

Algorithmic comparisons of decaying, isothermal, supersonic turbulence

S. Kitsionas^{1,*}, C. Federrath^{2,3}, R. S. Klessen^{1,2}, W. Schmidt⁴, D. J. Price^{5,6}, L. J. Dursi⁷, M. Gritschneider⁸, S. Walch^{8,9}, R. Piontek¹, J. Kim¹⁰, A.-K. Jappsen^{1,7,9}, P. Cielieglag^{8,11}, and M.-M. Mac Low¹²

¹ Astrophysikalisches Institut Potsdam, An der Sternwarte 16, 14482 Potsdam, Germany
e-mail: skitsionas@googlemail.com

² Institut für Theoretische Astrophysik, Universität Heidelberg, Albert-Ueberle-Str. 2, 69120 Heidelberg, Germany

³ Max-Planck-Institut für Astronomie, Königstuhl 17, 69117 Heidelberg, Germany

⁴ Lehrstuhl für Astronomie, Institut für Theoretische Physik und Astrophysik, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany

⁵ School of Physics, University of Exeter, Stocker Road, EX4 4QL Exeter, UK

⁶ CSPA, School of Mathematical Sciences, Monash University, Clayton Vic 3168, Australia

⁷ Canadian Institute for Theoretical Astrophysics, University of Toronto, 60 St. George Street, Toronto, ON, M5S 3H8, Canada

⁸ Universitäts-Sternwarte München, Scheinerstr. 1, 81679 München, Germany

⁹ School of Physics & Astronomy, Cardiff University, 5 The Parade, CF24 3AA Cardiff, UK

¹⁰ Korea Astronomy and Space Science Institute, 61-1 Hwaam-dong, Yuseong-gu, Daejeon, 305-348, Republic of Korea

¹¹ N. Copernicus Astronomical Center, Bartycka 18, 00-716 Warsaw, Poland

¹² American Museum of Natural History, Department of Astrophysics, Central Park West at 79th Street, New York, NY, 10024-5192, USA

Received 16 October 2008 / Accepted 24 September 2009

ABSTRACT

Context. Simulations of astrophysical turbulence have reached such a level of sophistication that quantitative results are now starting to emerge. However, contradicting results have been reported in the literature with respect to the performance of the numerical techniques employed for its study and their relevance to the physical systems modelled.

Aims. We aim at characterising the performance of a variety of hydrodynamics codes including different particle-based and grid-based techniques on the modelling of decaying supersonic turbulence. This is the first such large-scale comparison ever conducted.

Methods. We modelled driven, compressible, supersonic, isothermal turbulence with an rms Mach number of $M_{\text{rms}} \sim 4$, and then let it decay in the absence of gravity, using runs performed with four different grid codes (ENZO, FLASH, TVD, ZEUS) and three different SPH codes (GADGET, PHANTOM, VINE). We additionally analysed two calculations denoted as PHANTOM A and PHANTOM B using two different implementations of artificial viscosity in PHANTOM. We analysed the results of our numerical experiments using volume-averaged quantities like the rms Mach number, volume- and density-weighted velocity Fourier spectrum functions, and probability distribution functions of density, velocity, and velocity derivatives.

Results. Our analysis indicates that grid codes tend to be less dissipative than SPH codes, though details of the techniques used can make large differences in both cases. For example, the Morris & Monaghan viscosity implementation for SPH results in less dissipation (PHANTOM B and VINE versus GADGET and PHANTOM A). For grid codes, using a smaller diffusion parameter leads to less dissipation, but results in a larger bottleneck effect (our ENZO versus FLASH runs). As a general result, we find that by using a similar number of resolution elements N for each spatial direction means that all codes (both grid-based and particle-based) show encouraging similarity of all statistical quantities for isotropic supersonic turbulence on spatial scales $k \lesssim N/32$ (all scales resolved by more than 32 grid cells), while scales smaller than that are significantly affected by the specific implementation of the algorithm for solving the equations of hydrodynamics. At comparable numerical resolution ($N_{\text{particles}} \approx N_{\text{cells}}$), the SPH runs were on average about ten times more computationally intensive than the grid runs, although with variations of up to a factor of ten between the different SPH runs and between the different grid runs.

Conclusions. At the resolutions employed here, the ability to model supersonic to transonic flows is comparable across the various codes used in this study.

Key words. hydrodynamics – shock waves – methods: numerical – turbulence

1. Introduction

Laboratory and terrestrial fluid dynamics are often described as incompressible flow (e.g., Lesieur 1997); however, astrophysical fluids are usually characterised by highly compressible supersonic turbulent motions (see e.g. Elmegreen & Scalo 2004; Scalo & Elmegreen 2004). For example, the large observed line widths

in Galactic and extragalactic molecular clouds and star-forming regions show direct evidence of chaotic velocity fields with magnitudes in excess of the sound speed. This random motion carries enough kinetic energy to counterbalance and sometimes overcompensate for the effects of self-gravity in these clouds (e.g. Ballesteros-Paredes et al. 2007; Blitz et al. 2007). The intricate interplay between supersonic turbulence and self-gravity determines the overall dynamical evolution of these clouds and their observable features, such as their density structure, the star

* *Current address:* Hellenic-American Educational Foundation, Psychiko College, Stefanou Delta 15, GR-15452 P. Psychiko, Greece.

formation rate within them, and their lifetimes (see e.g. [Mac Low & Klessen 2004](#); [McKee & Ostriker 2007](#)).

Despite turbulence being a universal phenomenon, it is also one of the least understood natural phenomena. Turbulence arises as a result of the nonlinear terms in the Navier-Stokes equation that governs the dynamical behaviour of gases and fluids ([Frisch 1995](#); [Lesieur 1997](#)). A self-consistent mathematical formulation does not exist. Thus, analytical research mostly focuses on finding appropriate closure equations that capture the bulk behaviour of the system.

As a first approach, turbulence is characterised by two spatial scales that are connected by a self-similar cascade of kinetic energy that occurs over the so-called inertial range. Energy is injected into the system on some large scale L and dissipated on small scales ℓ that are comparable to the viscous length ℓ_{visc} . For incompressible turbulence, [Kolmogorov \(1941\)](#) described a simple heuristic model based on dimensional analysis that captures the basic behaviour of the flow surprisingly well. He assumed that turbulence driven on a large scale L forms eddies on that scale. These eddies interact to form slightly smaller eddies, transferring some of their energy to the smaller scale. The smaller eddies in turn form even smaller ones, and so on, until energy has cascaded all the way down to the dissipation scale ℓ_{visc} .

In order to maintain a steady state, equal amounts of energy must be transferred from each scale in the cascade to the next, and eventually dissipated, at a rate $\dot{E} = \eta v_L^3/L$, where v_L is the typical velocity on scale L and η is a constant determined empirically. [Kolmogorov \(1941\)](#) assumes this rate is constant throughout the scales, leading to $v_L \propto L^{1/3}$, or equivalently $v_k \propto k^{-1/3}$ for wavenumbers $k \propto 1/L$. The kinetic energy in the wavenumber interval $[k, k + dk]$ is $E_{\text{kin}}(k) \propto v_L^2 \propto L^{2/3} \propto k^{-2/3}$ and consequently the energy spectrum function $E(k) = dE_{\text{kin}}/dk \propto k^{-5/3}$. This describes the self-similar cascade of turbulent kinetic energy. Most of this energy resides at the top of this cascade near the driving scale, and the spectrum drops off steeply below ℓ_{visc} . Because of the apparently local nature of the cascade in wavenumber space, the viscosity only determines the behaviour of the energy distribution at the bottom of the cascade below ℓ_{visc} , while the driving only determines the behaviour near the top of the cascade on and above L .

Supersonic flows in highly compressible gas create strong density perturbations. Early attempts to understand turbulence in the interstellar medium ([von Weizsäcker 1943, 1951](#); [Chandrasekhar 1949](#)) were based on insights drawn from incompressible turbulence. An attempt to analytically derive the density spectrum and resulting gravitational collapse criterion was first made by [Chandrasekhar \(1951a,b\)](#). This work was followed up by several authors, culminating in the work by [Sasao \(1973\)](#) on density fluctuations in self-gravitating media. [Larson \(1981\)](#) qualitatively applied the basic idea of density fluctuations driven by supersonic turbulence to the problem of star formation. [Bonazzola et al. \(1992\)](#) used a re-normalization group technique to examine how the slope of the turbulent velocity spectrum could influence gravitational collapse. This approach was combined with low-resolution numerical models to derive an effective adiabatic index for subsonic compressible turbulence by [Panis & Pérault \(1998\)](#).

In supersonic turbulence, shock waves offer additional possibilities for dissipation. They can transfer energy between widely separated scales, removing the local nature of the turbulent cascade typical of incompressible turbulence. The spectrum may shift only slightly, however, as the power spectrum (Fourier spectrum) of a step function representative of a perfect shock

wave is k^{-2} . [Boldyrev \(2002\)](#) has proposed a theory of velocity structure function scaling based on the work of [She & Leveque \(1994\)](#) using the assumption that dissipation in supersonic turbulence primarily occurs in sheet-like shocks, rather than linear filaments at the centres of vortex tubes (see also [Boldyrev et al. 2002a,b](#)). Transport properties of supersonic turbulent flows in the astrophysical context have been discussed by [de Avillez & Mac Low \(2002\)](#) and [Klessen & Lin \(2003\)](#), and the fractal dimension of turbulent media by [Federrath et al. \(2009b\)](#).

As satisfying analytic models are rare, especially when dealing with compressible and supersonic turbulent flows, special attention is drawn to numerical approaches. A wide range of methods are used to study turbulence, ranging from simulating statistical processes such as random walks (e.g. [Metzler & Klafter 2000](#)), remapping models, or certain Hamiltonian systems ([Isichenko 1992](#)), to hydrodynamic large-eddy simulations (LES). In LES only the largest spatial scales are resolved directly using a hydrodynamic integrator. For the turbulent dynamics on smaller scales, a so-called subgrid scale (SGS) model is utilised. Among astrophysicists, the most often used SGS model is numerical dissipation, i.e. performing Implicit LES. It is not possible to optimise the use of closure models for astrophysical turbulence through comparisons with laboratory experiments. Therefore, the representation of the SGS behaviour provided by numerical dissipation must be sufficient, and indeed provides a reasonably good approximation ([Benzi et al. 2008](#)).

In the current study we focus on comparing different Implicit LES schemes. Our goal is to assess the applicability of different numerical schemes to the modelling of supersonic turbulent flows, and to compare their validity and accuracy in the astrophysical context. To keep this comparison simple, we focus on purely hydrodynamic turbulence in isothermal gaseous media in regions with periodic boundaries, and study the decay of fully developed turbulence. We follow the dissipation of kinetic energy due to the numerical viscosity intrinsic to any numerical scheme, and characterise the turbulent velocity field using volume- and density-weighted velocity power spectra¹, and probability distribution functions of density, velocity and velocity derivatives. We remind the reader that in supersonic turbulence energy is not only dissipated below $\ell_{\text{visc}}^{\text{art}}$ by the action of artificial viscosity on the smallest scale eddies, but also in shocks. In most of the codes employed here artificial viscosity is necessary also for the modelling of shocks. The main aim of our comparisons is to characterise the role of artificial viscosity in dissipating energy below $\ell_{\text{visc}}^{\text{art}}$, rather than the use of artificial viscosity in the modelling of shocks. A discussion on the shock capturing ability of the codes used in this study is provided in Sect. 2 (for a comprehensive such comparison, see also [Tasker et al. 2008](#)). One of the fundamental questions we want to address is at which numerical resolution are different numerical schemes capable of modelling supersonic turbulence adequately.

This is the first such comparative study; there has been no coherent comparison of the various hydrodynamic codes used in the literature for the study of supersonic turbulence². In spite of the fact that results from different codes appear to contradict each other and lead to different interpretations of the role of turbulence in astrophysics – e.g. the hydrodynamic simulations

¹ Volume-weighted velocity power spectra are often referred to as kinetic energy spectra for incompressible turbulence (e.g., [Frisch 1995](#)). However, for compressible turbulence the kinetic energy is proportional to the density-weighted velocity power spectrum.

² A limited comparison of two codes was presented for self-gravitating turbulence in [Klessen et al. \(2000\)](#) and [Heitsch et al. \(2001\)](#).

of Ballesteros-Paredes et al. (2006) and Padoan et al. (2007) in which different power-law slopes are obtained from the velocity power spectra leading to different interpretations of the role of turbulence on cloud fragmentation and the resulting core mass function – it has not been properly checked whether at least some of these differences are due to differences in the numerical schemes employed. We perform and analyse here a first set of low-resolution calculations aiming at extending our investigations in the future to higher resolution simulations, achieved either directly, or by using adaptive resolution techniques like Adaptive Mesh Refinement (Kritsuk et al. 2006; Schmidt et al. 2009), or Particle Splitting (Kitsionas & Whitworth 2002), or by using Subgrid Scale Models (Schmidt et al. 2006b,c), and/or combinations of the above. It should be emphasised that the typical number of resolution elements used here for the SPH calculations is quite large (number of particles $N = 215^3$) compared to the typical number of particles used in existing studies of supersonic turbulence and cloud fragmentation in the literature (e.g. Ballesteros-Paredes et al. 2006). On the other hand, the number of resolution elements used for the grid codes presented here is rather small (number of grid cells $N = 256^3$) compared to what is the current state-of-the-art resolution for such studies (e.g. Kritsuk et al. 2007; Padoan et al. 2007; Federrath et al. 2009a; Lemaster & Stone 2009; Schmidt et al. 2009).

The structure of our current study is as follows: in Sect. 2, we describe the setup of the experiments conducted as well as list the most important features of the codes used. In Sect. 3, we review the statistical measures used in this paper for the analysis of supersonic turbulence. In Sect. 4, we present the initial conditions employed for the decay simulations, while in Sect. 5 we discuss the results of the decay experiments and the comparison of the performance of the various codes. We summarise the computational efficiency of the codes and runs in Sect. 6, and present our conclusions in Sect. 7.

2. Experimental setup and numerical codes

Our aim is to study the decay of supersonic hydrodynamic turbulence using different grid- and particle-based codes³ and compare the performance of the codes in this experiment. Therefore, we need a turbulent gas distribution that will serve as an initial condition for all codes. For simplicity, the turbulent initial conditions are produced with one of the particle/SPH codes, and then the particle code data is interpolated onto a grid. This grid, in turn, provides the initial conditions for the grid-based codes.

The turbulent gas distribution is created with GADGET (Springel et al. 2001). We start with a box of side $L = 0.29$ pc. Inside this box, $215^3 = 9938375$ particles were distributed homogeneously representing a static, uniform, isothermal gas with temperature $T = 11.4$ K (corresponding to a sound speed $c_s = 0.2$ km s⁻¹), and mass $M = 120 M_\odot$. We impose a turbulent velocity field within our box using the driving scheme of Mac Low (1999), as it has been implemented for GADGET by Jappsen et al. (2005). Turbulence is driven on large scales (at wavenumbers between $k = 1$ and $k = 2$), aiming at an rms Mach number equal to $M_{\text{rms}} \sim 3.9$. The rms velocity has then reached $V_{\text{rms}} = M_{\text{rms}} c_s \sim 0.78$ km s⁻¹.

³ We use the term particle code as the generic antonym of grid code. In general, a particle code is a numerical scheme that uses sampling points that are not fixed in space but rather move, resembling in this respect the properties of fluid particles. In particular, all particle codes used here are different implementations of the Smoothed Particle Hydrodynamics (SPH) technique first introduced by Gingold & Monaghan (1977) and Lucy (1977).

