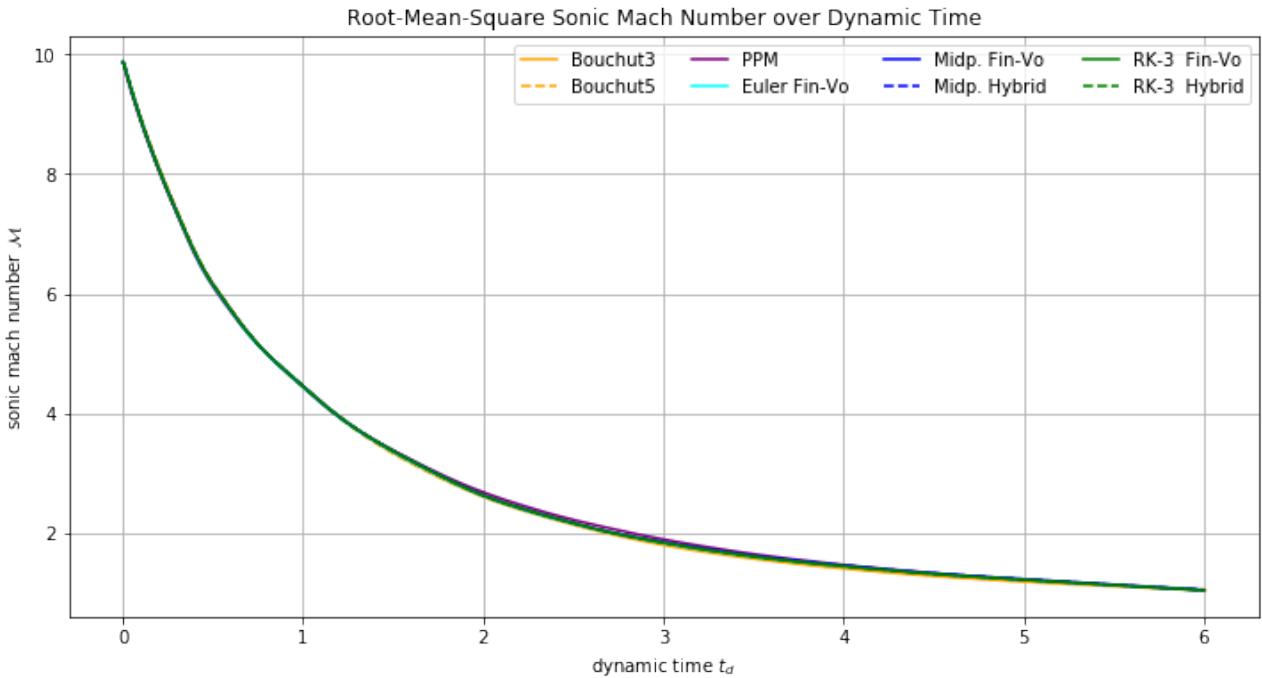


Figure 40: Time evolution of the root-mean-square mach number.

4.3.2 Column Density & Velocity

Here we present a series of column density and velocity snapshots of all conducted simulations at three successive stages of turbulent decay: $t_d = [1, 2, 3]$ with $\mathcal{M} = [4.2, 2.4, 1.9]$, respectively. See fig. 41 to fig. 49. The exact point of time, the solver and the Mach number are attached to each snapshot in the title and the figure caption.

First of all the large scale structures are congruent among all solvers comparing the snapshots depicting equal points in time. As already discussed in section ?? dissipation happens only at small scales. And since the velocity field is not influenced by external forces the large scale movements are conserved.

The originally smoothed initial state (cf. fig. 13) reestablished the shocks which can be clearly seen by the sharp edges of the filamentary structures in fig. 41 to fig. 43. As expected those structures gradually dissolve with increasing time which leads to a transformation of kinetic energy into internal energy. It then escapes the system via polytropic cooling. This process is indicated by the increased darkening of the column Mach number plots as time progresses.

It is impossible to tell from the snapshots if there are differences in the resolution of small scales. In analogy to the driven turbulence setup density PDFs and powerspectra will give a better insight into this matter.

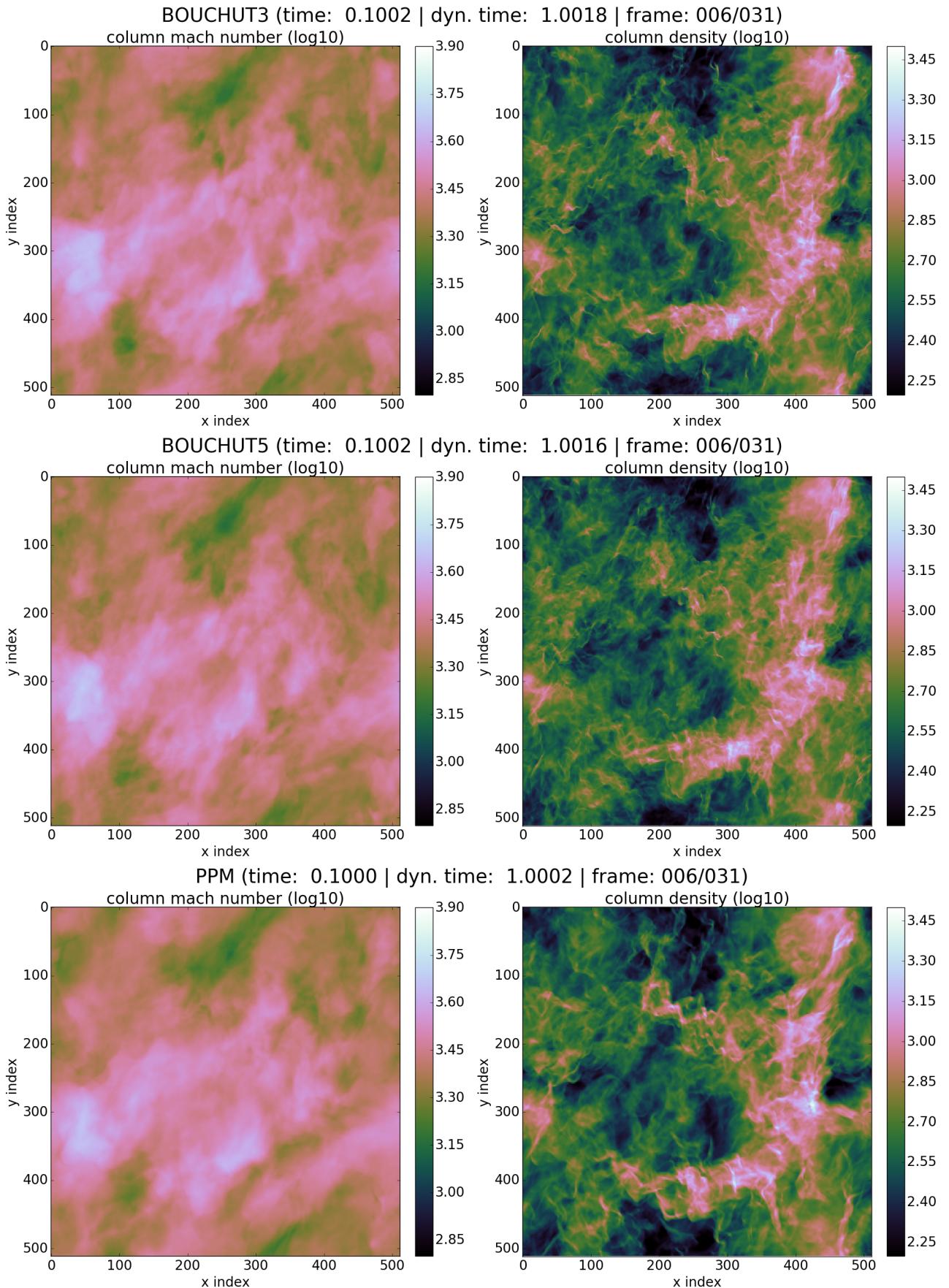


Figure 41: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 1.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 4.2$