We drive turbulence for a few code unit times. The code unit for time is arbitrarily chosen to be equal to the free-fall time $t_{\text{ff}} = \sqrt{3\pi/(32G\rho_0)}$, of the initial homogeneous and static gas distribution. Using $\rho_0 = M/L^3$, we obtain $t_{\text{ff}} \sim 0.115$ Myr. This does not imply that the gas in our box is or will become self-gravitating. A useful time unit for the comparison of non-self-gravitating turbulent flows is the turbulent crossing time $t_{\text{cross}} = L/(2V_{\text{rms}}) = L/(2M_{\text{rms}}c_s) \sim 0.182$ Myr, which is defined as the time it takes for a typical turbulent fluctuation to cross half of the computational domain (defined equivalently to Kritsuk et al. 2007; Federrath et al. 2009b).

In Sect. 4.1, we discuss for how long the turbulence needs to be driven to reach a statistical steady state. After driving has finished, the density and the velocity of the GADGET-particle distributions are interpolated onto a grid with 256^3 cells. We use this grid as the initial condition for following the decay with the grid codes. The particle distribution at the end of the driving phase provides the initial conditions to the SPH codes.

We follow the decay of turbulence for about six turbulent crossing times t_{cross} . The self-gravity of the gas is kept switched off at all times and the influence of magnetic fields is neglected, i.e. we follow the hydrodynamic decay only. An isothermal equation of state is used. Periodic boundary conditions are adopted.

Snapshots were taken at 0.0, 0.06, 0.31, 0.62, 3.1, and 6.2 t_{cross} . For the SPH codes, the particle distribution is interpolated onto a grid at each snapshot. The grid codes naturally provide their density and velocity fields on grids. From these grids, we then calculate spatially averaged quantities like the rms Mach number, velocity power spectra, and probability distribution functions of several quantities including the density, the velocity, its derivatives, and combinations of density and velocity.

In this paper, we include turbulence decay experiments using the SPH codes GADGET, PHANTOM (runs: A, B), and VINE, as well as the grid codes ENZO, FLASH, TVD, and ZEUS. In the following paragraphs, the general features of the codes used in this work are listed briefly. For more details on each of the codes, please refer to the references given below. All codes were run in parallel. The parallelisation architectures that the codes were run on, the total number of CPUs used, and the number of CPU hours consumed are listed in Sect. 7 for each of the runs studied here.

Codes that are used to perform simulations of supersonic turbulence are chosen for their performance in the highly compressible regime typical of astrophysical flows. Thus, codes with the ability to capture accurately the sharp discontinuities and high density ratios that result in supersonic turbulence are preferred for this study.

Grid-based codes for supersonic flows are often based on finite volume Riemann-solvers using the Godunov scheme (see e.g. Toro 1997). By their conservative nature, these codes maintain correct shock speeds, and, because of the Riemann-solver approach in calculating fluxes, they maintain very sharp discontinuities across a shock (typically within a few zones). These methods are often implemented in a dimensionally split way (Strang 1968).

A very common such method used for astrophysical flows is the Piecewise Parabolic Method (PPM), described in Colella & Woodward (1984), which is formally third-order accurate in space for smooth flows but switches to linear order to maintain step-function-like shocks or contact discontinuities near such features. Although not typically crucial, a small amount of artificial viscosity is implemented to ensure no or little oscillations behind shocks. Both ENZO (Norman & Bryan 1999;

O’Shea et al. 2004) and FLASH (Fryxell et al. 2000; Dubey et al. 2008), used in this paper, are of this type. ENZO was used here in version 1.0.1, and FLASH was used in version 2.5. FLASH in particular has been extensively tested against laboratory experiments (Calder et al. 2002) and other codes (Dimonte et al. 2004; Heitmann et al. 2005). For the FLASH run of this work, a PPM diffusion parameter of $K = 0.1$ (Colella & Woodward 1984) was used, whereas for the ENZO run the PPM diffusion parameter was set to $K = 0.0$. This provides an additional test of the effects of artificial shock diffusion on the results of grid codes using the PPM. The effects of varying the PPM diffusion parameter has been investigated before: setting $K = 0.0$ provides less dissipation, but produces stronger post-shock oscillations and a more pronounced bottleneck-effect (Kritsuk et al. 2007; Federrath et al. 2009a).

Other methods that attempt to maintain flatness of the solution behind shocks include Total Variation Diminishing (TVD) methods, which impose restrictions on the reconstruction of flow variables to ensure that the total variation of variables strictly diminishes over time. The TVD code used here employs an approximate (second-order-accurate), Roe-type (upwind) Riemann solver and TVD interpolation to maintain sharp shocks and smooth flow behind the shocks. Recipes for building the isothermal MHD code based on the TVD scheme are presented in Kim et al. (1999). For the turbulence comparisons of this project, we have used an isothermal hydrodynamic version of the code.

The ZEUS code (Stone & Norman 1992a,b) is a second-order accurate code using the van Leer (1977) monotonic advection algorithm. It resolves shocks using artificial viscosity, and does not include explicit techniques for keeping shocks sharp. A staggered mesh approach is adopted: the velocity is stored at cell interfaces, and density and energy at cell centres. This version of ZEUS is however different from the official version of ZEUS-MP, which was employed for instance by Vernaleo & Reynolds (2006) and Hayes et al. (2006), in that the version used here was parallelised by Piontek.

An entirely different method is employed in Smoothed Particle Hydrodynamics (SPH) codes (e.g. Lucy 1977; Gingold & Monaghan 1977; Monaghan 1988, 1992, 2005). SPH codes do not involve a grid at all, but track (fixed)-mass packets of fluid in a Lagrangian sense. While this approach to fluid dynamics requires the use of an artificial viscosity to maintain shock structure (e.g. Monaghan & Gingold 1983), it has the advantage for highly compressible flows that resolution automatically increases in high-density regions, as the Lagrangian fluid packets follow the mass flow. Three SPH codes, namely GADGET, PHANTOM (runs: A, B), and VINE, are used in this work. More details of these codes are given below.

GADGET is an MPI parallel tree-SPH and N -body code designed by Springel et al. (2001). We here used GADGET version 1. The code uses individual and adaptive timesteps for all particles, and it combines this with a scheme for dynamic tree update. As a time integrator, it uses a variant of the leapfrog integrator, involving an explicit predictor step. The smoothing lengths are derived according to the commonly used M4 kernel (Monaghan & Lattanzio 1985; Springel et al. 2001). In the GADGET run performed here, we set the number of neighbours of each SPH particle to 40 ± 5 neighbours. The influence of changing the number of SPH neighbours N_{neigh} has been investigated by Attwood et al. (2007). They argued that using a fixed number of neighbours prohibiting any variation in N_{neigh} may reduce numerical dissipation. Commerçon et al. (2008) also studied the influence of changing the number of SPH neighbours in simulations of gravitational fragmentation. They find a weak dependency of their

results on the number of neighbours indicating that increasing N_{neigh} speeds up gravitational fragmentation slightly. All SPH codes used here employed roughly the same number of SPH neighbours (see below). A number of about $N_{\text{neigh}} \sim 50 \pm 5$ SPH neighbours is the typical setup for most SPH calculations reported in the literature. Thus, our comparison of SPH runs with each other should not be systematically affected by our choice of SPH neighbours. However, varying the number of neighbours should be investigated in a detailed systematic study of supersonic turbulence in the future.

PHANTOM is a low-memory, highly efficient SPH code written especially for studying non-self-gravitating problems. The code is made very efficient by using a simple neighbour finding scheme based on a fixed grid and linked lists of particles. In particular, it uses an $\eta = 1.2$ term in calculating the smoothing length h through $h = \eta (m/\rho)^{1/3}$. This corresponds to about 58 SPH neighbours in a uniform density distribution, though it is the $h - \rho$ relation (to a tolerance of 10^{-4}) that provides the deciding criterion, not the neighbour number. The code implements the full “grad-h” SPH formulation developed by Price & Monaghan (2004) and Price & Monaghan (2007), whereby the smoothing length and density are mutually dependent and iterated self-consistently, resulting in exact conservation of momentum, energy and entropy in the SPH equations. Shocks are treated using the Monaghan (1997) formulation of artificial viscosity in SPH as described in Price (2004), though modified slightly in PHANTOM to allow a more efficient calculation. Timestepping is performed using a Kick-Drift-Kick leapfrog integrator. The standard PHANTOM run is labeled as PHANTOM A. We conduct a second run with PHANTOM, labeled PHANTOM B, in which dissipation is reduced away from shocks using the viscosity switch proposed by Morris & Monaghan (1997).

VINE (Wetzstein et al. 2009; Nelson et al. 2009; Gritschneider et al. 2009) is an OpenMP parallel, tree-SPH and N -body code. The code scales linearly up to a high number of processors, and is designed for a combined usage of generic CPUs and the special purpose hardware GRAPE. Time integration is performed here with a Drift-Kick-Drift leapfrog integrator, which allows for individual particle timesteps. The smoothing lengths are derived according to the M4 kernel (Monaghan & Lattanzio 1985). In the VINE simulations, we set the number of neighbours of each SPH particle to 50 ± 5 . Shocks are treated with the time dependent artificial viscosity prescription introduced by Morris & Monaghan (1997) (i.e. similar to the PHANTOM B run).

3. Methods for the analysis of statistical measures

3.1. SPH interpolation onto a grid

For interpolating the density and the velocity distribution of the SPH particles onto a grid, we use the generic SPH interpolation formula (e.g. Monaghan 1992)

$$A(\mathbf{r}) = \sum_i \frac{A_i m_i}{\rho_i h_i^3} W\left(\frac{|\mathbf{r} - \mathbf{r}_i|}{h_i}\right), \quad (1)$$

where ρ_i is the density, m_i is the mass, and h_i is the smoothing length of the i -th particle. The vector \mathbf{r} is the position vector to the centre of each grid cell, A_i is the particle quantity to be interpolated to the grid (density and velocity for our purposes), and $W(s)$ is the kernel function used for smoothing the particle

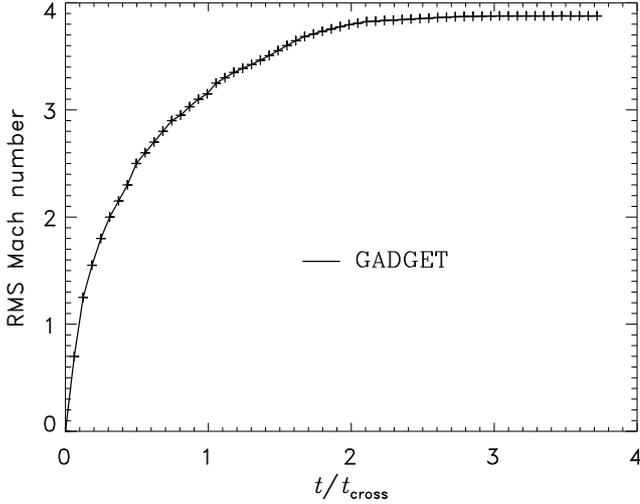


Fig. 1. Time evolution of the rms Mach number during the driving phase. After driving for about 2 turbulent crossing times t_{cross} (see text), the rms Mach number has reached a statistical steady state. The initial conditions for the decaying turbulence code comparison were chosen randomly at $t = 2.5t_{\text{cross}}$ in the regime of fully developed supersonic turbulence, $t > 2t_{\text{cross}}$ (e.g. Federrath et al. 2009b), when the rms Mach number has reached its statistical steady state of $M_{\text{rms}} \sim 3.9$.

mass in space to derive an SPH density. The M4 kernel, which is based on spline functions (Monaghan & Lattanzio 1985),

$$W(s) = \frac{1}{\pi} \begin{cases} 1 - 3s^2/2 + 3s^3/4, & 0 \leq s \leq 1, \\ (2 - s)^3/4, & 1 \leq s \leq 2, \\ 0, & s \geq 2, \end{cases} \quad (2)$$

is used for the interpolation. For each grid point, the above summation is over a limited number of neighbouring SPH particles due to the compact support of the kernel function, which vanishes for $|\mathbf{r} - \mathbf{r}_i| \geq 2h_i$.

3.2. Volume-weighted velocity power spectrum

The velocity power spectrum is calculated as follows. For each velocity component we take the Fourier transform of the velocity field $\mathbf{v} = (v_1, v_2, v_3)$. We denote these Fourier transforms as $\mathbf{v}' = (v'_1, v'_2, v'_3)$. Using these definitions the volume-weighted velocity power spectrum is defined as

$$E(k) = \frac{1}{2} (\mathbf{v}' \cdot \overline{\mathbf{v}'}) , \quad (3)$$

where $\overline{\mathbf{v}'} = (\overline{v'_1}, \overline{v'_2}, \overline{v'_3})$ is the complex conjugate of the transformed velocity. A wavenumber mapping, $\mathbf{k} = (k_1, k_2, k_3)$, is applied on the cells of the $E(k)$ -cube, with each k_1, k_2 , and k_3 ranging from $-N/2$ to $(N/2) - 1$. The compressible (longitudinal) part of the velocity power spectrum is calculated as

$$E_{\text{com}}(k) = E(k) \frac{(\mathbf{v}' \cdot \mathbf{k})(\overline{\mathbf{v}'} \cdot \mathbf{k})}{(\mathbf{k} \cdot \mathbf{k})(\mathbf{v}' \cdot \overline{\mathbf{v}'})} , \quad (4)$$

and the solenoidal (transverse) part as

$$E_{\text{sol}}(k) = E(k) - E_{\text{com}}(k) . \quad (5)$$

For each wavenumber $k = |\mathbf{k}|$, we collect from the $E(k)$ -cube all cells lying at distances in the $[k, k+1]$ interval⁴. The mean

⁴ Distances are measured with respect to the (0, 0, 0) cell, i.e. the central cell.

of the $E(k)$ -values of these cells is normalised by the area of the sphere element with radius $k + 0.5$. This gives the volume-weighted velocity power spectrum $E(k)$. The above process is repeated for $E_{\text{sol}}(k)$ and $E_{\text{com}}(k)$.

To calculate the volume-weighted dissipation rate, we estimate at each snapshot the integral over the volume-weighted velocity power spectrum. This is formally given as

$$\frac{1}{2} V_{\text{rms}}^2 = \int_0^\infty E(k) dk , \quad (6)$$

and is estimated numerically. The dissipation rate is then given as the rate of change of this integral as the decay proceeds.

The sonic scale, $l_s = 2\pi k_s$, separates supersonic turbulent flows on large scales (small k) from subsonic turbulent flows on smaller scales (high k). We estimate the sonic scale by solving the following equation,

$$\frac{1}{2} c_s^2 \simeq \int_{k_s=2\pi/l_s}^\infty E(k) dk , \quad (7)$$

implicitly for the sonic wavenumber k_s .

3.3. Density-weighted velocity power spectrum

For density-weighted velocity power spectra, we substitute

$$\mathbf{v}_{\text{mw}} = (\rho/\rho_0)^{1/2} \mathbf{v} , \quad (8)$$

where ρ_0 is the mean density in the cube. Then, the above process for the calculation of velocity power spectra is repeated. The density-weighted velocity power spectrum is defined as

$$E_{\text{mw}}(k) = \frac{1}{2} (\mathbf{v}'_{\text{mw}} \cdot \overline{\mathbf{v}'_{\text{mw}}}) , \quad (9)$$

in analogy to Eq. (3). The density-weighted dissipation rate is computed in analogy to Eq. (6), from the density-weighted velocity power spectrum $E_{\text{mw}}(k)$.

3.4. Probability distribution functions

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, $\omega = |\nabla \times \mathbf{v}|$. To obtain these distributions, the corresponding quantities are binned linearly. From the cumulative distribution, F , we derive the probability distribution function (PDF) of quantity A at each bin n by computing

$$\text{PDF}_n(A) = \frac{F_n(A) - F_{n-1}(A)}{A_n - A_{n-1}} . \quad (10)$$

For the trace free rate of strain, the spatial derivatives of the velocity components are first calculated as

$$v_{i,j} = \frac{\partial v_i}{\partial x_j} . \quad (11)$$

The (symmetric) strain tensor has components, $S_{ij} = 0.5(v_{i,j} + v_{j,i})$. The rate of strain is then

$$|S|^2 = 2 \sum_i \sum_j S_{ij} S_{ij} . \quad (12)$$

The trace of the strain tensor is $d = \sum_i S_{ii}$, so that the trace free strain tensor has components $S_{ij}^* = S_{ij} - \delta_{ij} d/3$, where δ_{ij} is

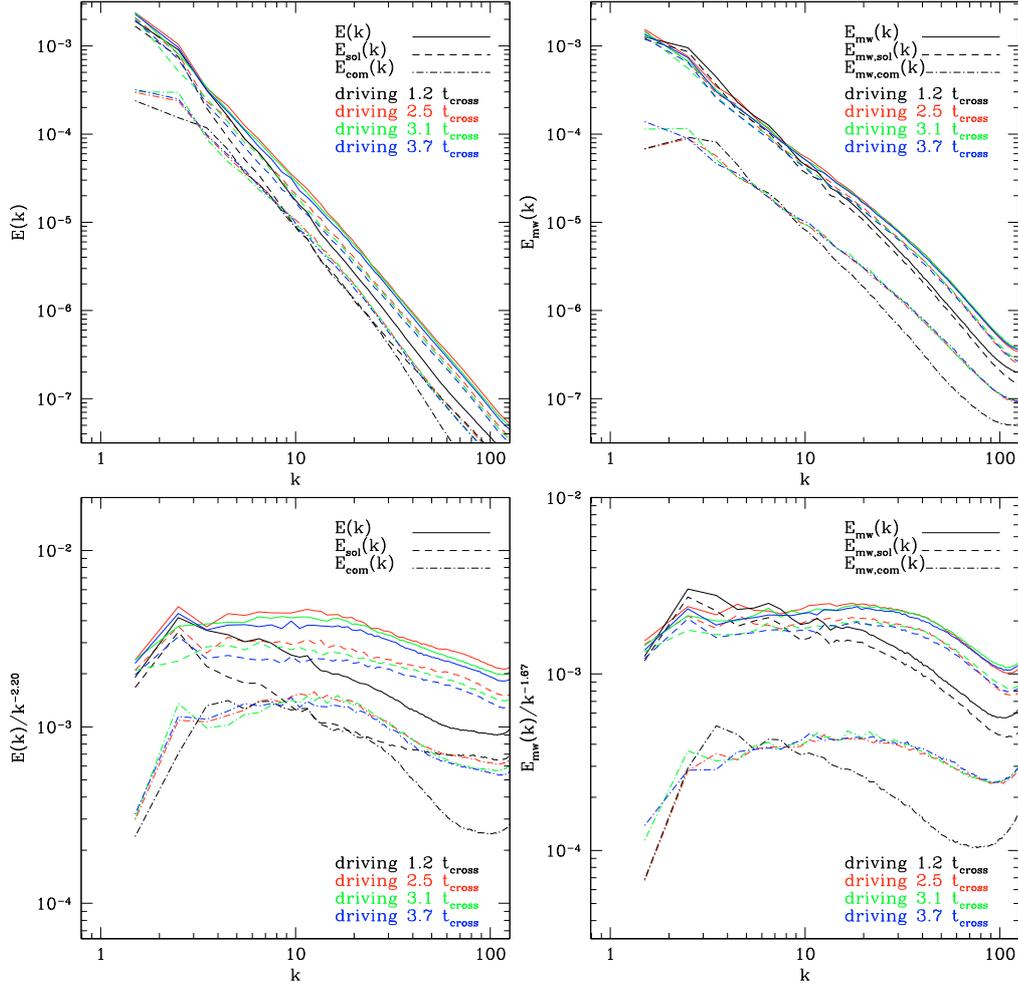


Fig. 2. *Top:* velocity power spectra obtained at four different snapshots during the driving phase (volume-weighted spectra on the left, and density-weighted spectra on the right panels). The spectrum of each snapshot is plotted with a different colour (1.2 t_{cross} : black lines; 2.5 t_{cross} : red lines; 3.1 t_{cross} : green lines; 3.7 t_{cross} : blue lines). The decay experiments using the SPH and grid codes starts with the snapshot at 2.5 t_{cross} when turbulence is fully established. The solid lines correspond to $E(k) = E_{\text{sol}} + E_{\text{com}}$ (volume-weighted, *left*), and $E_{\text{mw}}(k) = E_{\text{mw,sol}} + E_{\text{mw,com}}$ (density-weighted, *right*), the dashed lines to the solenoidal (transverse) part, and the dash-dotted lines to the compressible (longitudinal) part of the spectra. *Bottom:* The same spectra as on the top, but compensated with power-law slopes of 2.20 (*left panel* – volume-weighted case) and 1.67 (*right panel* – density-weighted case).

the Kronecker unit function. The trace free rate of strain then becomes

$$|S^*|^2 = 2 \sum_i \sum_j S_{ij}^* S_{ij}^*. \quad (13)$$

$|S^*|$ gives the rate at which a fluid element is deformed without changing its volume, e.g. by the act of a shear flow.

4. Driving phase

4.1. Driving time

We simulate driven turbulence using GADGET following the prescription in Jappsen et al. (2005). The driving phase starts with an initially homogeneous density distribution with $\rho_0 = 3.3 \times 10^{-19} \text{ g cm}^{-3}$ at rest. As turbulence gets driven, the rms Mach number increases with time. It levels off at a value of $M_{\text{rms}} \sim 3.9$ after about 4 t_{ff} (see Fig. 1). This time corresponds to roughly 2.5 crossing times, t_{cross} . We start the decay experiments at this time, i.e. after turbulence has been driven for 2.5 t_{cross} . The particle distribution obtained at this time is interpolated onto a grid. This grid is used as the initial condition for the decay simulations using the grid codes. For the decay simulations using the

SPH codes, we directly use the particle distribution obtained with GADGET at the same time ($t = 2.5 t_{\text{cross}}$). For the following calculations the time was reset to zero, and the turbulence decay was followed over roughly six crossing times with each of the grid and SPH codes.

Turbulence has been established when the decay experiments start. This is shown in Fig. 2, where we plot the velocity power spectra, calculated as explained in Sect. 3.2 and Sect. 3.3 for the volume- and density-weighted spectra, respectively, of four different snapshots taken along the driving phase. On the left panels, volume-weighted spectra are plotted whereas density-weighted spectra are shown on the right panels. Driving for 1.2 t_{cross} (black lines) was not sufficient to produce a statistically fully established turbulent flow, as there was not enough time for turbulence (driven on large scales) to cascade down to the smallest spatial scales. However, the turbulence is fully established after about 2.5 t_{cross} : there is no significant variation of the velocity power spectra when we attempted to drive turbulence for longer times (cf. the red, green and blue lines corresponding to driving times of 2.5, 3.1, and 3.7 t_{cross} , respectively). We conclude that starting the decay with the gas distribution obtained after 2.5 t_{cross} of driving is a reasonable

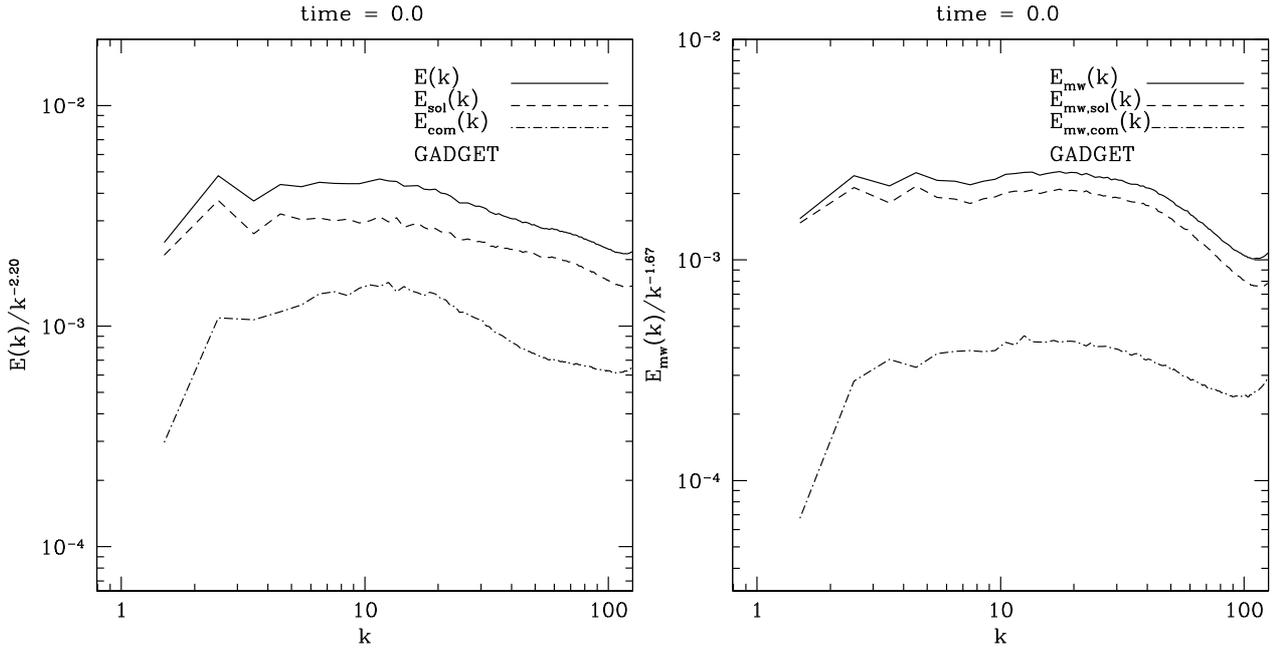


Fig. 3. Velocity power spectra of the initial conditions used for the decay experiments (volume-weighted spectrum on the left, and density-weighted spectrum on the right panel). The spectra were compensated with power-law slopes of 2.20 (*left panel* – volume-weighted case) and 1.67 (*right panel* – density-weighted case).

choice of initial conditions. Schmidt et al. (2009), Federrath et al. (2009b), and Glover et al. (2009) also conclude that after about $2 t_{\text{cross}}$, supersonic turbulence has established a statistical invariant state. However, significant statistical fluctuations from snapshot to snapshot remain (Federrath et al. 2009a), which explains the slight changes visible in the velocity power spectra at $t = 2.5$, 3.1 , and $3.7 t_{\text{cross}}$ in Fig. 2. The variations seen in Fig. 2 for $t \gtrsim 2.5 t_{\text{cross}}$ are at most in the order of the typical snapshot-to-snapshot variations introduced by intermittent fluctuations, i.e., less than 10% (see also Kritsuk et al. 2007, Fig. 1). The initial conditions used for our code comparison therefore constitute a statistically fully established supersonic turbulent density and velocity distribution.

4.2. The result of driving: initial conditions for the decay experiments

In this section, we present the velocity power spectra of the initial conditions used for the decay experiments. These initial conditions have been produced with GADGET using the turbulence driving routine developed by Mac Low (1999), and employed in Jappsen et al. (2005). They present the state of the system after $2.5 t_{\text{cross}}$ of driving (see Sect. 4.1). On the left panel of Fig. 3, we plot the volume-weighted velocity spectrum, with the density-weighted velocity spectrum shown on the right panel. The spectra were compensated with power-law slopes of 2.20 (left panel – volume-weighted case) and 1.67 (right panel – density-weighted case).

4.2.1. Volume-weighted velocity power spectrum of the initial conditions

From the volume-weighted velocity power spectrum computed with GADGET we derive a slope of about 2.2, which is obtained in the wavenumber range $4 \lesssim k \lesssim 12$. If any inertial-range scaling

could be inferred at all due to our limited numerical resolution, it may only exist in the close vicinity of $k \sim 8$.

In the presence of a bottleneck effect (e.g., Dobler et al. 2003; Haugen & Brandenburg 2004; Schmidt et al. 2006a), Porter & Woodward (1994) argued that the bottleneck affects all scales up to $k \sim N/32 = 8$ for grids of size $N = 256$. Schmidt et al. (2006a) suggested that the bottleneck peaks at $k \sim N/10 \sim 26$. These authors also argued that, in codes showing no bottleneck, numerical dissipation will start acting at wavenumbers smaller than $k \sim N/10$. Since our initial conditions do not seem to exhibit a bottleneck, dissipation will certainly start at $k \lesssim 26$. Since a power law is established for scales $4 \lesssim k \lesssim 12$, and since this power law breaks down at $k \gtrsim 12$, we argue that dissipation did not play a significant role for wavenumbers $k \lesssim 12$ in the spectra of the initial conditions.

Padoan et al. (2007) and Kritsuk et al. (2007) performed high-resolution hydrodynamic simulations of driven turbulence using ENZO at resolutions of 1024^3 grid cells. They obtained a significantly shallower slope of 1.9 for the volume-weighted velocity spectrum. Federrath et al. (2009a) showed that driving turbulence with the FLASH code at resolutions of 512^3 and 1024^3 grid cells results in a slope consistent with those of Padoan et al. (2007) and Kritsuk et al. (2007). In contrast, at the resolution of 256^3 (as used here), a steeper slope of the order of 2.1 was obtained (Federrath et al. 2009a), which is consistent with our result for the volume-weighted spectra. Federrath et al. (2009a) directly demonstrated the steepening of the velocity spectrum with decreasing numerical resolution (cf. their Fig. C.1). However, they also find that the slope of the velocity spectrum is converged to within less than 3% going from 512^3 to 1024^3 in their numerical experiments. Considering the low resolution in the present study, it is therefore not surprising that we find a slope of about 2.2 for the volume-weighted velocity power spectrum. This result however can also be taken as an indication that the initial conditions used for our comparison experiments did not strongly depend on the method by which these initial conditions have been produced (i.e. GADGET).