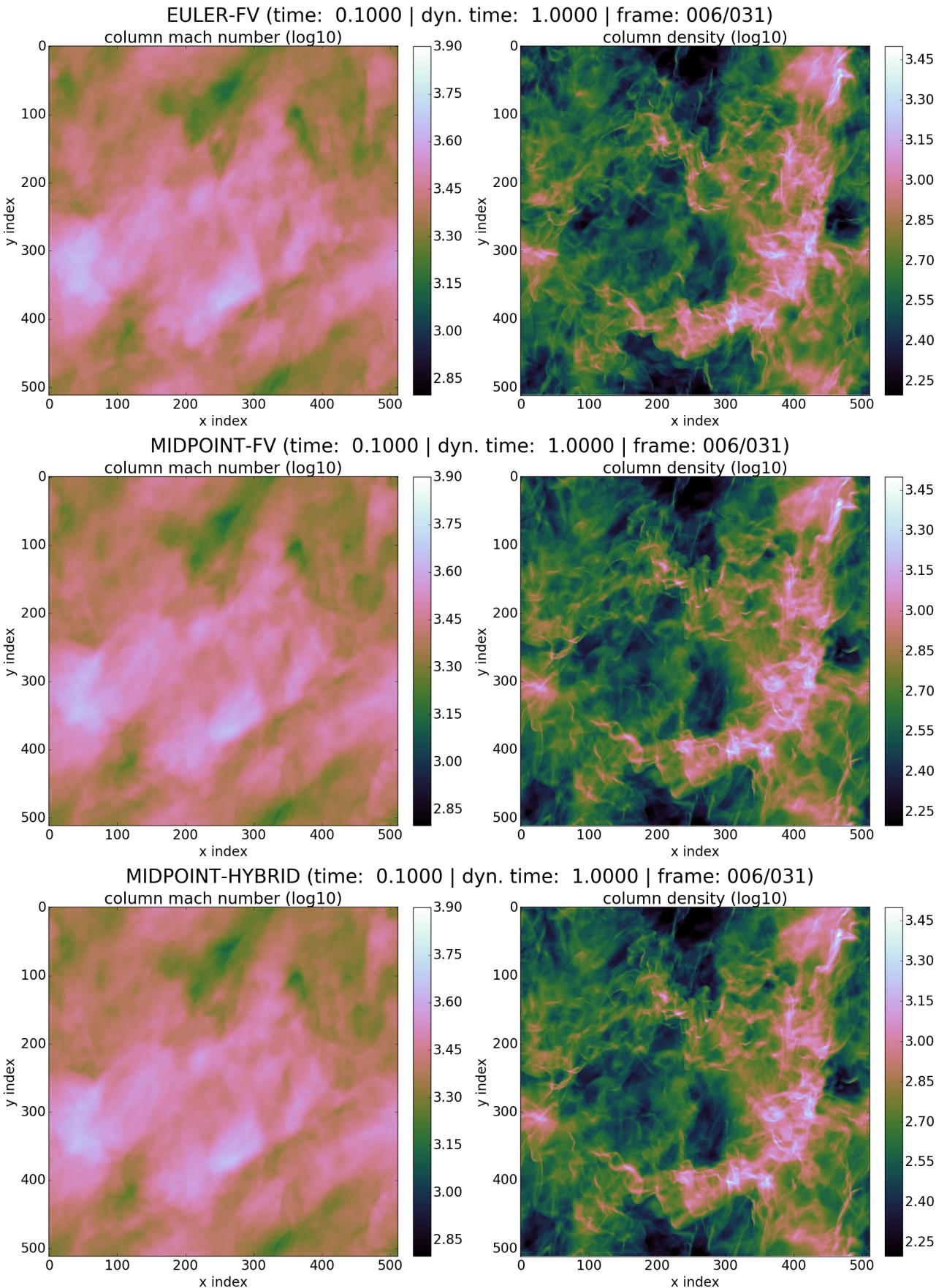


Figure 42: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 1.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 4.2$

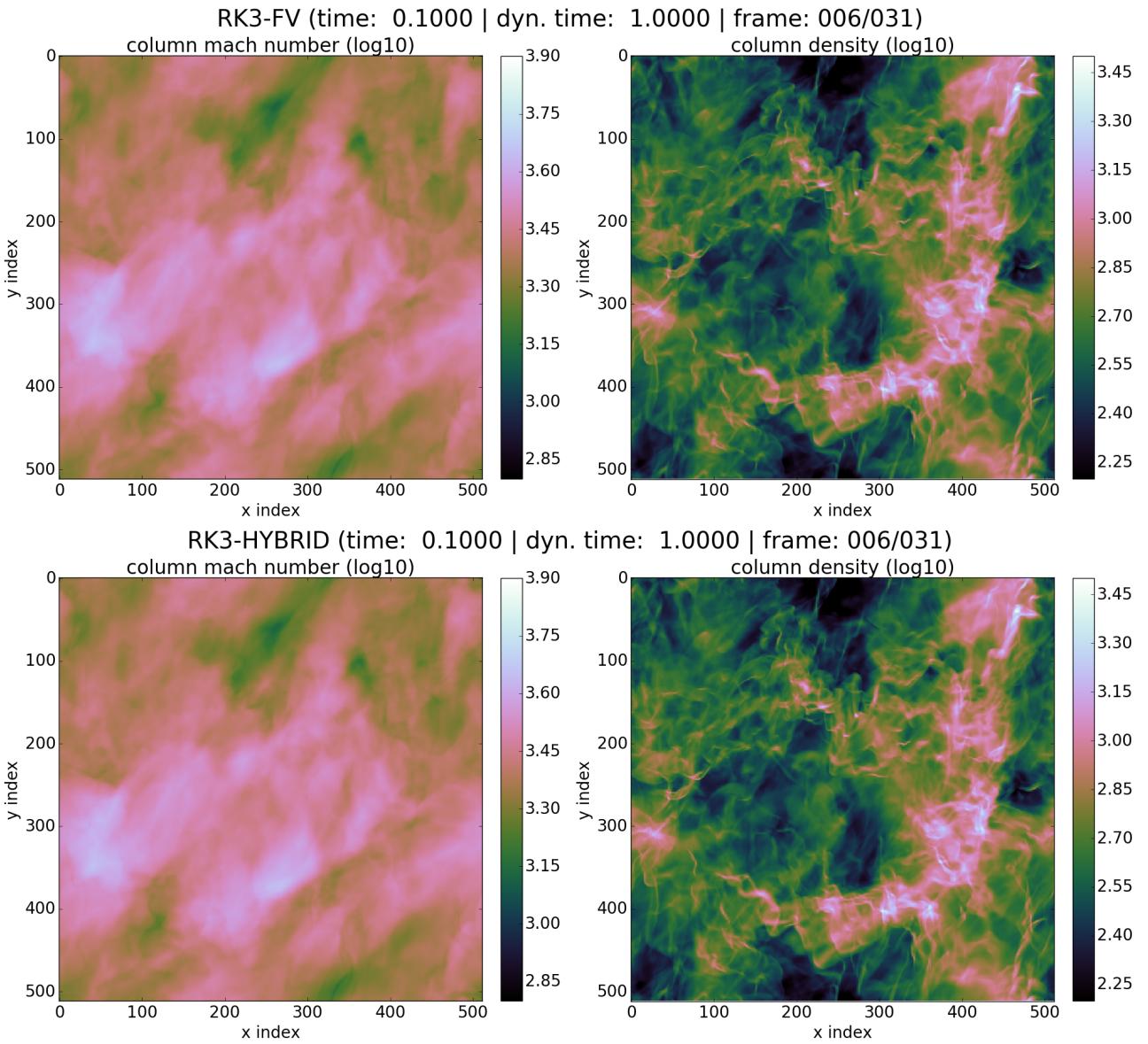


Figure 43: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 1.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 4.2$

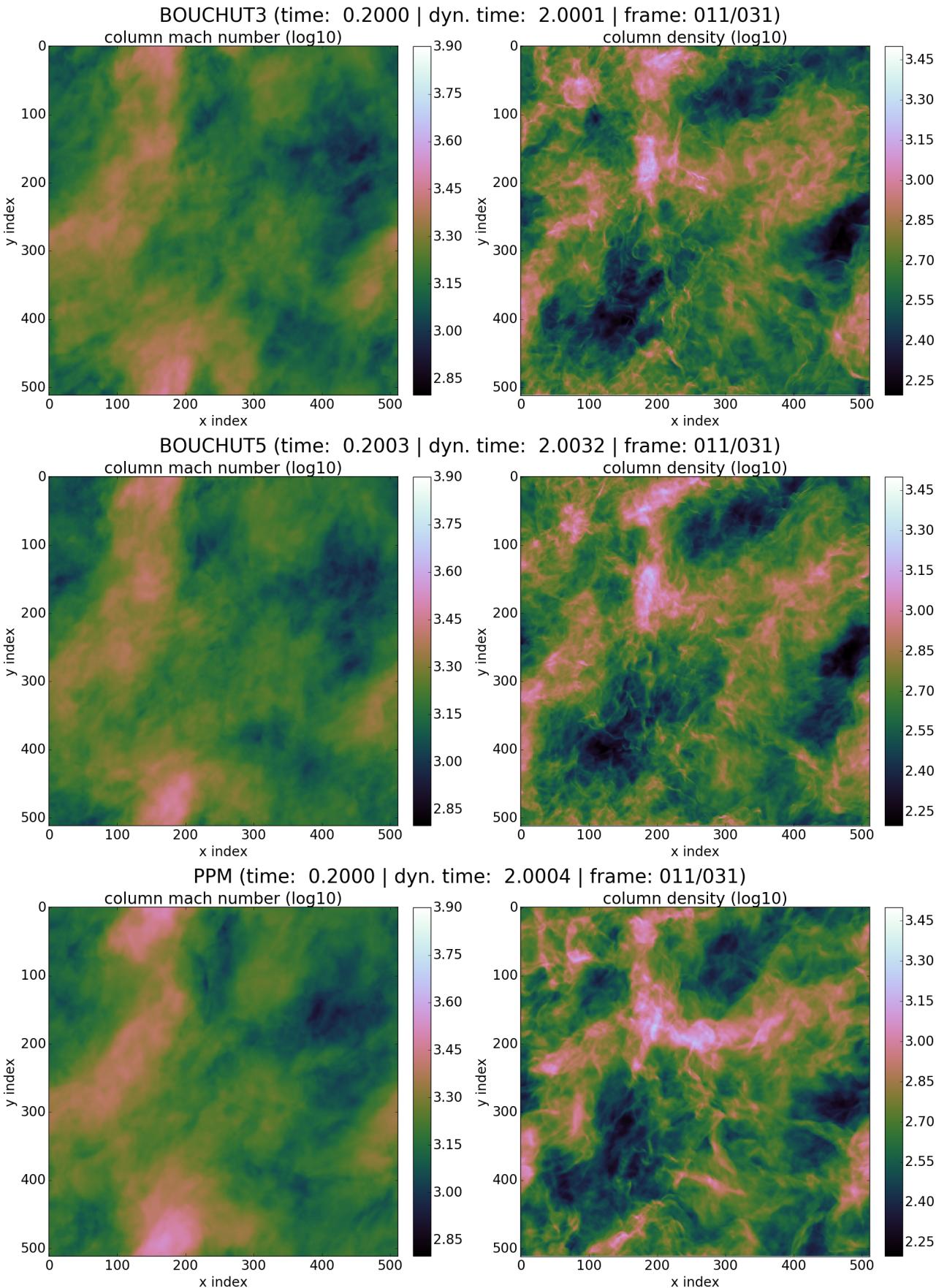


Figure 44: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 2.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 2.4$