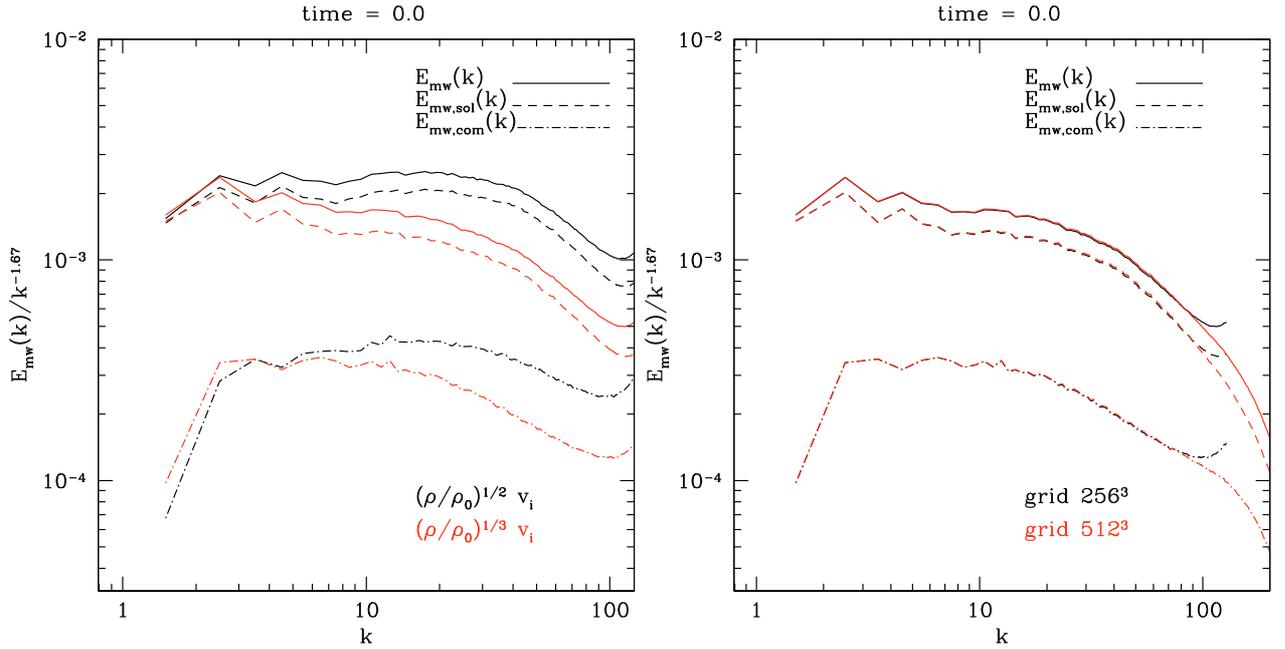


Fig. 4. *Left panel:* density-weighted velocity power spectra of the initial conditions using different velocity weights: $\mathbf{v}_{\text{mw}} = (\rho/\rho_0)^{1/2} \mathbf{v}$ (black lines), and $\mathbf{v}_{\text{mw}} = (\rho/\rho_0)^{1/3} \mathbf{v}$ (red lines). *Right panel:* density-weighted velocity power spectra [$\mathbf{v}_{\text{mw}} = (\rho/\rho_0)^{1/3} \mathbf{v}$] of the initial conditions interpolated to grids of 256^3 cells (black lines) and 512^3 cells (red lines).

4.2.2. Density-weighted velocity power spectrum of the initial conditions

From the density-weighted velocity power spectrum we obtain a scaling close to the Kolmogorov (1941) scaling with a slope of about $5/3$. This slope is found within $5 \leq k \leq 20$. We have used $\mathbf{v}_{\text{mw}} = (\rho/\rho_0)^{1/2} \mathbf{v}$ (see Sect. 3.3) instead of $\mathbf{v}_{\text{mw}} = (\rho/\rho_0)^{1/3} \mathbf{v}$, which was used by Kritsuk et al. (2007), Kowal & Lazarian (2007), Schmidt et al. (2008), and Federrath et al. (2009a). The $(\rho/\rho_0)^{1/2}$ weights correspond to a quantity that has physical reference to kinetic energy $(1/2) \rho |\mathbf{v}|^2$, while the $(\rho/\rho_0)^{1/3}$ weights correspond to a constant kinetic energy dissipation rate within the inertial range (Kritsuk et al. 2007). For GADGET only, we additionally compute energy spectra with the $(\rho/\rho_0)^{1/3}$ weights. The comparison of the $(\rho/\rho_0)^{1/2}$ to the $(\rho/\rho_0)^{1/3}$ weights is shown in Fig. 4 (left panel). We find a steeper slope of about 1.8 for the $(\rho/\rho_0)^{1/3}$ weights. The fact that Kritsuk et al. (2007) obtain Kolmogorov-type scaling using the $(\rho/\rho_0)^{1/3}$ weights is a consequence of their volume-weighted velocity power spectrum being shallower than ours with slopes of 1.9 and 2.2, respectively. We find this steeper slope of about 2.2 due to our limited numerical resolution as discussed in the previous section. Therefore, the fact that our density-weighted spectra show scaling close to Kolmogorov scaling is a result of the rather small numerical resolution adopted in this comparison. Clearly, the slopes of the density-weighted spectra depend not only on the velocity statistics, but also on the convolution of density and velocity statistics.

However, we argue that the density information should be taken into consideration in the statistical analysis of compressible turbulence, as most of the mass ends up in small volumes through shocks. This fact is neglected by statistical measures that take into account volume-weighted velocities only (for instance, some models of the mass distribution of molecular cloud cores and stars are based on the volume-weighted velocity power spectrum, e.g., Padoan & Nordlund 2002).

4.2.3. The effective SPH resolution

We would like to comment here on the Lagrangian nature of the SPH method. In Fig. 3, there is a rise in power on scales $k \gtrsim 100$, particularly prominent in the density-weighted case (right panel). This is a consequence of the adaptivity in resolution that is intrinsic to the SPH method: as the SPH particles move with the flow, the build-up of high densities is accompanied by an increase in the number of particles (sampling-points) for a fixed volume element. Therefore, as high densities build up on small scales due to the shocks developed in supersonic turbulence, the SPH particle concentration increases on these scales. In particular, in driven turbulence the effective SPH resolution in high-density regions eventually becomes superior to the resolution initially employed. Hence, the extra power developed on scales $k \gtrsim 100$ is a result of the interpolation of the SPH particle distribution onto a grid of resolution lower than the effective SPH resolution⁵. When the SPH particle distribution is interpolated onto a larger grid ($N = 512^3$) this rise in power is no longer observed (right panel of Fig. 4). In other words, because of the finite extent of the grid we used for our comparison experiments ($N = 256^3$), all SPH information on scales smaller than k_{max} gets interpolated into the k_{max} bin. This appears as a rise in the power on the smallest scales of our grid.

5. Results of the turbulence decay code comparison

5.1. Volume-weighted and density-weighted velocity power spectra

Figures 5 and 6 present the time evolution of the volume- and density-weighted velocity power spectra obtained with the various codes employed in this work. Data from the following

⁵ Note that the total number of sampling points for the SPH runs (215^3 particles) was smaller than the number of sampling points employed for our grids (256^3 grid cells).

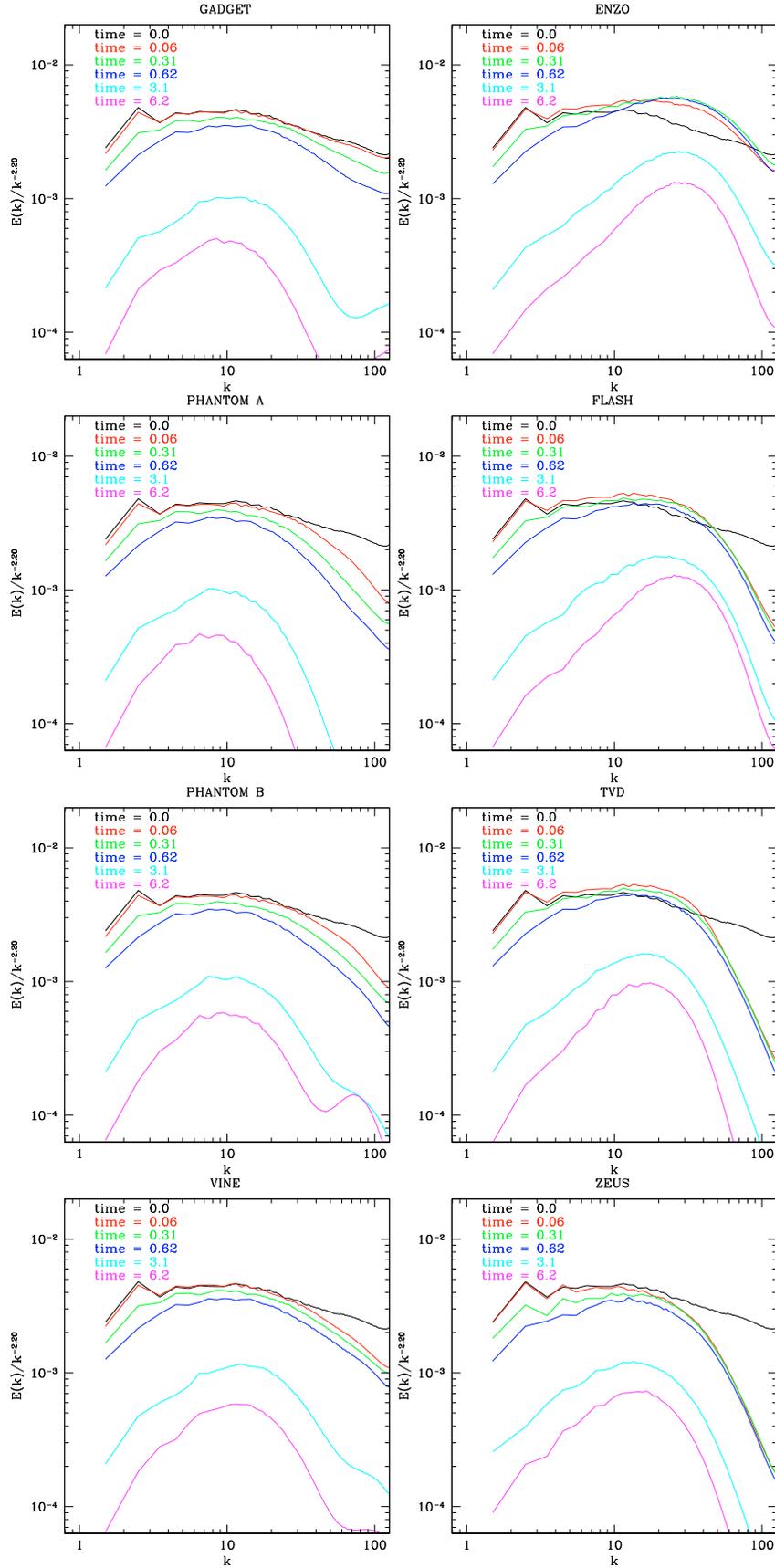


Fig. 5. Time evolution of the volume-weighted velocity power spectra (compensated with power-law slopes of 2.20) of the SPH codes GADGET, PHANTOM (runs: A, B), and VINE (*left column*) and of the grid codes ENZO, FLASH, TVD, and ZEUS (*right column*) for the following snapshots: $t = 0.0, 0.06, 0.31, 0.62, 3.1,$ and $6.2 t_{\text{cross}}$.

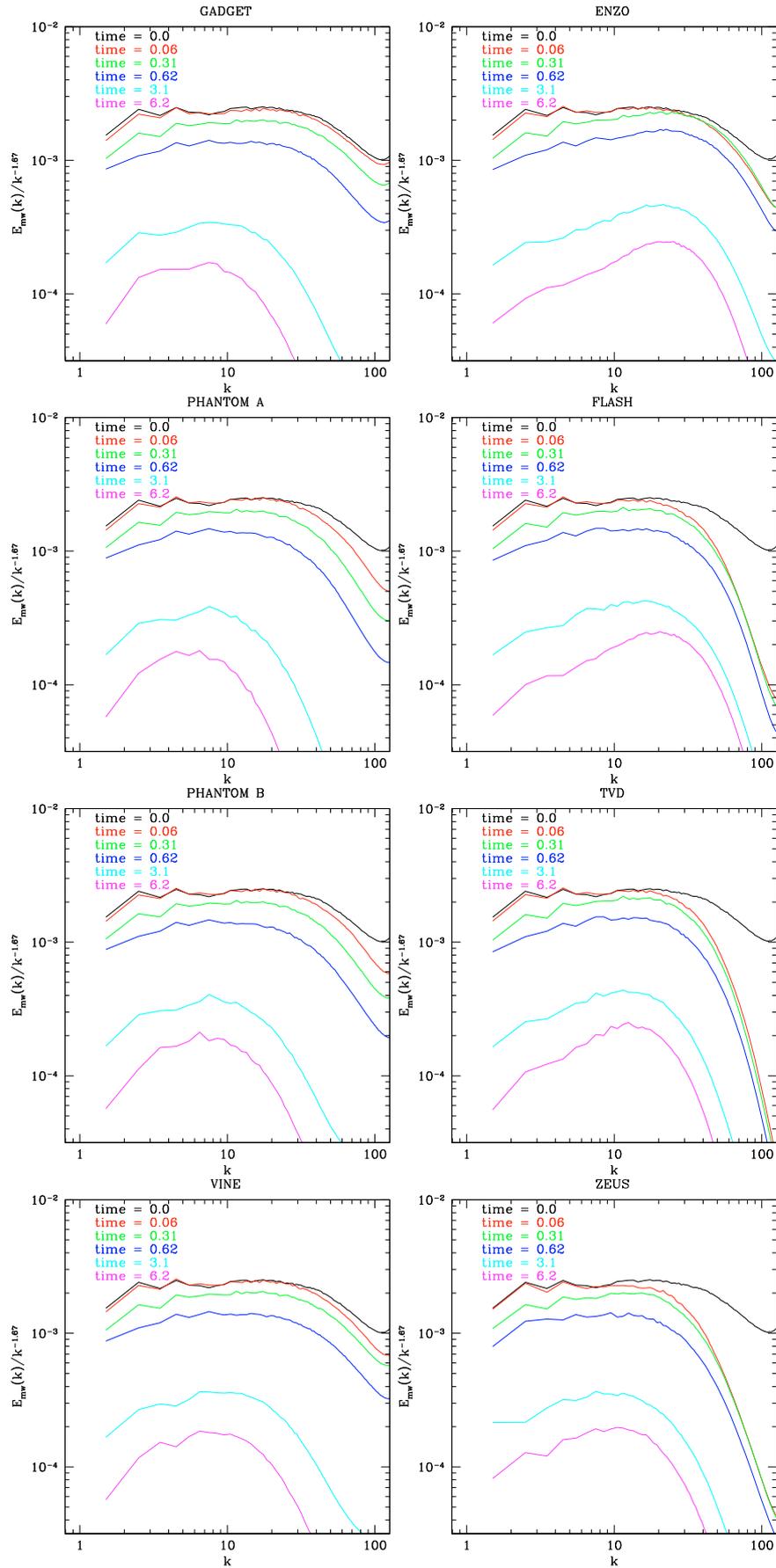


Fig. 6. Same as Fig. 5, but the density-weighted velocity power spectra (compensated with power-law slopes of 1.67) are shown.

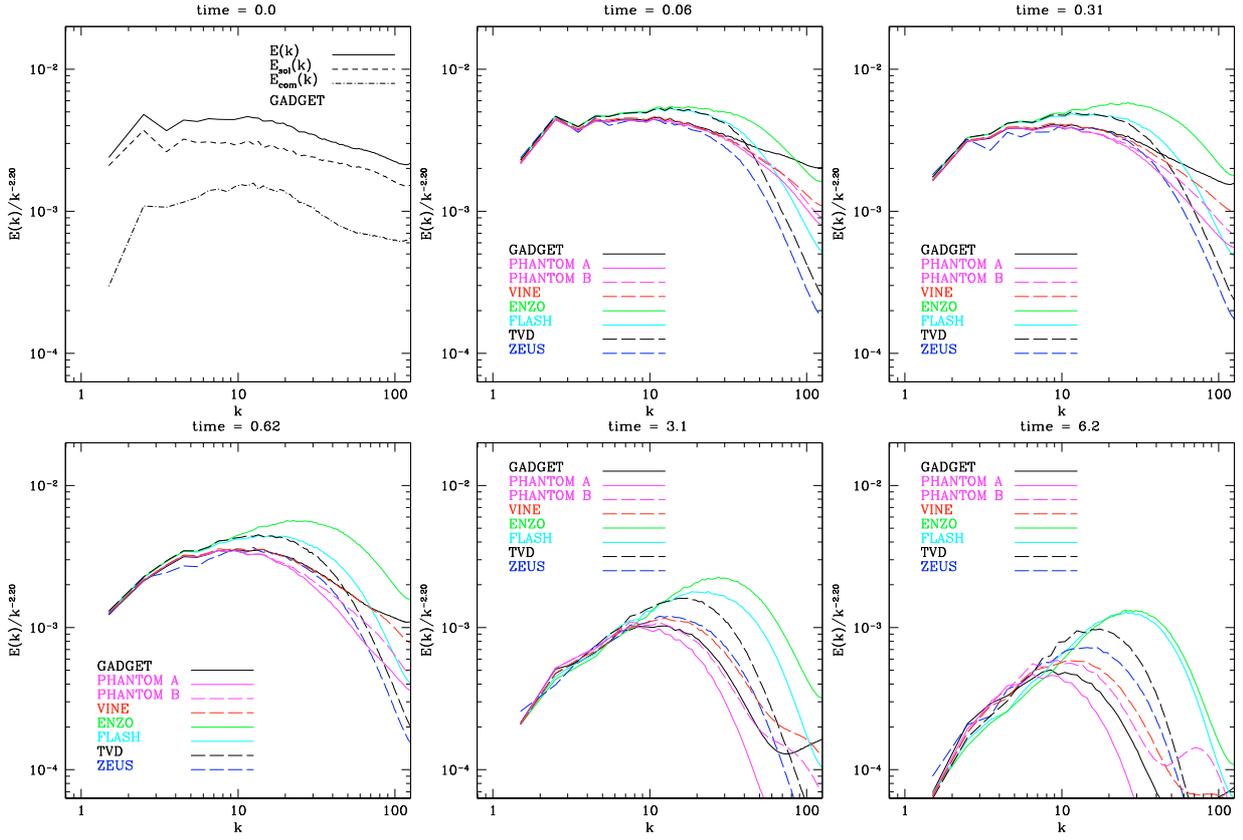


Fig. 7. Comparison of the volume-weighted velocity power spectra (compensated with power-law slopes of 2.20) of all codes at different times along the decay: $t = 0.0, 0.06, 0.31, 0.62, 3.1,$ and $6.2 t_{\text{cross}}$.

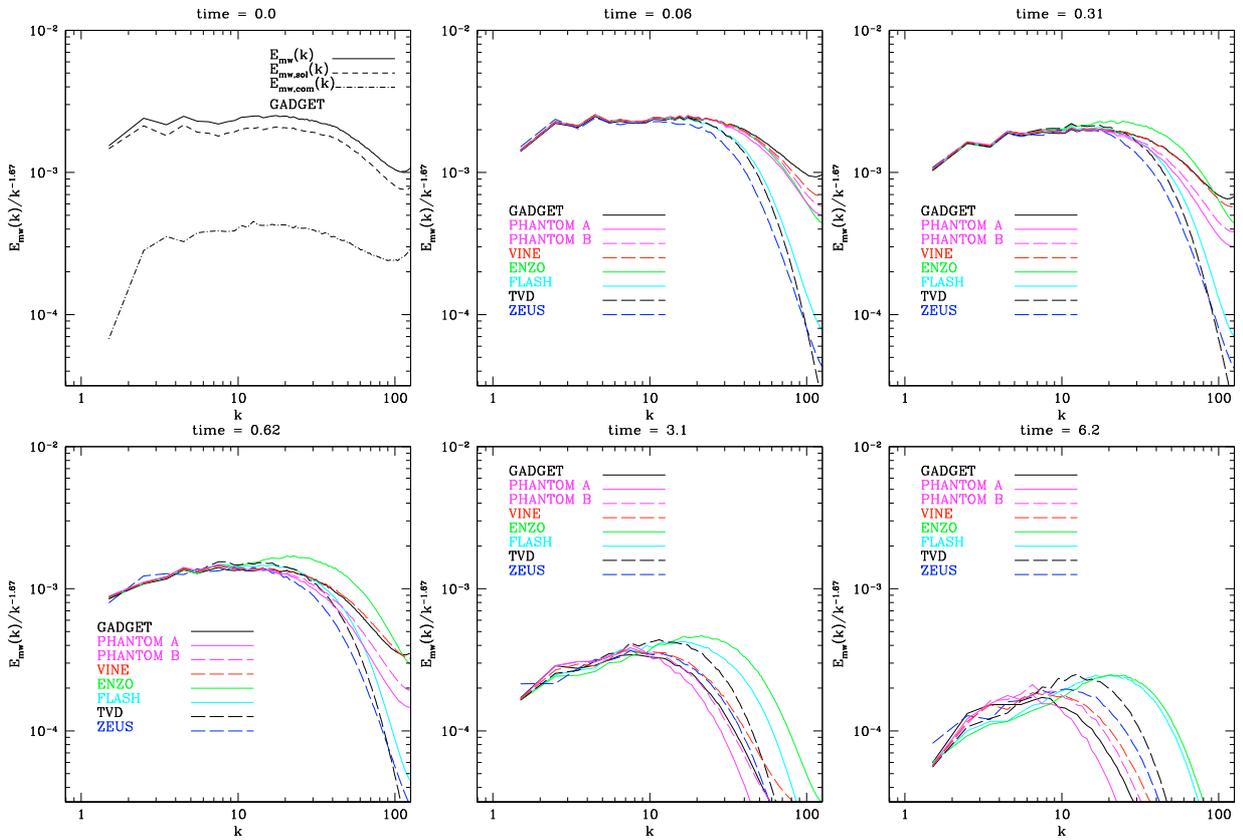


Fig. 8. Same as Fig. 7, but the density-weighted velocity power spectra (compensated with power-law slopes of 1.67) are shown.

snapshots are shown: initial conditions at $t = 0.0 t_{\text{cross}}$ (black lines), $t = 0.06 t_{\text{cross}}$ (red lines), $t = 0.31 t_{\text{cross}}$ (green lines), $t = 0.62 t_{\text{cross}}$ (blue lines), $t = 3.1 t_{\text{cross}}$ (cyan lines), and $t = 6.2 t_{\text{cross}}$ (magenta lines). Figures 7 and 8 show the volume- and density-weighted velocity power spectra of each code plotted on top of each other in a single plot as a function of time, so that the spectra obtained for each snapshot can be directly compared across all codes for each time.

The grid codes dissipate the power produced on small spatial scales ($k \gtrsim 100$) of the initial conditions faster than the SPH codes. This is a result of the SPH interpolation onto the grid (see Sect. 4.2.3). Due to their Lagrangian nature the SPH runs (GADGET, PHANTOM A, PHANTOM B, and VINE), maintained power at $k \gtrsim 20$ for a longer time. All grid codes have lost slightly more power for $k \gtrsim 20$ than the SPH codes, immediately after the decay simulations start (i.e. from the first snapshot at $0.06 t_{\text{cross}}$). The differences seen at early times on the small scales of the SPH power spectra is a result of the different methods that each of the SPH codes adopt for calculating particle smoothing lengths and/or the use of different smoothing kernels, and different implementations of artificial viscosity.

The volume-weighted velocity power spectra obtained with the SPH codes and with the grid code ZEUS are quite similar for $k \lesssim 20$ at $t = 0.06 t_{\text{cross}}$, with power law slopes of about 2.2. ENZO, FLASH, and TVD have a slightly shallower slope of about 2.1. This slope agrees with the low-resolution models in Federrath et al. (2009a) using 256^3 grid cells. At $t = 0.31$, and $0.62 t_{\text{cross}}$, also ZEUS develops a slightly shallower slope that roughly agrees with the slopes obtained using the other grid codes. The slopes have dropped to about 1.95 and 1.9 at $t = 0.31$, and $0.62 t_{\text{cross}}$, respectively. The wavenumber ranges over which these slopes are maintained are slightly smaller for ZEUS than for TVD and FLASH (up to $k \sim 12$), while ENZO maintains the slopes up to $k \sim 18$. The SPH codes again have a slightly steeper slope by about 0.1 than the grid codes, and the wavenumber range over which this slope is maintained is comparable with the range obtained using the ZEUS code.

The density-weighted velocity power spectra are shallower than the volume-weighted spectra for all codes with slopes of about 1.6 at $t = 0.06 t_{\text{cross}}$. This much shallower slope is a result of the low resolution of our numerical experiments as discussed in Sect. 4.2.2. Similar to the results obtained from the volume-weighted spectra, all grid codes dissipate the initial power on scales $k \gtrsim 20$ faster than the SPH codes with ZEUS having dissipated most. However, there is an important exception to this result concerning the grid codes: the density-weighted velocity power spectrum produced by the grid code ENZO is almost identical to the power spectra produced with the SPH codes at $t = 0.06 t_{\text{cross}}$, while FLASH, TVD, and ZEUS have lost a considerable amount of their power at $k \gtrsim 20$. The power spectrum obtained with the ZEUS code shows the break into the dissipation range already at $k \sim 10$ and produces a slightly steeper slope of about 1.65 than all other codes. At later times ($t = 0.31 t_{\text{cross}}$, and $t = 0.62 t_{\text{cross}}$) all codes produced similar density-weighted power spectra for $k \lesssim 20$, while the ENZO code develops a clear bottleneck (see e.g. Dobler et al. 2003; Haugen & Brandenburg 2004; Schmidt et al. 2006a), which manifests itself in the excess power seen at $k \gtrsim 10$. Since ENZO was run here with a PPM diffusion parameter set to $K = 0.0$ (Colella & Woodward 1984), the bottleneck effect is quite strong (see also Kritsuk et al. 2007; Federrath et al. 2009a). Although the FLASH code uses the same numerical technique as ENZO it does not show such a pronounced bottleneck effect, because FLASH was used with the

recommended PPM diffusion parameter of $K = 0.1$ (Colella & Woodward 1984).

As the decay progresses ($t = 3.1$, and $6.2 t_{\text{cross}}$), power gets dissipated differently by the various codes at $k \gtrsim 8$. However, on large scales ($k \lesssim 8$), all codes used in the present study show very similar volume- and density-weighted velocity power spectra with slight variations that can be attributed to statistical fluctuations (Kritsuk et al. 2007; Federrath et al. 2009a). This is an important result, as it shows that all codes, despite having different dissipation mechanisms acting on small scales at $k \gtrsim 8$, on large spatial scales the results of the decaying turbulence experiments presented here are quite robust. They do not show considerable systematic differences for the codes employed here at the resolutions studied. Moreover, it is important to note that the dissipation ranges at $k \gtrsim 8$ are not just different when we compare grid codes with SPH codes, but they are also different among the SPH codes and among the grid codes. Thus, we conclude that the dissipation range is strongly dependent on the code being used, while scales with $k \lesssim 8$ are similarly well reproduced by all the hydrodynamic codes employed here.

The overall performance of the codes, as seen through the analysis of their velocity power spectra at times $t = 3.1 t_{\text{cross}}$, and $t = 6.2 t_{\text{cross}}$ shows that the numerical viscosity of grid codes is generally lower than that of SPH codes, and details of the method used will determine the detailed ranking. For example, using $K = 0$ in PPM, as done for the ENZO run, yields a lower dissipation value, although with a stronger bottleneck effect than the $K = 1$ value used for the FLASH run. Similarly, we find that the viscosity implementation by Morris & Monaghan (1997) used in the PHANTOM B and VINE runs is superior to that of GADGET and the PHANTOM A run.

Considering the resolution of $N^3 = 256^3$ cells used in the present study (215³ SPH particles interpolated to a 256^3 grid), the fact that the velocity power spectra are different at $k \gtrsim 8$ implies that one should be cautious with the interpretation of results obtained with power spectra at wavenumbers $k \gtrsim N/32$. This means that length scales smaller than about 32 grid cells for grid codes (and SPH codes using a similar number of resolution elements interpolated to a grid of equivalent size), are affected by the individual dissipation mechanisms acting in hydrodynamical codes. In contrast, the results of the various codes are robust for $k \lesssim N/32$. This is encouraging, because the results of all the hydrodynamical codes used here agree well in this regime, and one is free to choose a code for modelling supersonic turbulence as long as only results for scales $k \lesssim N/32$ are considered. However, this also means that one needs resolutions of at least 1024^3 grid cells to obtain roughly one full decade in length scale over which a power law could be fitted to turbulent velocity spectra. In practice, this range turns out to be even smaller than one decade in length scales at a resolution of 1024^3 grid cells (Kritsuk et al. 2007; Klein et al. 2007; Federrath et al. 2009a).

5.2. Kinetic energy dissipation rates

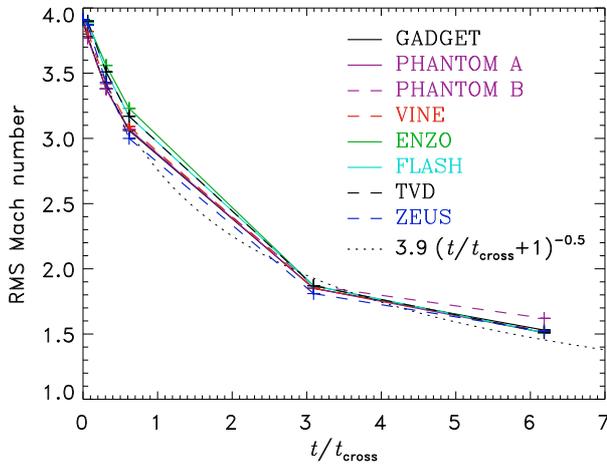
Tables 1 and 2 list the integrals over the volume- and density-weighted velocity spectra, respectively for each code and run at each of the snapshots presented here. Up to $t = 0.31 t_{\text{cross}}$, all SPH codes dissipate volume-weighted power slightly faster than the grid codes, while for the density-weighted power, both SPH and grid codes dissipate kinetic energy at roughly the same rate. At $t = 0.62 t_{\text{cross}}$, however, ZEUS has dissipated about 15% more power in velocity fluctuations than the other codes, while all other grid codes have still dissipated less than the SPH codes. The density-weighted integral gives more similar results for all

Table 1. Time evolution of $E_{\text{tot}} = \int E(k) dk$ in units of $[c_s^2]$, for all codes/runs.

Time $t [t_{\text{cross}}]$	GADGET	PHANTOM A	PHANTOM B	VINE	ENZO	FLASH	TVD	ZEUS
0.0	7.64	7.64	7.64	7.64	7.64	7.64	7.64	7.64
0.06	7.16	7.12	7.12	7.31	7.58	7.60	7.61	7.51
0.31	5.72	5.72	5.71	5.83	6.32	6.16	6.17	5.88
0.62	4.68	4.72	4.72	4.76	5.22	5.03	5.04	4.51
3.1	1.72	1.70	1.74	1.71	1.75	1.74	1.75	1.64
6.2	1.16	1.14	1.32	1.15	1.14	1.14	1.14	1.15

Table 2. Time evolution of $E_{\text{mw,tot}} = \int E_{\text{mw}}(k) dk$ in units of $[\rho_0 c_s^2]$, for all codes/runs.