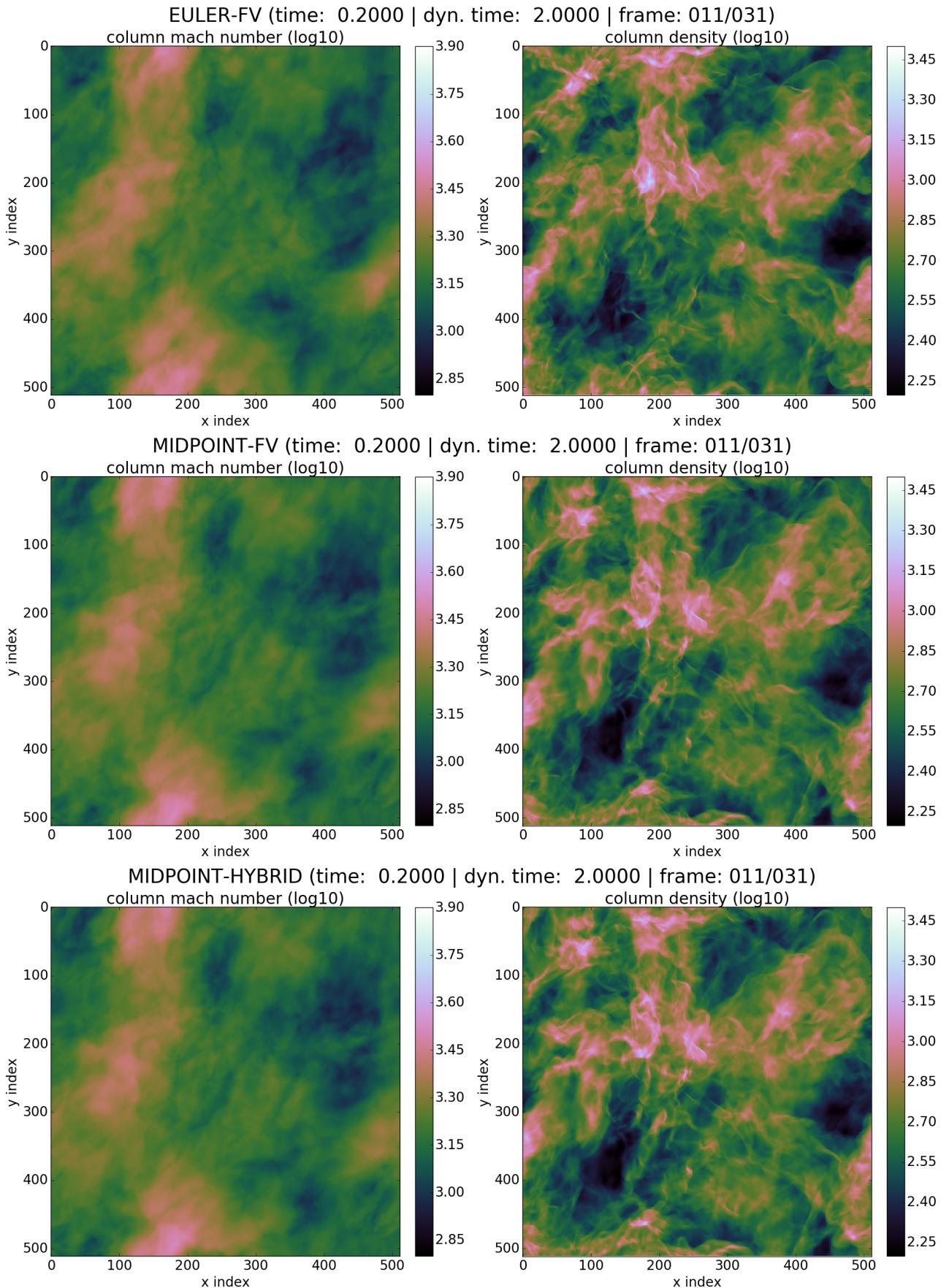


Figure 45: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 2.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 2.4$

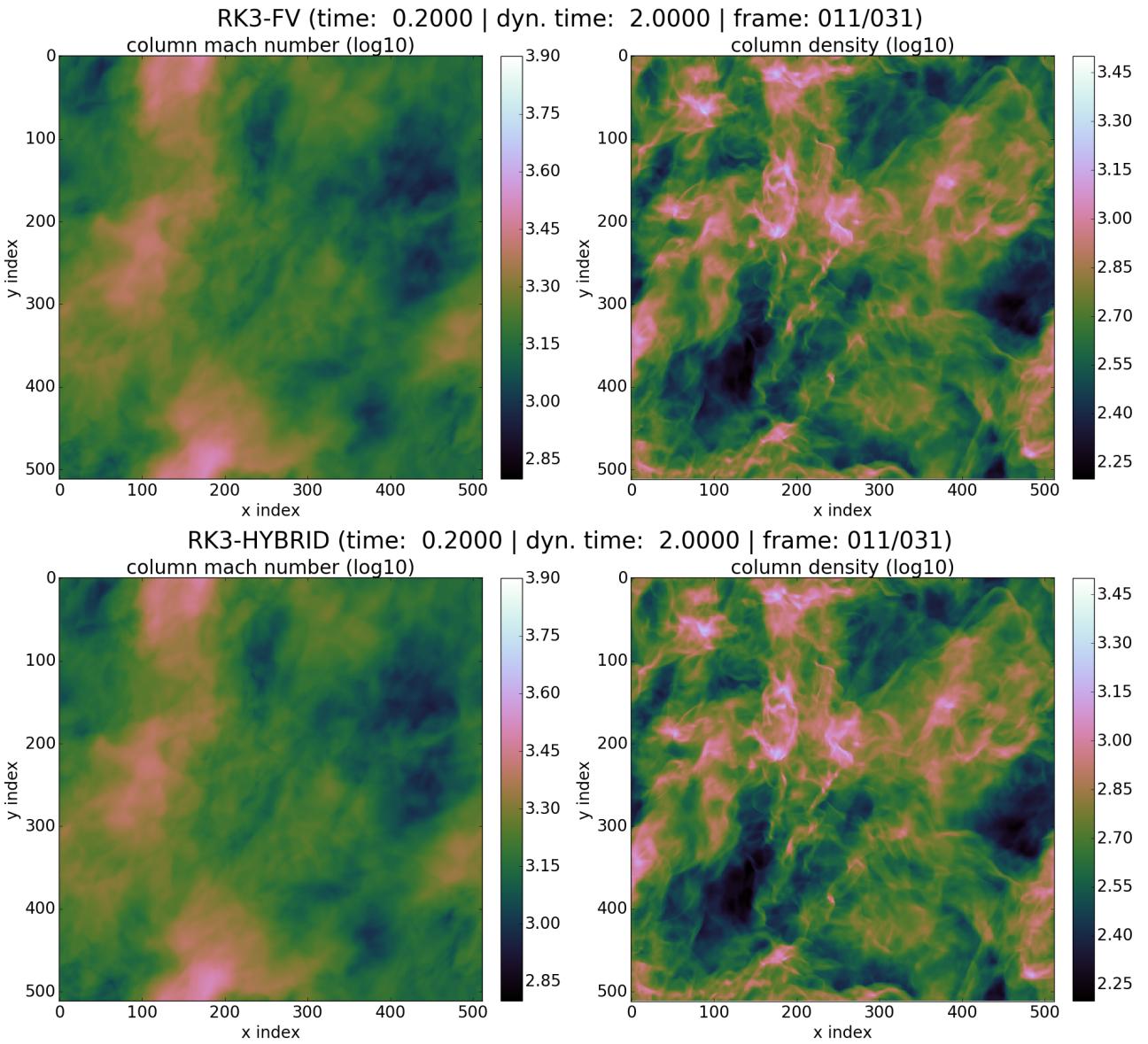


Figure 46: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 2.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 2.4$