Time $t [t_{\text{cross}}]$	GADGET	PHANTOM A	PHANTOM B	VINE	ENZO	FLASH	TVD	ZEUS
0.0	7.22	7.22	7.22	7.22	7.22	7.22	7.22	7.22
0.06	6.84	6.93	6.95	7.00	7.00	6.82	6.83	6.82
0.31	5.42	5.49	5.50	5.54	5.70	5.41	5.43	5.39
0.62	4.30	4.36	4.38	4.38	4.55	4.31	4.32	4.17
3.1	1.56	1.56	1.61	1.56	1.57	1.56	1.55	1.56
6.2	1.12	1.11	1.29	1.12	1.11	1.10	1.10	1.13

**Fig. 9.** Evolution of the rms Mach number as a function of time in units of the turbulent crossing time t_{cross} for all codes/runs. The dotted line shows the expected power-law decay rate $M_{\text{rms}} \propto t^{-1/2}$ for supersonic turbulence (Mac Low et al. 1998; Stone et al. 1998; Mac Low 1999).

codes at all times analysed. At $t = 3.1$, and $t = 6.2 t_{\text{cross}}$, all codes have roughly dissipated the same amount of volume- and density-weighted power, except for the PHANTOM B run having kinetic energies about 16% larger than all other codes.

5.3. Time evolution of the rms Mach number

The time evolution of the rms Mach number is shown in Fig. 9. The dotted line shows the expected power-law decay rate $M_{\text{rms}} \propto t^{-1/2}$ for supersonic turbulence (Mac Low et al. 1998; Stone et al. 1998; Mac Low 1999), starting at an rms Mach number of $M_{\text{rms}} \sim 3.9$:

$$M_{\text{rms}}(t) = 3.9 \left(\frac{t}{t_{\text{cross}}} + 1 \right)^{-1/2}. \quad (14)$$

Clearly, the rms turbulent flow remains supersonic (i.e. $M_{\text{rms}} > 1$) for all times analysed in the present study. However, turbulent velocity fluctuations become smaller on smaller scales. The transition scale separating supersonic motions on large scales and subsonic motions on small scales is called the sonic scale k_s (e.g., Vázquez-Semadeni et al. 2003; Federrath et al. 2009a).

We computed an estimate of the sonic scale using the definition of Eq. (7).

5.4. Time evolution of the sonic scale

Table 3 lists the evolution of the sonic scale for all codes/runs and all snapshots. The sonic scale decreases fastest for PHANTOM A, PHANTOM B and ZEUS, and slowest for ENZO. GADGET, VINE, FLASH and TVD show quite similar results for the evolution of the sonic scale. The sonic scale does not differ considerably among the various codes at later times ($t = 3.1 t_{\text{cross}}$ and $t = 6.2 t_{\text{cross}}$). However, during the initial stages of the decay, there are differences up to 30%. This is partly a result of our computation of the sonic scale (cf. Sect. 3.2). Since the different codes have quite different dissipation properties, the sonic scale is affected accordingly (cf. Sect. 5.1). Also note that the sonic scale is given as an integer. Thus, the fact that, for instance, ENZO maintains $k_s = 10$ until $t = 0.62 t_{\text{cross}}$ does not imply that its sonic scale stays exactly the same for all times $t < 0.62 t_{\text{cross}}$, but will have also decreased slightly. However, our grid of wavenumbers is binned such that only integer values of k are permitted, and thus, rounding errors introduce uncertainties of about 10% in the k_s -values at early times of the decay ($t \lesssim 1 t_{\text{cross}}$).

5.5. Probability distribution functions

5.5.1. Probability distribution functions of the gas density

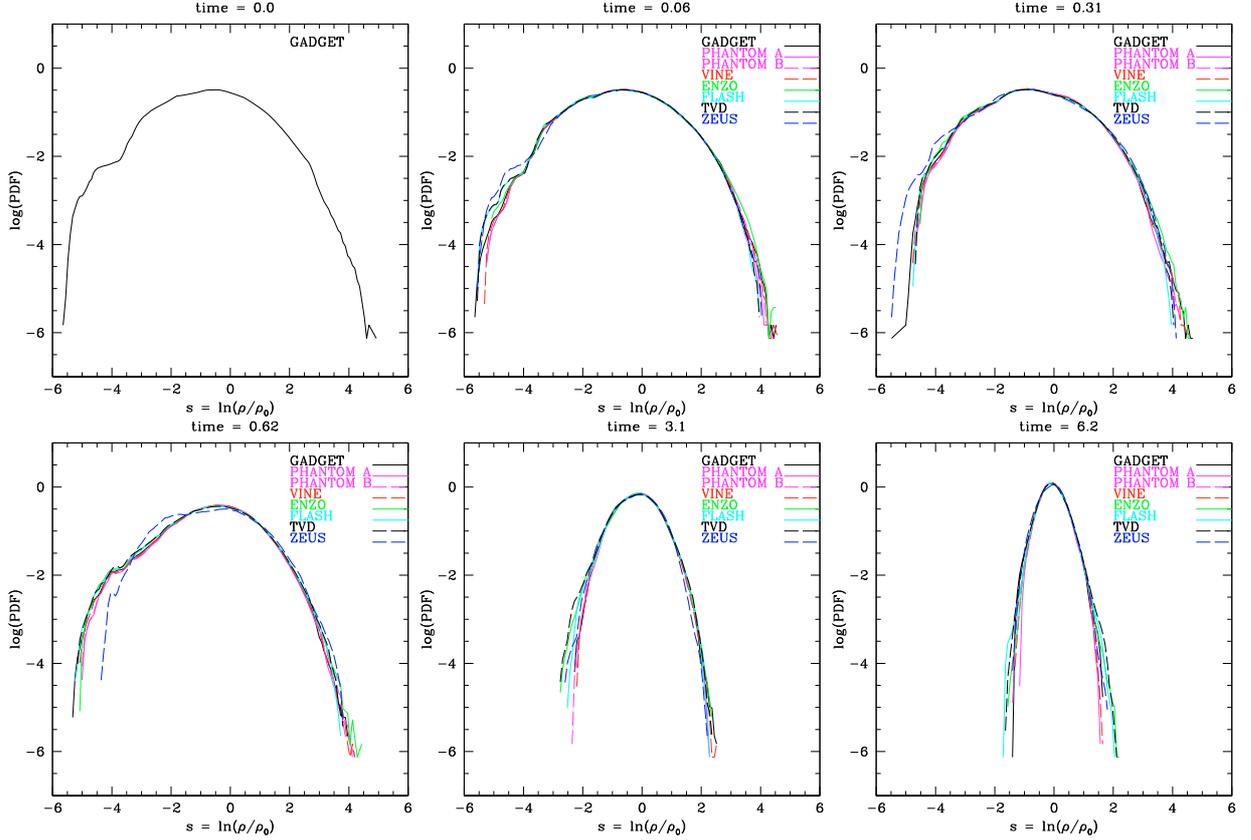
Figure 10 shows volume-weighted probability distribution functions (PDFs) of the gas density. Each panel shows the comparison of the density PDFs for all codes at $t = 0.0, 0.06, 0.31, 0.62, 3.1$, and $6.2 t_{\text{cross}}$ after they have been interpolated to grids of 256^3 cells. The density PDFs were computed from the logarithm of the density $s = \ln(\rho/\rho_0)$. The PDF $p(s)$ is expected to follow roughly a Gaussian distribution

$$p(s) ds = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left[-\frac{(s-s_0)^2}{2\sigma_s^2}\right] ds \quad (15)$$

where σ_s is the logarithmic density dispersion and s_0 is the mean value of s (e.g., Vázquez-Semadeni 1994; Padoan et al. 1997; Stone et al. 1998; Mac Low 1999; Nordlund & Padoan 1999; Ostriker et al. 1999; Klessen 2000; Ostriker et al. 2001;

Table 3. Time evolution of the sonic scale k_s as defined by Eq. (7) for all codes/runs.

Time t [t_{cross}]	GADGET	PHANTOM A	PHANTOM B	VINE	ENZO	FLASH	TVD	ZEUS
0.0	10	10	10	10	10	10	10	10
0.06	9	9	9	9	10	10	10	9
0.31	9	8	8	8	10	9	9	8
0.62	8	7	7	8	10	9	9	7
3.1	3	3	3	3	3	3	3	3
6.2	2	2	2	2	2	2	2	2

**Fig. 10.** Comparison of the volume-weighted density PDFs of all codes at different times along the decay: $t = 0.0, 0.06, 0.31, 0.62, 3.1,$ and $6.2 t_{\text{cross}}$.**Table 4.** Time evolution of the mean s_0 and standard deviation σ_s of the logarithmic density $s = \ln(\rho/\rho_0)$ for all codes/runs.

Time t [t_{cross}]	GADGET		PHANTOM A		PHANTOM B		VINE		ENZO		FLASH		TVD		ZEUS	
	s_0	σ_s	s_0	σ_s	s_0	σ_s	s_0	σ_s	s_0	σ_s	s_0	σ_s	s_0	σ_s	s_0	σ_s
0.0	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23
0.06	-0.72	1.22	-0.71	1.21	-0.71	1.21	-0.72	1.22	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23
0.31	-0.72	1.22	-0.70	1.20	-0.70	1.21	-0.72	1.22	-0.73	1.23	-0.73	1.23	-0.73	1.23	-0.73	1.23
0.62	-0.54	1.10	-0.52	1.08	-0.53	1.08	-0.54	1.09	-0.58	1.14	-0.58	1.13	-0.59	1.14	-0.68	1.19
3.1	-0.15	0.56	-0.15	0.55	-0.15	0.56	-0.15	0.56	-0.16	0.56	-0.16	0.56	-0.16	0.58	-0.16	0.58
6.2	-0.06	0.34	-0.05	0.32	-0.06	0.33	-0.06	0.34	-0.06	0.34	-0.06	0.34	-0.06	0.35	-0.06	0.36

Boldyrev et al. 2002a; Li et al. 2003; Padoan et al. 2004; Glover & Mac Low 2007; Kritsuk et al. 2007; Beetz et al. 2008; Federrath et al. 2008b; Lemaster & Stone 2008; Schmidt et al. 2009; Federrath et al. 2009a; Glover et al. 2009).

The density PDFs of all codes show little variation around the peak of the distribution and at the high-density tail, and they are all roughly consistent with log-normal distributions, Eq. (15). The low density tails show stronger variations. This is because the low density tail is subject to stronger temporal variations caused by intermittent fluctuations (Kritsuk et al. 2007; Federrath et al. 2009a). In Table 4 we list the values of s_0 and

σ_s for each code. Note that for a log-normal volume-weighted density distribution, $s_0 = -\sigma_s^2/2$. The means and standard deviations of the PDFs are similar for all codes and vary only by about 10%, except for our ZEUS run at $t = 0.62 t_{\text{cross}}$, which has a mean value $|s_0|$ about 28% larger than the average over all runs at that time. This appears slightly too high a variation to be attributed to temporal fluctuations. The difference of the density PDF around its peak obtained with the ZEUS run at $t = 0.62 t_{\text{cross}}$ is also visible in Fig. 10 (bottom left panel). However, at later times, the ZEUS density PDFs are almost identical to the ones obtained with the other codes.

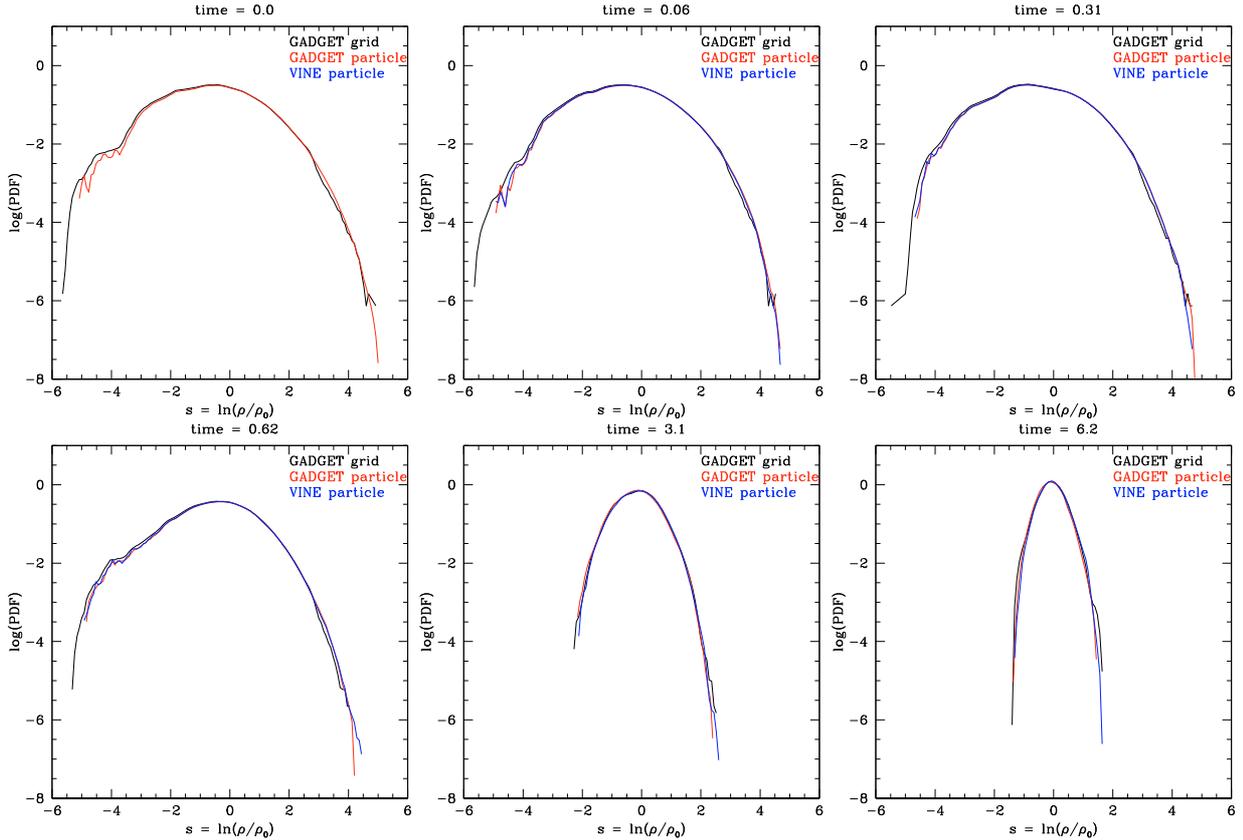


Fig. 11. Volume-weighted density PDFs at $t = 0.0, 0.06, 0.31, 0.62, 3.1,$ and $6.2 t_{\text{cross}}$ based on all cells of the interpolated grid (black line), and based on all SPH particle densities for GADGET (red line), and all SPH particle densities for VINE (blue line). The SPH particle density PDFs were computed on the SPH particles directly. Thus, they do not involve any interpolation to a grid, but they needed to be converted to a volume-weighted density PDF by using $p_v = p_m/\rho$, in order to allow for a direct comparison to the grid-interpolated PDF (black line). Both the density PDF computed from the grid and the SPH densities agree very well, with the SPH density PDF extending to slightly higher densities, and exhibiting slightly less scatter at the high-density tail.