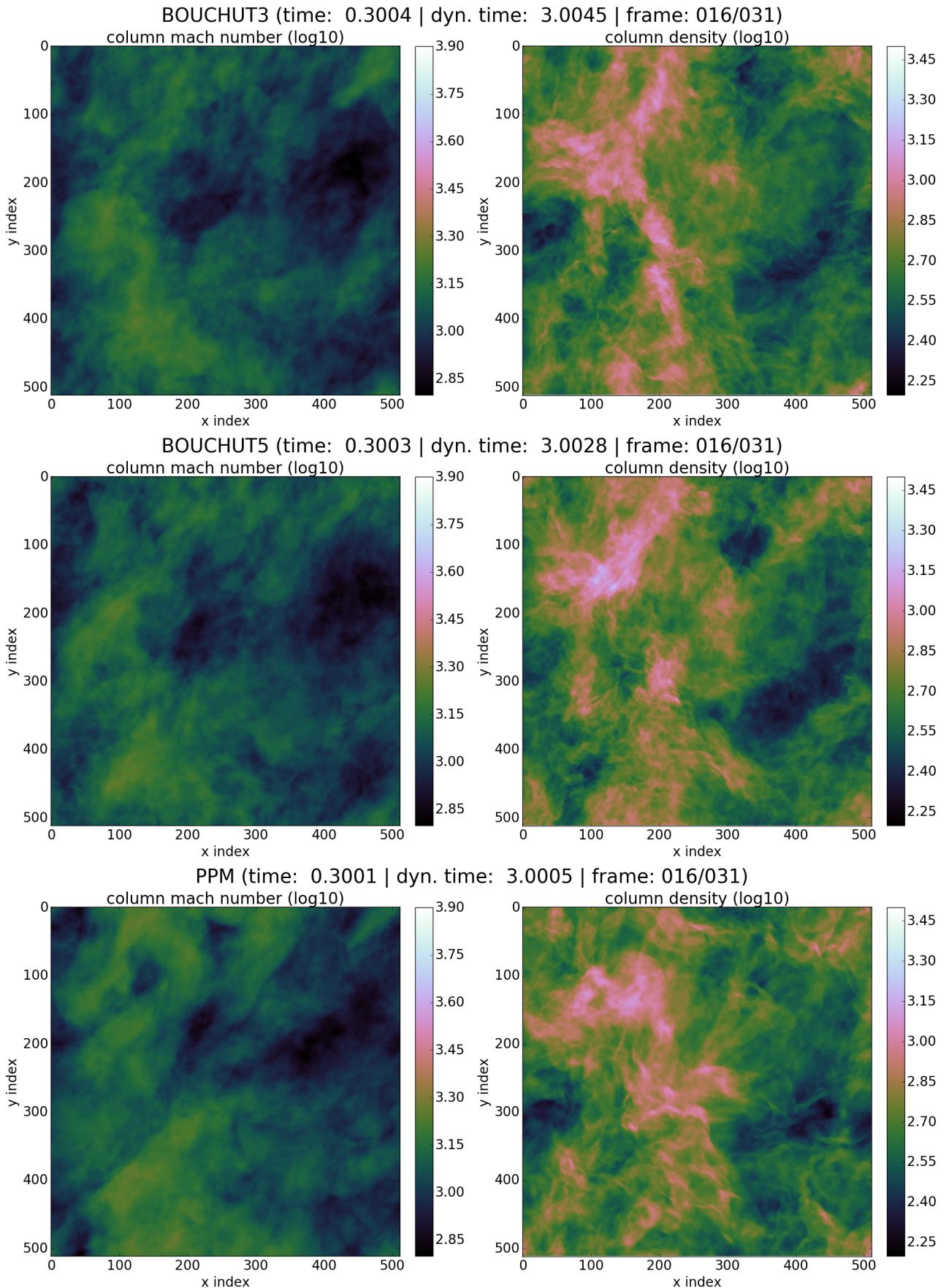


Figure 47: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 3.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 1.9$

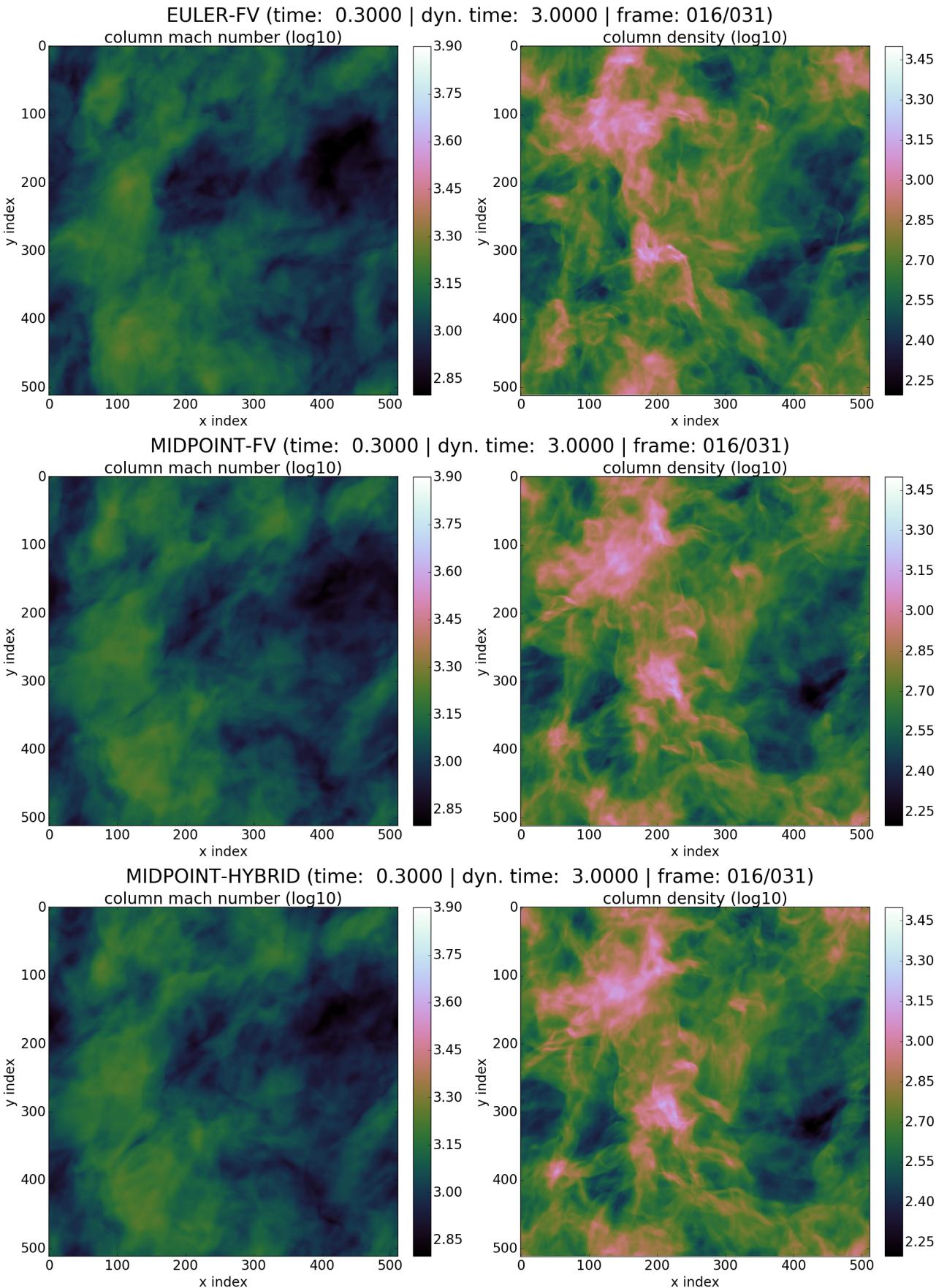


Figure 48: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 3.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 1.9$