One would expect that SPH codes can resolve high-density regions better than grid codes. This is because the effective SPH resolution increases with increasing density (cf. Sect. 4.2.3). On the other hand, low-density gas will become less resolved as particles move towards higher density regions and away from low-density regions. However, the density PDFs of grid codes and SPH codes agree very well after interpolation of the particle data to grids. This may be caused by our interpolation procedure. We therefore would like to test the density PDF obtained from the SPH particle density directly without interpolation to a grid. The density PDF of SPH particles is naturally mass-weighted. Therefore, in order to obtain the volume-weighted density PDF directly from the particles we must weight the contribution of each particle into a density bin by the inverse of its density (for equal mass particles). Note that in general, volume-weighted PDFs, p_v , are related to mass-weighted PDFs, p_m , by $p_v = p_m/\rho$ (e.g. Ostriker et al. 2001; Li et al. 2003). This conversion is necessary as particles and grid cells do not sample the computational volume in the same manner, i.e. the particles are concentrated in high-density regions, whereas the grid cells sample the volume homogeneously. The same applies for the distribution of Lagrangian tracer particles used in grid simulations (Federrath et al. 2008a; Price & Federrath 2009). Therefore, we must compensate the particle PDF for the fact that most particles are located in high-density regions.

Figure 11 shows the density PDF of the GADGET run as obtained both after interpolation to a grid (black line), and directly

obtained from the SPH particle density (red line). Additionally, we show the SPH density PDF obtained from the VINE run for comparison. The SPH density PDFs for GADGET and VINE were transformed into volume-weighted PDFs using $p_v = p_m/\rho$ to allow for a direct comparison of the grid and the SPH density PDFs. The grid-interpolated and the SPH density PDFs show no significant differences: the grid-interpolated PDF exhibits less scatter in the low-density regime than the SPH particle PDF, indicating that the grid-interpolated PDF samples the low-density regime slightly better than the particle-based PDF. On the other hand, the particle-based density PDF shows better sampling of the high-density tail and extends to slightly higher densities and lower probability densities. This is to be expected because the effective resolution is larger for the particle-based distribution at high densities, and SPH density peaks are smoothed to slightly smaller densities by the interpolation technique, Eq. (1). This analysis offers an additional illustration of the fact that the adaptivity of the SPH method is suppressed when the SPH information is interpolated onto a grid of resolution smaller than the effective SPH resolution (cf. Sect. 4.2.3).

It is an encouraging result that all codes are able to reproduce the general form of the density PDF quite well. This is important, because the turbulent density PDF is an essential ingredient to many models of star formation: to understand the mass distribution of cores in molecular clouds and stars (Padoan & Nordlund 2002; Hennebelle & Chabrier 2008, 2009), the star formation rate (Krumholz & McKee 2005; Padoan & Nordlund 2009),

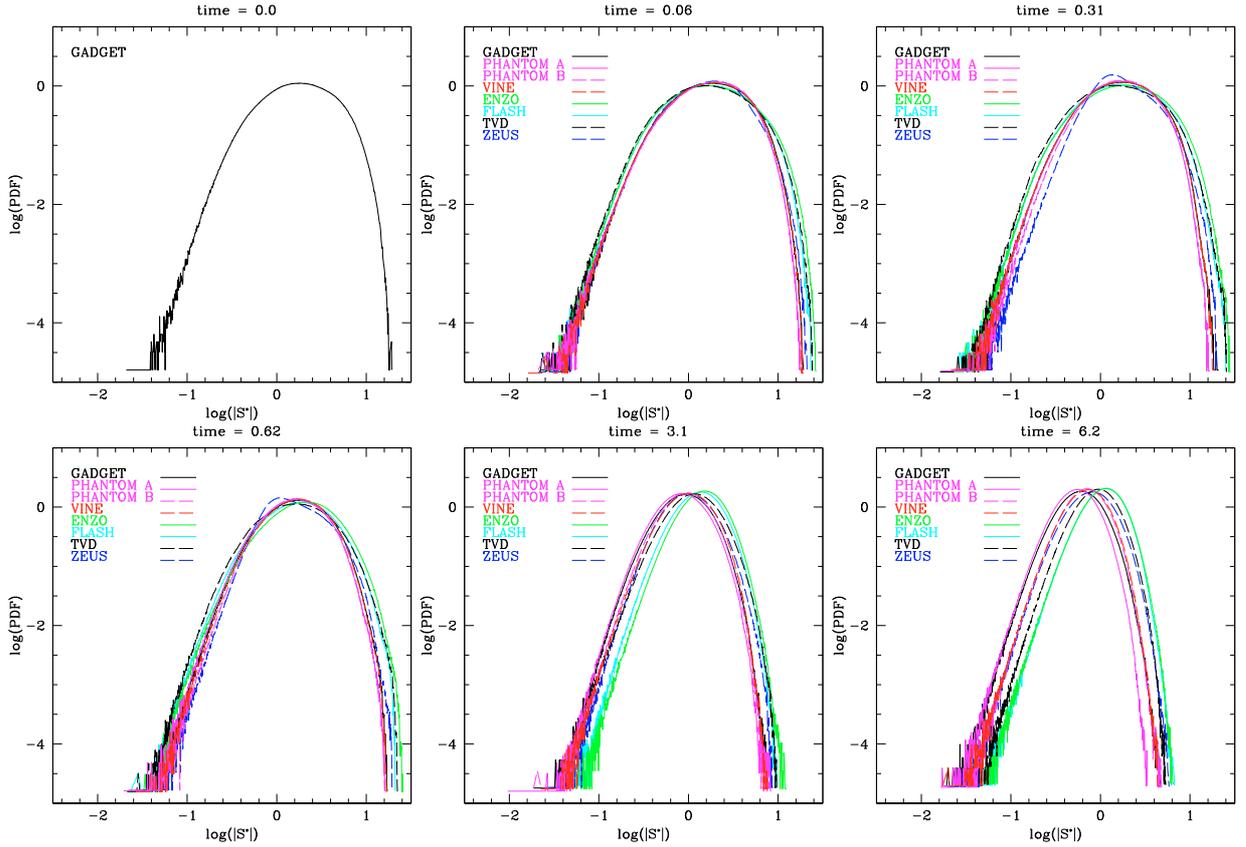


Fig. 12. Same as Fig. 10, but the PDFs of the trace-free rate of strain are shown, as defined in Eq. (13).

the star formation efficiency (Elmegreen 2008), and the Kennicutt-Schmidt relation on galactic scales (Elmegreen 2002; Kravtsov 2003; Tassis 2007).

5.5.2. Probability distribution functions of the rate of strain

Figure 12 presents the trace free rate of strain PDFs computed as explained in Sect. 3.4. Each panel shows the comparison of the PDFs of all codes at $t = 0.0, 0.06, 0.31, 0.62, 3.1,$ and $6.2 t_{\text{cross}}$. Within the first t_{cross} (top row and bottom-left panel of Fig. 12), all SPH codes and ZEUS have PDFs that are slightly narrower than those of the remaining grid codes (top-right and bottom-left panels of Fig. 12). At later times (bottom-middle and bottom-right panels of Fig. 12), all codes appear to have distributions with similar widths, but with the PDFs of the SPH codes and the ZEUS code peaking at almost half the $|S^*|$ value of the other grid codes. In particular, GADGET, PHANTOM A, PHANTOM B, VINE and ZEUS have their peaks at $\log |S^*| \lesssim 0$, while ENZO and FLASH have their peaks at $\log |S^*| \gtrsim 0$. TVD appears to peak in between. At $t = 6.2 t_{\text{cross}}$ the velocity power spectra of the codes are maintained up to k -values that can be ordered as follows (from higher to lower values): ENZO, FLASH, TVD, ZEUS, VINE, PHANTOM B, GADGET, PHANTOM A (cf. the bottom-right panels of Figs. 7 and 8 in Sect. 5.1). This is the order of the $|S^*|$ -value of the peaks of the rate of strain PDFs at this time (bottom-right panel of Fig. 12).

5.5.3. Probability distribution functions of the vorticity

In Fig. 13, we present the vorticity PDFs. The vorticity was computed as explained in Sect. 3.4. Each panel shows the comparison of the vorticity PDFs of all codes at $t = 0.0, 0.06, 0.31, 0.62,$

$3.1,$ and $6.2 t_{\text{cross}}$. The vorticity and the trace free rate of strain PDFs show a similar behaviour: within the first t_{cross} (top row and bottom-left panel of Fig. 13), the SPH codes and ZEUS show narrower distributions than the remaining grid codes. At later times (bottom-middle and bottom-right panels of Fig. 13), the vorticity distributions of the SPH codes and ZEUS peak at almost half the vorticity value of the other grid codes. Again, the peaks of PHANTOM B, VINE, TVD, and ZEUS are bracketed by the peaks of the remaining codes, with GADGET and PHANTOM A peaking at the lowest values of the vorticity, and ENZO and FLASH peaking at the highest vorticity.

Since the vorticity is related to the ability of the codes to model turbulent eddies, these comparisons indicate that SPH codes exhaust their ability earlier than grid codes, most likely because of their excess viscosity acting on the smallest of these eddies and erasing them. As in the case of the rate of strain above, the order of the vorticity values of the peaks of the vorticity PDFs is the same as the order with which the codes maintain their velocity power spectra in the high- k regime (cf. the bottom right panel of Fig. 7 and 8, and the bottom-right panel of Fig. 12).

5.5.4. Probability distribution functions of the velocity

Figure 14 shows the PDFs of the velocity component v_z . Each panel shows the comparison of the PDFs of all codes at times $t = 0.0, 0.06, 0.31, 0.62, 3.1, 6.2 t_{\text{cross}}$. For all these snapshots, the velocity PDFs are very similar for all codes. Apart from statistical fluctuations showing up in the wings of the distributions, the velocity PDFs are roughly Gaussian distributions. As the standard deviation of the density PDFs (cf. Fig. 10), the standard deviation of the velocity PDFs decreases with time, which simply reflects the turbulence decay (cf. Fig. 9).

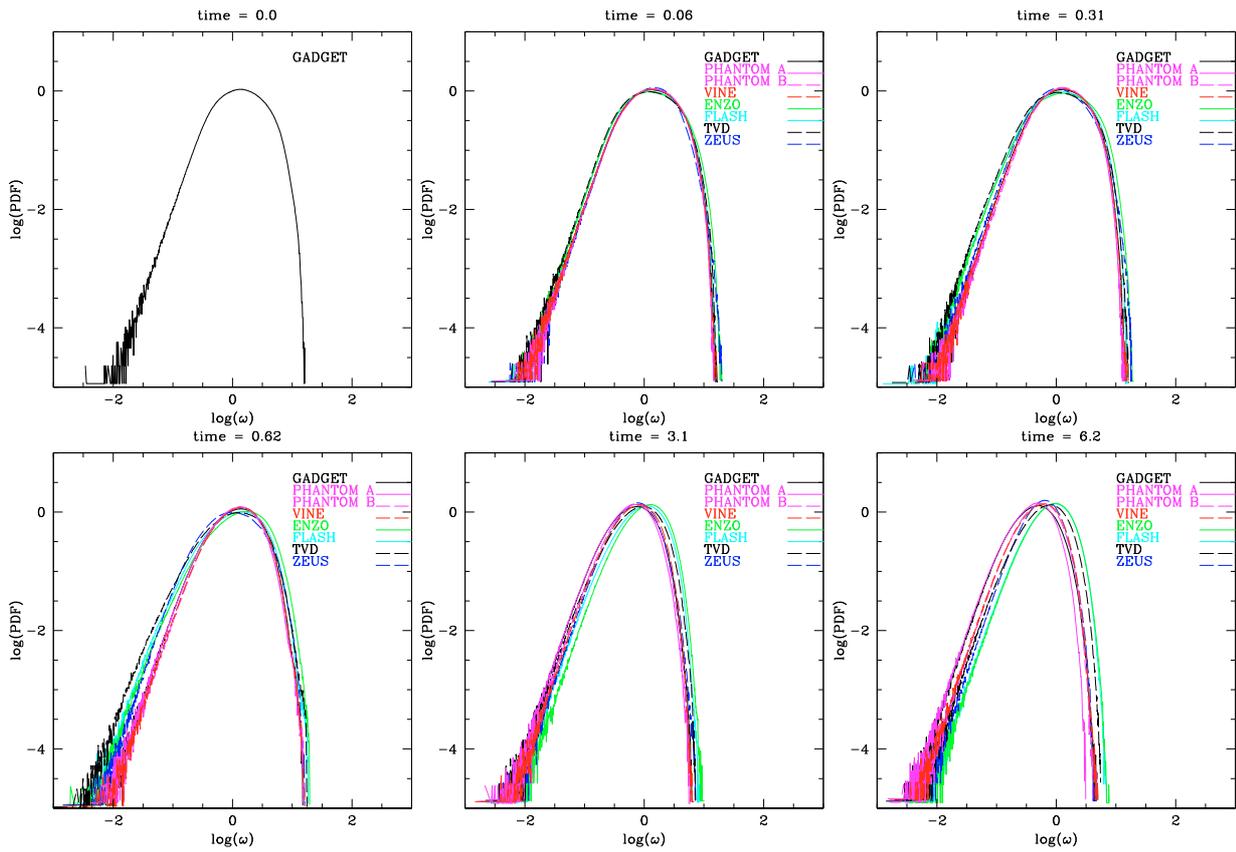


Fig. 13. Same as Fig. 10, but the PDFs of the vorticity are shown.

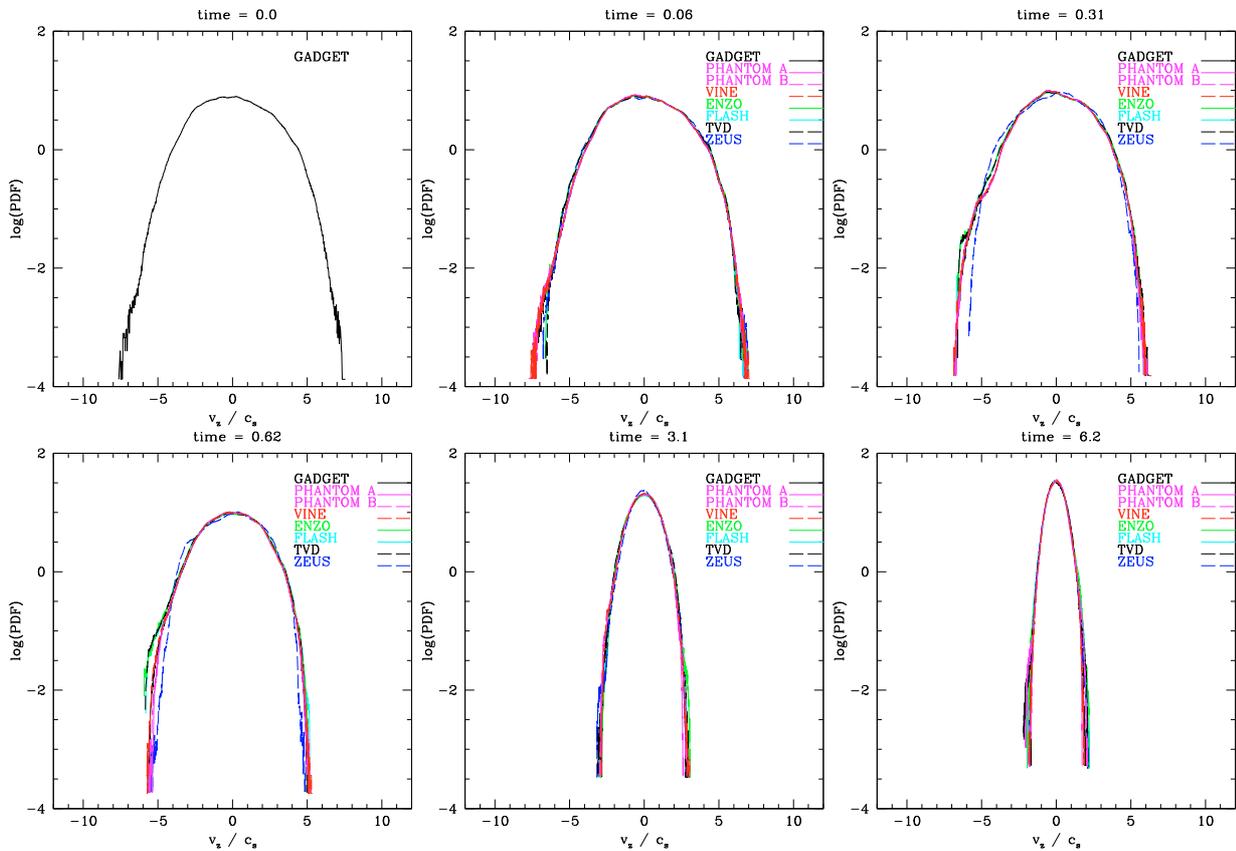


Fig. 14. Same as Fig. 10, but the PDFs of the z -component of the velocity (v_z) in units of the sound speed c_s are shown.