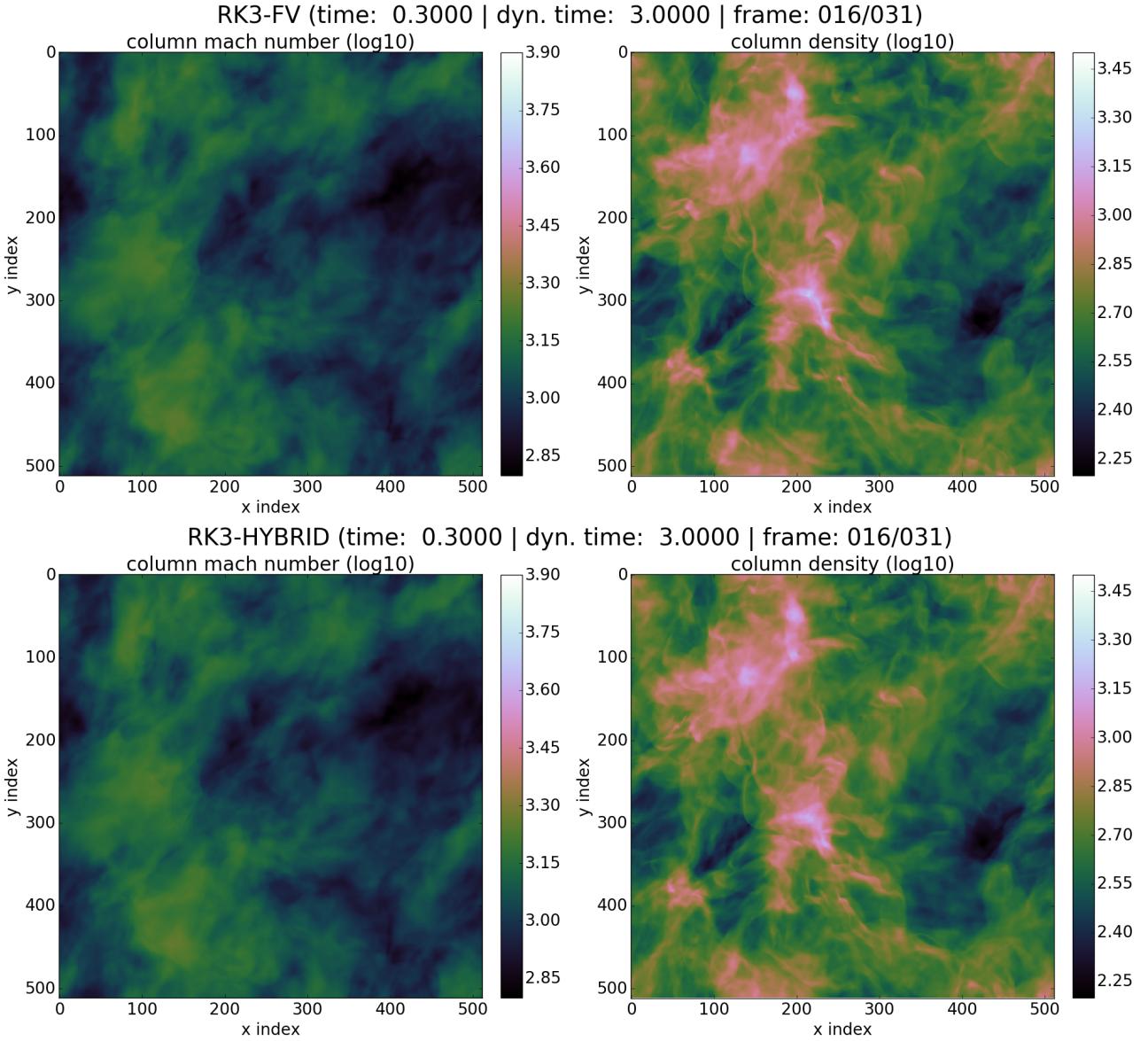


Figure 49: Decaying Turbulence from Mach 10: Column Sonic Mach Number and column density along z-axis at $t_d = 3.0$ when the root-mean-square mach number has fallen to $\mathcal{M} = 1.9$

4.3.3 FV-DG Mode Switching

The initially smoothed turbulence immediately shapes strong shocks everywhere, hence the sudden jump to over 95% FV elements in fig. 50. After that the situation alleviates a bit and more DG elements can be reintroduced. For not too long the second culmination of shocks appears and the amount of FV elements rises again. From $t_d = 1$ onwards the ratio declines in the same fashion as in fig. 27.

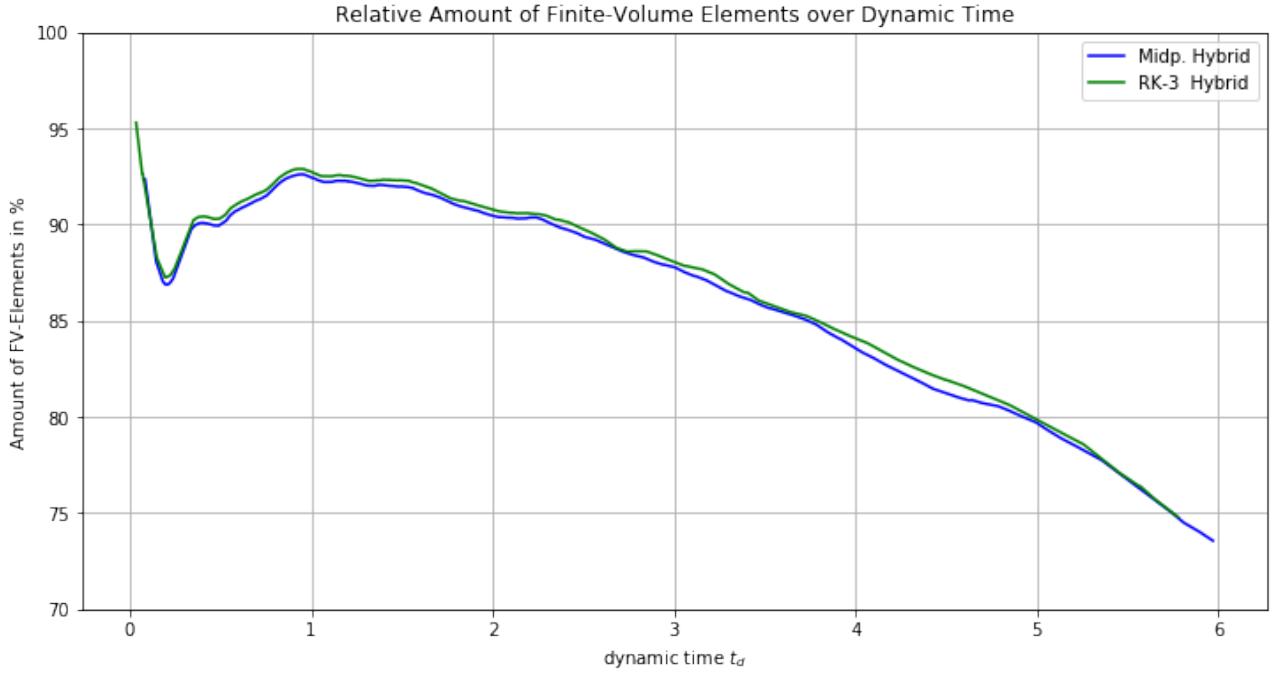


Figure 50: Time evolution of the fraction of Finite-Volume Elements to the total number of elements. *Remark* The other solvers would stay at 100% since they operate solely with Finite-Volumes.

4.3.4 Energy Dissipation

The course of the total energy fig. 51 is plotted in logarithmic scale so that the kinetic energy can be visually distinguished from the total energy. Polytropic cooling keeps the system at constant temperature hence the internal energy stays steady over time. There is no obvious difference among the solvers.

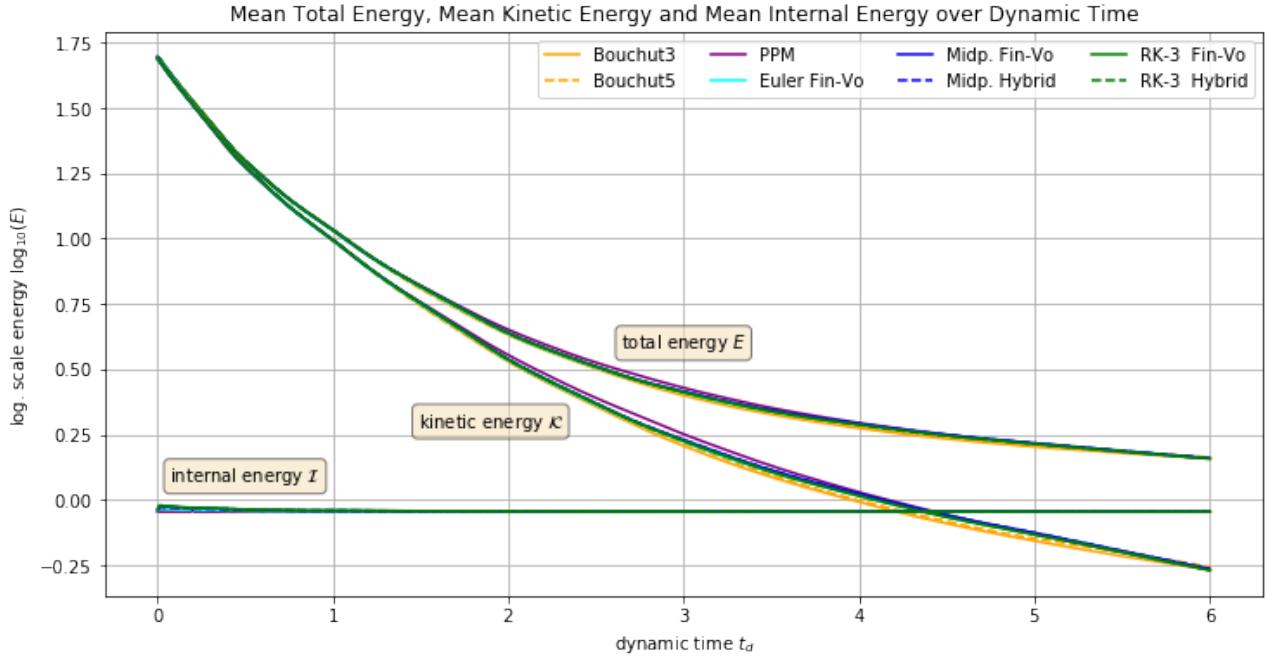


Figure 51: Time evolution of the energy. The logarithmic scaling helps to visually distinguish the course of total and kinetic energy. The total energy E is the sum of the kinetic \mathcal{K} and internal energy \mathcal{I} : $\log_{10}(E) = \log_{10}(\mathcal{I} + \mathcal{K})$.

The deprived internal energy is equivalent to the dissipated kinetic energy (fig. 52). See discussion under section ???. Right after simulation has started strong shocks emerge and the dissipation rate shoots up to its highest peak. By applying numerical integration under the averaged plot of the solvers $\int dt_d - \frac{d\mathcal{K}}{dt_d} = 48.251 \pm 0.007$ the initial kinetic energy is recovered $\mathcal{K}_0 = 48.75 \pm 0.03$ besides a tiny fraction of $\Delta\mathcal{K} = 0.5 \pm 0.03$ still left in the system. This little calculation reveals two insights. First, there are no noticeable energy sources or sinks besides polytropic cooling. Thus, the energy accounting is balanced. Second, the variation in dissipated energy among solvers is negligible. Conclusively, there is no difference in dissipation with this setup which contradicts the results of the driven turbulence.

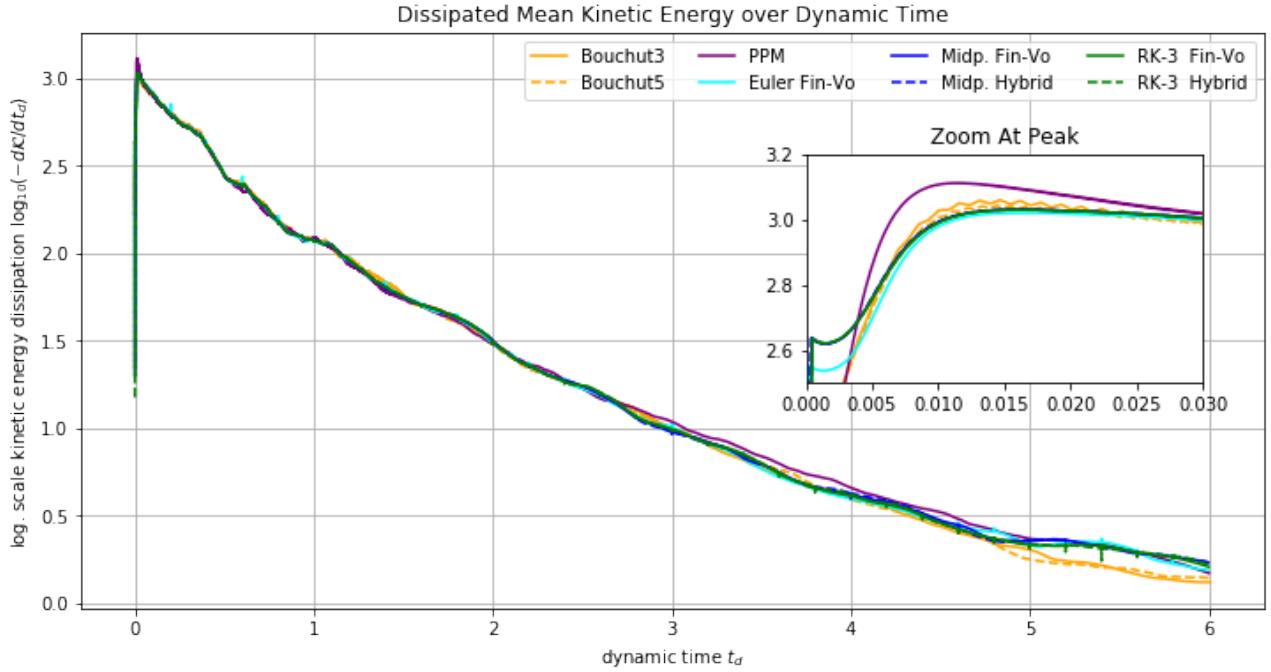


Figure 52: Time evolution of the kinetic energy dissipation in logarithmic scale $\log_{10}(-\frac{dK}{dt_d})$. The averaged area (numerical integration) under the plots amounts to $\int dt_d - \frac{dK}{dt_d} = 48.251 \pm 0.007$ which is almost all of the the initial kinetic energy: $K_0 = 48.75 \pm 0.03$.

4.3.5 Density & Velocity Distributions

In agreement with fig. 31 FLEXI's solvers retain more mass on the outer scales than FLASH's. Fitting was done with eqn. (52) in the same manner as in the driven turbulence (cf. section 4.2.5). From the width of the log-normal fits we try to reconstruct the sonic Mach number at that time. The result is shown in fig. 54.

The volume-weighted velocity PDFs in fig. 55 do not show anything surprising besides the fact that there is no significant bulk motion (cf. section ??).

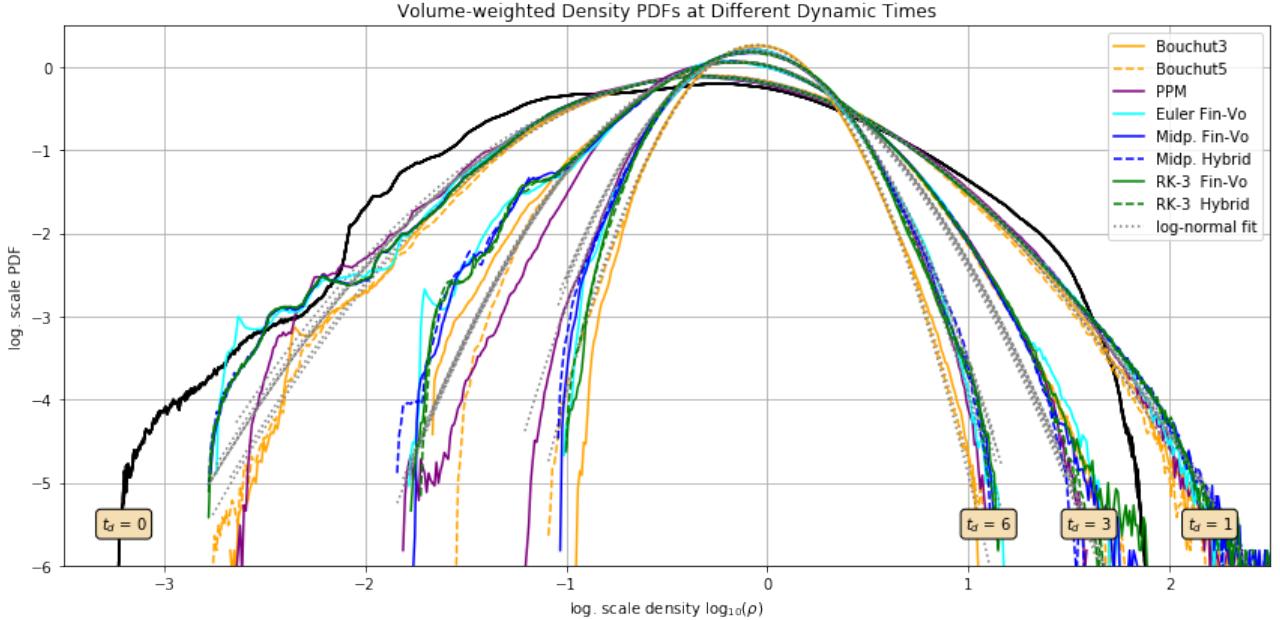


Figure 53: Log-log scale volume-weighted density PDFs. As the decay progresses the widths of the distributions decrease and the mean values return to the initial density: $\log_{10}(\rho_0) = \log_{10}(1) = 0$. The solid black curve depicts the density distribution of the smoothed initial state (cf. fig. 13). The applied Gaussian blur truncated the high velocity range. The log-normal fit (dotted lines) were done with eqn. (52).

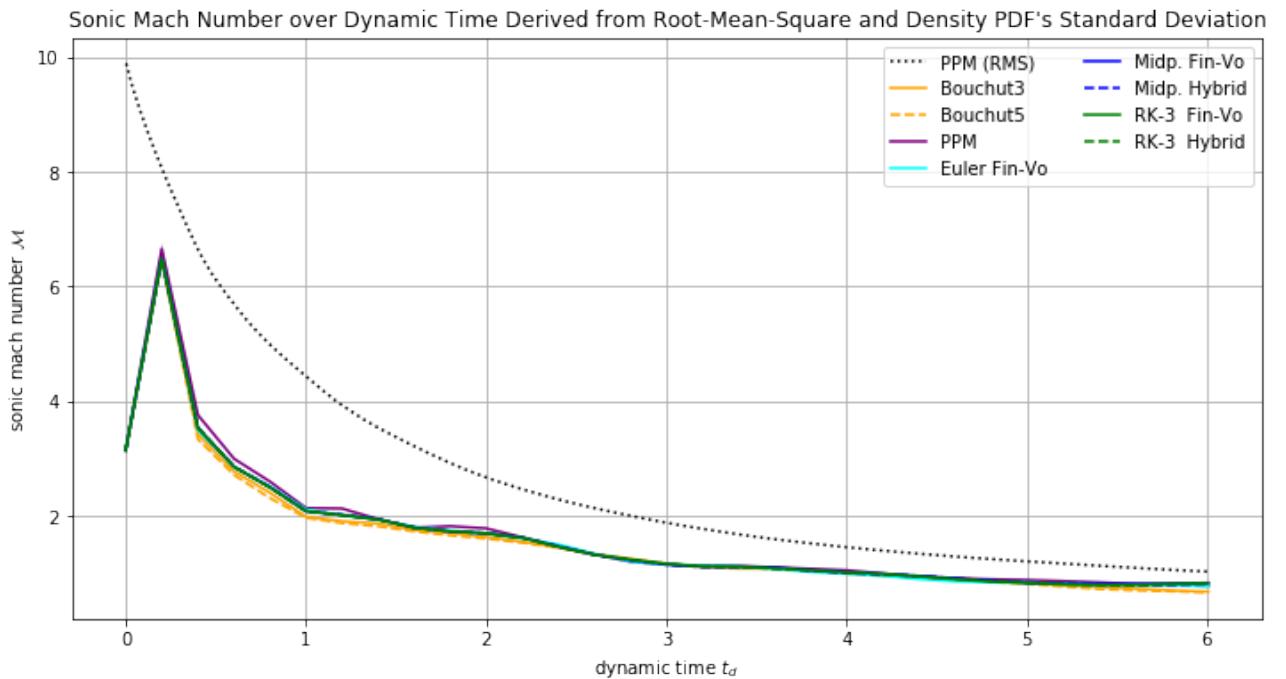


Figure 54: Time evolution of the sonic mach number over time derived from the width of the volume-weighted density PDFs of the decaying turbulence. The dashed line represents the root-mean-square velocity of the PPM solver and serves as a reference. Due to the lognormal relationship the fit yielded very low error margins. Hence, the error bars are to small to see. The spike on the left is caused by the originally smoothed initial state. The turbulence has to redevelop in order to yield sensible results. However, the Mach number estimation is systematically below the reference nevertheless.

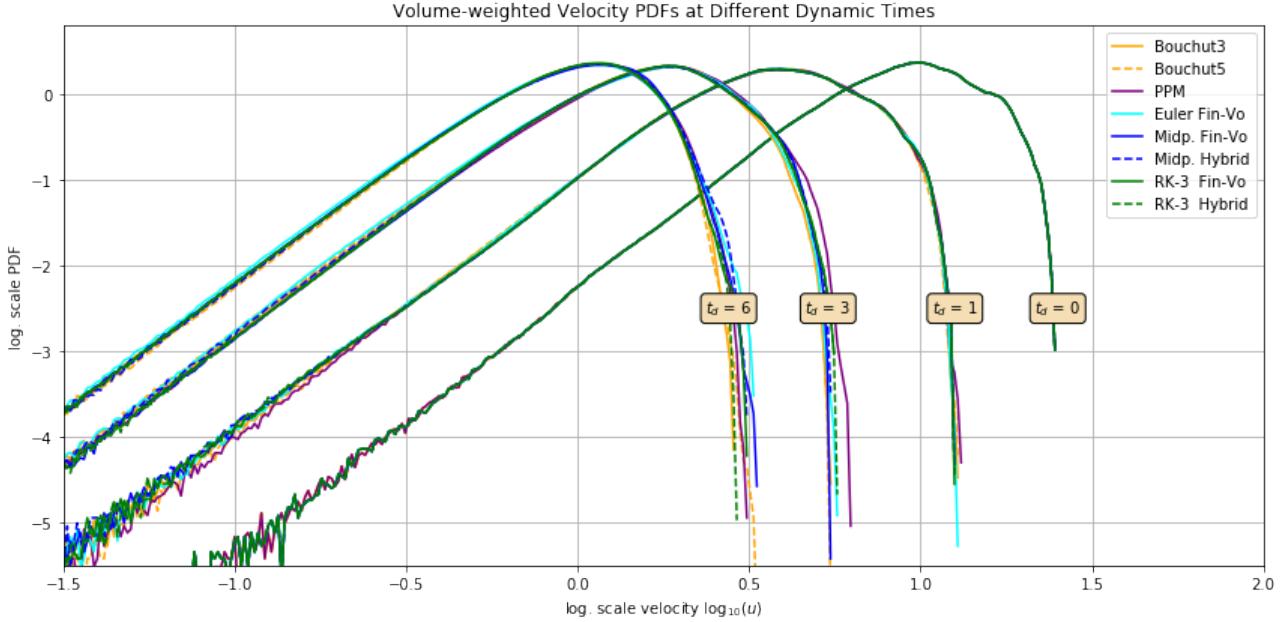


Figure 55: Log-log. scale volume-weighted velocity PDFs moving from right to left with increasing time due to kinetic energy decline of the decaying turbulence. The median point of the initial PDF at $t_d = 0$ is precisely over Mach 10: $\mu_0 = \log_{10}(1) = 10$. A non-existent dispersion of the PDFs over time indicates no bulk motion which is a good sign (cf. section 4.2.5).

4.3.6 Energy & Velocity Powerspectra

In analogy with section 4.2.6 we will take a look at the kinetic energy and the velocity powerspectra. In addition to the density PDF discussed in the previous section they allow us to gain an insight into the resolution of movement on small spatial scales. Since the kinetic energy declines over time the powerspectra in fig. 56 shift downwards. At the end of the simulation $t_d = 6$ PPM retained the most energy in the dissipation range. It is interesting to see that there still is an imprint left from the turbulent driver who produced the original turbulence in the first place. The buckling of the spectra within the large scale range is typical for driven turbulences. During the evaluation of the column density/velocity snapshots under section 4.3.2 we discussed the preservation of large scale structures over long periods of time. This observation resurfaces here again.

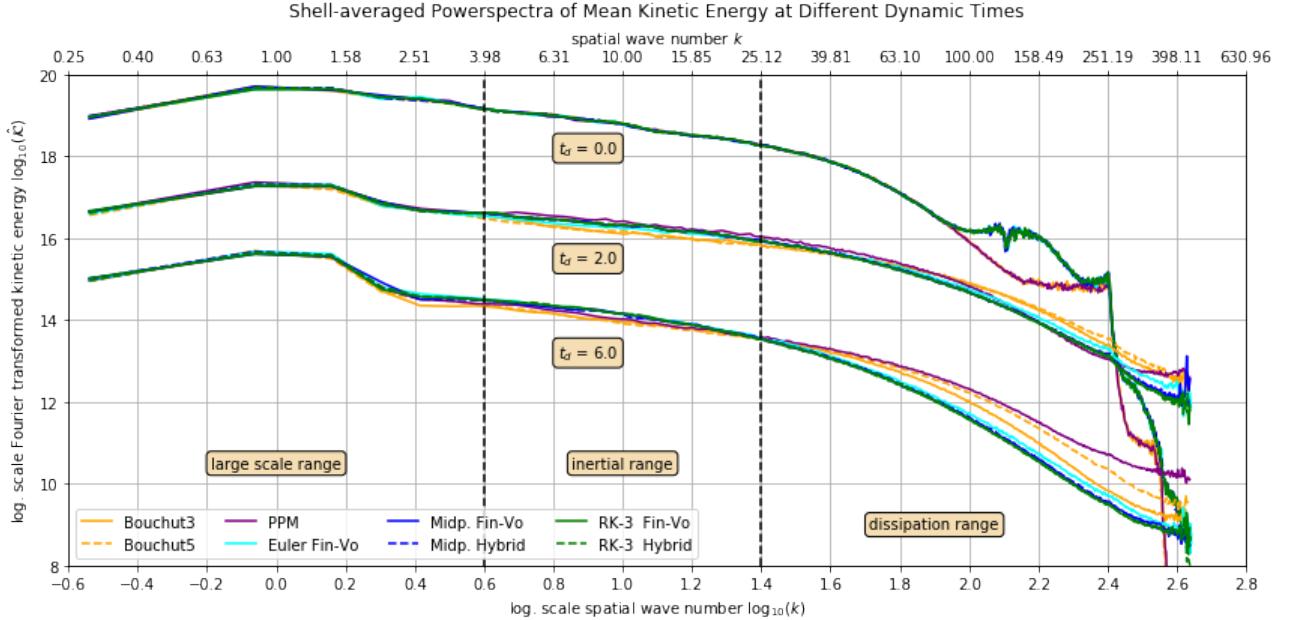


Figure 56: Powerspectra of the volume-weighted mean kinetic energy field shown for three stages of decay: $t_d = 0$, $t_d = 3.0$ and $t_d = 6.0$. Since the total kinetic energy declines over time the powerspectra shift downwards. The strange looking spectrum from $t_d = 0$ is the result of the applied Gaussian blur on the initial state.

Fig. 57 and fig. 58 show that all simulations maintain the energy cascade already present in the initial state. The related powerspectra $t_d = 0$ are a bit distorted from the Gaussian blur but the initial slope of the inertial range is very near the expectation marked by the solid black line on top of the plot. As described in detail in section 4.2.6 the slopes flatten with increasing time since the energy cascade slowly breaks down with progressing the decay.

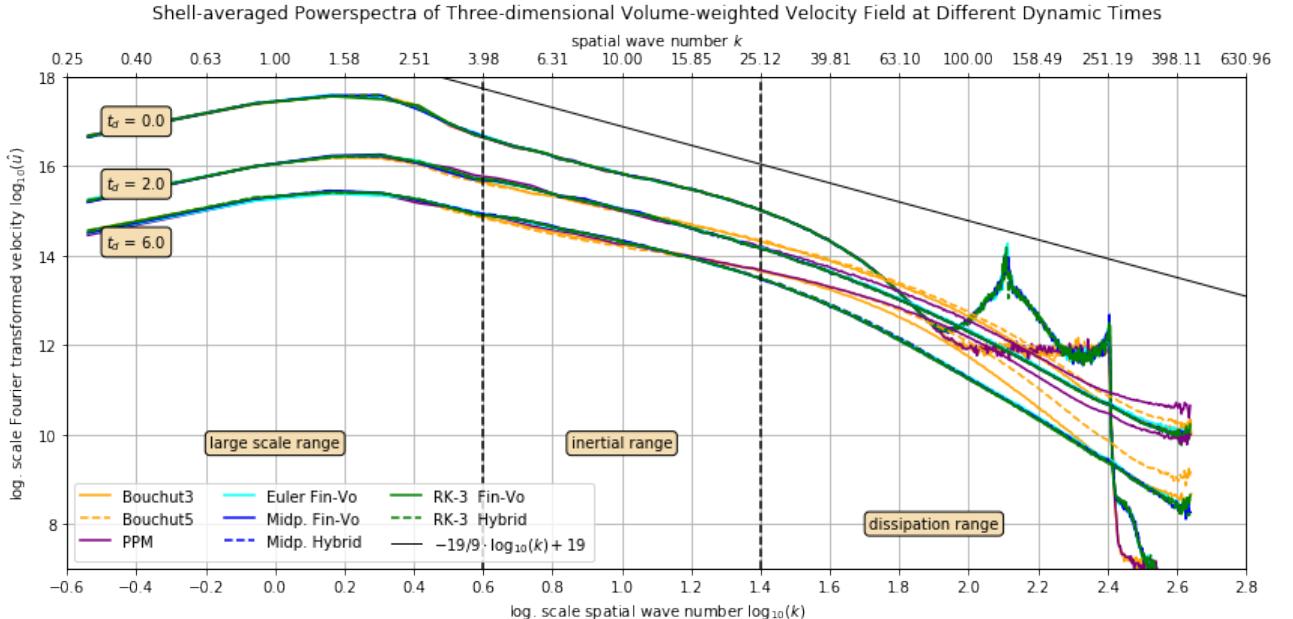


Figure 57: Powerspectra of the volume-weighted velocity field for three stages of decay: $t_d = 0$, $t_d = 3.0$ and $t_d = 6.0$. Since the total kinetic energy declines over time the powerspectra shift downwards. The strange looking spectrum from $t_d = 0$ is the result of the applied Gaussian blur on the initial state.

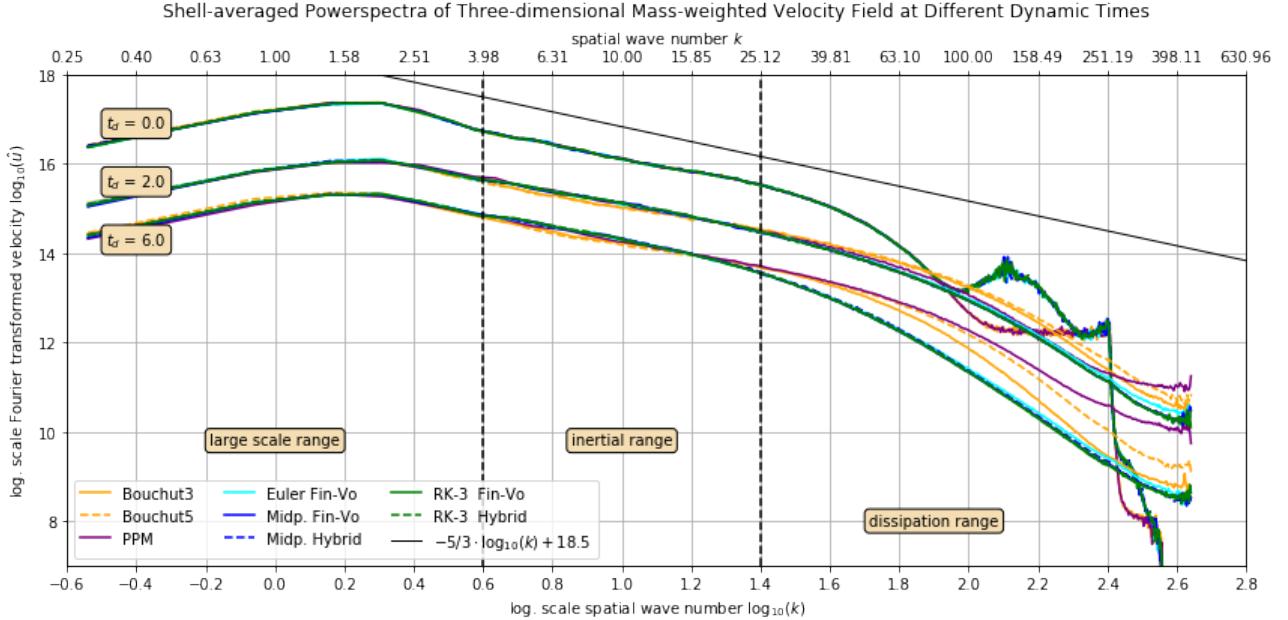


Figure 58: Powerspectra of the mass-weighted velocity field for three stages of decay: $t_d = 0$, $t_d = 3.0$ and $t_d = 6.0$. Since the total kinetic energy declines over time the powerspectra shift downwards. The strange looking spectrum from $t_d = 0$ is the result of the applied Gaussian blur on the initial state.

4.3.7 Summary

The decaying turbulence setup was conducted in analogy with the driven setup under section 4.2. A fully developed Mach 10 turbulence was let to decay by FLASH's and FLEXI's solvers. In a comparative study we evaluated columnn density/velocity snapshots at three different stages of decay and underpinned the observations with density distributions and powerspectra.

All schemes provided very similar solutions with minor differences on small scales. See fig. 41 to fig. 49. A variance in energy dissipation rates (cf. fig. 52) was not quantifiable. In accordance with the results of the decay phase of the driven turbulence (cf. section 4.2.5) the relation between density PDF and sonic Mach number (cf. section 2.2.2) systematically underestimates the correct root-mean-square Mach number (cf. fig. 54). Apparently, the relation does not hold for decaying turbulences. The accurate modelling of the energy cascade was confirmed by the velocity powerspectra in fig. 57 and fig. 58.

There is no clear winner who resolves small scale structures best. On the hand FLASH's solver contain more energy in the dissipative range of the kinetic energy powerspectra (cf. fig. 56) throughout the simulation. But on the other hand the solver from FLEXI show more amount of mass on both ends of the density spectrum (cf. fig. ??plots/decayturb/pdf-vw-dens.png). Though, one should not forget that all plots are in log-log scale which means that subtle differenes get blown up and pretend to be important.

The decaying turbulence setup was revealing in the sense that the influence of turbulent forcing on dissipation rates is immense. Subsequent driven and decay simulations with lower and higher Mach numbers come to the same conclusion.

5 Conclusion & Outlook

A multitude of reasons are conceivable for what systematically prevents FLEXI's schemes from pushing energy into small scales. First of all

The high-resolution finite methods discussed in these lectures can be used very successfully for a wide range of problems. They are not foolproof, however, and one should never accept computed results without a critical study of their accuracy. This is often hard to assess for complex problems where the exact solution is not known, but the following techniques can help:

- Investigate simple cases where exact solutions might be known, or at least the correct qualitative behavior of the solution is well understood.
- Reduce the number of space dimensions by considering radially symmetric solutions, for example. Then a fine-grid solution in one space dimension can be used as a reference solution for the multi-dimensional solution.
- Perform grid refinement studies on the real problem of interest. If you refine the grid does the solution remain basically the same? If not, then you probably cannot trust either solution. (If so, both solutions may still be completely incorrect. An error in the code that changes the equations might lead to a method that converges very nicely to a solution of the wrong equation.) A number of specific difficulties that can arise in solving the Euler equations have already been mentioned, such as
- The use of a nonconservative method can give shocks that look reasonable but which travel at the wrong speed (Sect. 4.3).
- Stiff source terms can lead to similar results (Sect. 5.4).
- The computed solution may not satisfy the entropy condition, leading to discontinuities where there should be a smooth rarefaction (Sect. 4.6.4).

This same sort of artifact is also often seen when two shocks collide or when a shock reflects off a solid wall. Consider two identical shocks approaching each other with zero velocity in between (which also models reflection at a wall halfway between the shocks — see Sect. 4.9.3). Each shock may have settled down to some numerical traveling wave that does not appear to generate any noise. But when the shocks collide, the result is two new out-going shocks with a different state in between than before the collision, with higher density and pressure but still zero velocity. During the interaction phase considerable noise will be generated in the other families, and in particular a spurious entropy wave will be generated which is then stationary in the zero-velocity region between the out-going shocks. This wave yields a dip in the density. The pressure, however, is nearly constant and so this dip in density results in an increase in the temperature $T = p/Tzg$. The gas appears to have been heated at the point where the collision occurs. In particular, in computing the reflection of a shock off a solid wall, this spurious temperature rise occurs at the wall itself. This phenomenon is frequently observed in numerical simulations where shocks reflect off physical boundaries, and is known as wall heating in the literature. See, for example, [73], [175].

7.7 Grid-Aligned Shocks In a multi-dimensional calculation it may seem advantageous to have a shock aligned with the grid so that Riemann problems normal to the cell edges are also in the physically correct direction. However, a shock that is nearly aligned with the grid can also suffer certain numerical instabilities that have no analog in one-dimensional calculations. Figure 7.3 shows density contours at a sequence of times for a colliding flow problem. Initially $g = p = 1$, $v = 0$ everywhere, while the a_i -velocity is $u = +20$ on the left half of the domain and $u = -20$ on the right. This colliding flow should give rise to two symmetric shock waves propagating outwards. If the initial data is exactly uniform then the calculation will yield a reasonable approximation to this. For the calculation in Fig. 7.3, however, the density was initially perturbed