Table 5. Computational efficiency of all codes/runs.

	GADGET	PHANTOM A	PHANTOM B	VINE	ENZO	FLASH	TVD	ZEUS
architecture (CPU type)	Itanium	Clovertown	Clovertown	Itanium	Itanium	Xeon	Xeon	Xeon
CPU clock rate	1.6 GHz	2.66 GHz	2.66 GHz	1.3 GHz	1.6 GHz	1.6 GHz	2.66 GHz	2.33 GHz
number of CPUs used	32	8	8	64	8 (32)	64	8	64
parallelisation	MPI	OpenMP	OpenMP	OpenMP	MPI	MPI	MPI	MPI
total CPU-h	6,490	248	248	15,000	165 (203)	256	10	315
total CPU-h (norm. ^a)	10,960	697	697	20,600	165 (203)	256	17	459

^a CPU hours normalised to 256^3 resolution elements, and normalised to a clock rate of 1.6 GHz on an Intel Itanium/Xeon chip. These are very rough estimates that should only be accurate to within factors of a few, because of the different parallelisation hardware used on the various supercomputing platforms. Note however that GADGET and ENZO were run on the same supercomputing platform, and with the best run time and parameter optimisations that we could achieve on that supercomputer for both codes. The total CPU time for these two runs can thus be compared directly. ENZO was also run on 32 CPUs (values given in brackets).

6. Computational efficiency of the codes

Table 5 provides a summary of the computational efficiency of our codes/runs for the present setup. We remind the reader that in the current study all SPH codes used 215^3 resolution elements, while the grid codes used 256^3 resolution elements. We compensated for this discrepancy in the number of resolution elements, as well as compensated roughly for the different CPU clock rates in the last row of Table 5. However, the various runs were performed on different parallel machines throughout the world, some with, others without optimisations. Furthermore, different supercomputers use different hardware solutions for the parallelisation. Thus, the given numbers should only be taken as a rough estimate that may be accurate to within factors of a few. However, we emphasise that we have performed both the GADGET and the ENZO run on exactly the same supercomputing platform with the same optimisations, such that we can compare the performance of these two runs directly.

The fastest of all runs was performed with the TVD grid code. It is roughly ten times faster than our ENZO run, about 15 times faster than FLASH, and about 27 times faster than the version of ZEUS used here. However, the official ZEUS-MP version is expected to be faster and to scale better. Due to its specific design and implementation, PHANTOM is the fastest of all the SPH codes employed, but still roughly 50% slower than the slowest grid code (our version of ZEUS). GADGET and VINE are about 16 times, and about 30 times, respectively slower than PHANTOM. However, it must be remembered that the extra cost in SPH reflects the fact that resolution elements are placed to follow the mass, and thus preferentially to resolve high-density regions. Thus, additional information is calculated on small scales. This information however does not enter this comparison as the particles are interpolated onto a fixed grid. The extra cost for the SPH runs is similar to what can occur with grid-based Adaptive Mesh Refinement (AMR) (e.g. Berger & Colella 1989), because of the additional overhead to store and iterate over the AMR hierarchy (see e.g. Schmidt et al. 2009). However, a quantitative analysis of the performance of AMR versus SPH is beyond the scope of this paper, and should be discussed elsewhere.

The fact that GADGET and VINE are more than one order of magnitude slower than any of the grid codes makes it computationally expensive to study supersonic turbulence at resolutions higher than about 256^3 SPH particles. This may partly explain why no SPH calculations of supersonic turbulence have so far been using more than 512^3 particles. The latter was only

achieved with the PHANTOM code in Price & Federrath (2009) and in the KITP code comparison project⁶.

7. Conclusions

In this paper we report the comparison of the performance of four grid-based and three particle-based hydrodynamic codes on the modelling of supersonic turbulence decay. In particular, we have studied the decay of compressible, supersonic, isothermal turbulence in the absence of gravity within a periodic box using simulations with resolutions of 256^3 grid cells for the grid codes, and 215^3 particles for the SPH codes.

We have simulated driven turbulence with the SPH code GADGET. The SPH particle distribution at the end of the driving has been interpolated onto a grid that provides the initial conditions to the grid codes employed, namely ENZO, FLASH, TVD and ZEUS. We have also followed the decay of turbulence using several implementations of the SPH method, namely GADGET, PHANTOM (runs: A, B), and VINE, where PHANTOM B used the Morris & Monaghan (1997) viscosity switch, while PHANTOM A was run without the switch.

The turbulence decay was followed for about six turbulent crossing times. During the whole decay phase considered here, the turbulent flow stays in the supersonic regime. The turbulent energy dissipation was measured for all codes. For times greater than about 3 crossing times (when any initial transient phase due to small variations in the initial conditions has disappeared, and all codes have developed their individual dissipation signatures) a comparison of volume- and density-weighted velocity spectra indicates that the numerical viscosity of grid codes is generally lower than that of SPH codes, with details of the method like the order of the code contributing secondarily. We show that the differences between ENZO and FLASH are due to our choice of using different PPM diffusion parameters for the two codes, i.e., in this study ENZO was used with a PPM diffusion parameter of $K = 0.0$, while FLASH was used with the PPM diffusion parameter set to $K = 0.1$ as recommended by Colella & Woodward (1984). Switching-off PPM diffusion completely (as in our ENZO run) results in less dissipation, but produces a stronger bottleneck effect (see also Kritsuk et al. 2007; Federrath et al. 2009a). Use of the Morris & Monaghan (1997) viscosity implementation for SPH provides less dissipation as observed in our PHANTOM B and VINE runs in comparison with the GADGET and PHANTOM A runs. In general, the viscosity acts differently for the different grid- and SPH-codes at wavenumbers $k \gtrsim 8$ at the resolutions

⁶ <http://kitpstarformation07.wikispaces.com/Star+Formation+Test+Problems>

studied here, shown by our analysis of velocity power spectra in Sect. 5.1. However, all codes produced velocity spectra that are in good agreement for $k \lesssim 8$.

Using Fourier spectra we also showed that the additional information that the SPH method can offer in high-density regions and/or on small scales will be suppressed if it is not interpolated onto a high enough resolution grid, as discussed in Sects. 4.2.3 and 5.1.

The trace-free rate of strain and the vorticity PDFs confirm the ordering of the runs according to their dissipation given above. The density PDFs are very similar for all the runs performed in the present study. The means and standard deviations of the logarithmic density varied by less than 10% for all codes at all times analysed, with one exception (the mean logarithmic density obtained in our ZEUS run at $t = 0.62 t_{\text{cross}}$ varied by about 30%, which is also seen in its density PDF). For the SPH code GADGET we have shown that the density PDF obtained from the SPH particle distribution samples the high density tail slightly better, while our results indicate that using a grid, the low-density tail is slightly better sampled. However, the overall shape is very close to a log-normal distribution in density, and its mean and standard deviation are quite robust for all codes employed in the present study.

Our results demonstrate that different codes have different dissipation mechanisms affecting spatial scales $k \gtrsim N/32$. However, our code comparison also shows that SPH and grid codes give similar results for an equivalent number of resolution elements N for each direction in space on scales $k \lesssim N/32$, though with the SPH runs being about ten times more computationally expensive than the grid runs on average. Careful choice of numerical algorithm can extend this scaling range slightly, indicating that grid codes tend to show a slightly longer scaling range than SPH codes (cf. Figs. 7 and 8). However, at the numerical resolutions employed in the present study, all slopes inferred from the volume- and density-weighted spectra are too steep compared with higher resolution data from the literature. It is thus rather a question of resolution than a question of the specific properties of the hydrodynamical codes used in this study that determines their performance in reproducing turbulence scaling relations. This must be tested in a future comparison employing at least one order of magnitude more resolution elements for both grid and SPH codes.

Acknowledgements. The authors would like to thank Jens Niemeyer for many interesting and stimulating discussions, especially during the setup phase of this project. We thank the anonymous referee for a balanced and detailed report. S.K. kindly acknowledges support by an EU Commission “Marie Curie Intra-European (Individual) Fellowship” of the 6th Framework Programme with contract number MEIF-CT-2004-011226. S.K. also acknowledges financial assistance by the EU Commission Research Training Network “Constellation” of the 6th Framework Programme. C.F. acknowledges financial support from the International Max Planck Research School for Astronomy and Cosmic Physics (IMPRS-A) and the Heidelberg Graduate School of Fundamental Physics (HGSFP). The HGSFP is funded by the Excellence Initiative of the German Research Foundation DFG GSC 129/1. The ENZO and GADGET simulations used computational resources from the HLRBII project grant h0972 and pr32lo at the Leibniz Rechenzentrum Garching and from a project grant by CASPUR, Italy. C.F. and R.S.K. are grateful for subsidies from the DFG SFB 439 Galaxies in the Early Universe. R.S.K. and C.F. acknowledge financial support from the German Bundesministerium für Bildung und Forschung via the ASTRONET project STAR FORMAT (grant 05A09VHA) and from the Deutsche Forschungsgemeinschaft (DFG) under grants no. KL 1358/1, KL 1358/4, KL 1359/5. R.S.K. furthermore thank for subsidies from a Frontier grant of Heidelberg University sponsored by the German Excellence Initiative and for support from the Landesstiftung Baden-Württemberg via their programme International Collaboration II. D.J.P. is supported by a Royal Society University Research Fellowship (UK). The PDF figures were partly produced using SPLASH (Price 2007). MG and SW acknowledge support by the DFG

cluster of excellence “Origin and Structure of the Universe”. All VINE and part of the FLASH calculations were performed on an SGI Altix 3700 Bx2 supercomputer that was funded by the DFG cluster of excellence “Origin and Structure of the Universe”. The simulations performed by J.K. utilized a high performance cluster that was built with funding from the Korea Astronomy and Space Science Institute (KASI) and the Korea Science and Engineering Foundation through the Astrophysical Research Center for the Structure and Evolution of Cosmos (ARCSEC). This work of J.K. was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2009-0062863). AKJ acknowledges support by the Human Resources and Mobility Programme of the European Community under the contract MEIF-CT-2006-039569. M.M.M.L. acknowledges partial support for his work from NASA Origins of Solar Systems grant NNX07AI74G. The software used in this work was in part developed by the DOE-supported ASC/Alliance Center for Astrophysical Thermonuclear Flashes at the University of Chicago.

References

- Attwood, R. E., Goodwin, S. P., & Whitworth, A. P. 2007, *A&A*, 464, 447
 Ballesteros-Paredes, J., Gazol, A., Kim, J., et al. 2006, *ApJ*, 637, 384
 Ballesteros-Paredes, J., Klessen, R. S., Mac Low, M.-M., et al. 2007, in *Protostars and Planets V*, ed. B. Reipurth, D. Jewitt, & K. Keil, 63
 Beetz, C., Schwarz, C., Dreher, J., et al. 2008, *Phys. Lett. A*, 372, 3037
 Benzi, R., Biferale, L., Fisher, R. T., et al. 2008, *Phys. Rev. Lett.*, 100, 234503
 Berger, M. J., & Colella, P. 1989, *J. Comput. Phys.*, 82, 64
 Blitz, L., Fukui, Y., Kawamura, A., et al. 2007, in *Protostars and Planets V*, ed. B. Reipurth, D. Jewitt, & K. Keil, 81
 Boldyrev, S. 2002, *ApJ*, 569, 841
 Boldyrev, S., Nordlund, Å., & Padoan, P. 2002a, *ApJ*, 573, 678
 Boldyrev, S., Nordlund, Å., & Padoan, P. 2002b, *Phys. Rev. Lett.*, 89, 031102
 Bonazzola, S., Perault, M., Puget, J. L., et al. 1992, *J. Fluid Mechanics*, 245, 1
 Calder, A. C., Fryxell, B., Plewa, T., et al. 2002, *ApJS*, 143, 201
 Chandrasekhar, S. 1949, *ApJ*, 110, 329
 Chandrasekhar, S. 1951a, *R. Soc. London Proc. Ser. A*, 210, 18
 Chandrasekhar, S. 1951b, *R. Soc. London Proc. Ser. A*, 210, 26
 Colella, P., & Woodward, P. R. 1984, *J. Comput. Phys.*, 54, 174
 Commerçon, B., Hennebelle, P., Audit, E., Chabrier, G., & Teyssier, R. 2008, *A&A*, 482, 371
 de Avillez, M. A., & Mac Low, M.-M. 2002, *ApJ*, 581, 1047
 Dimonte, G., Youngs, D. L., Dimitis, A., et al. 2004, *Phys. Fluids*, 16, 1668
 Dobler, W., Haugen, N. E. L., Yousef, T. A., et al. 2003, *Phys. Rev. E*, 68, 026304
 Dubey, A., Fisher, R., Graziani, C., et al. 2008, in *Numerical Modeling of Space Plasma Flows*, ed. N. V. Pogorelov, E. Audit, & G. P. Zank, *ASP Conf. Ser.*, 385, 145
 Elmegreen, B. G. 2002, *ApJ*, 577, 206
 Elmegreen, B. G. 2008, *ApJ*, 672, 1006
 Elmegreen, B. G., & Scalo, J. 2004, *A&A*, 42, 211
 Federrath, C., Glover, S. C. O., Klessen, R. S., et al. 2008a, *Phys. Scr. T*, 132, 014025
 Federrath, C., Klessen, R. S., & Schmidt, W. 2008b, *ApJ*, 688, L79
 Federrath, C., Duval, J., Klessen, R., Schmidt, W., & Mac Low, M.-M. 2009a, *A&A*, submitted [arXiv:0905.1060]
 Federrath, C., Klessen, R. S., & Schmidt, W. 2009b, *ApJ*, 692, 364
 Frisch, U. 1995, *Turbulence* (Cambridge Univ. Press)
 Fryxell, B., Olson, K., Ricker, P., et al. 2000, *ApJS*, 131, 273
 Gingold, R. A., & Monaghan, J. J. 1977, *MNRAS*, 181, 375
 Glover, S. C. O., & Mac Low, M.-M. 2007, *ApJ*, 659, 1317
 Glover, S. C. O., Federrath, C., Mac Low, M.-M., et al. 2009, *MNRAS*, in press
 Gritschneider, M., Naab, T., Walch, S., Burkert, A., & Heitsch, F. 2009, *ApJ*, 694, L26
 Haugen, N. E., & Brandenburg, A. 2004, *Phys. Rev. E*, 70, 026405
 Hayes, J. C., Norman, M. L., Fiedler, R. A., et al. 2006, *ApJS*, 165, 188
 Heitmann, K., Ricker, P. M., Warren, M. S., et al. 2005, *ApJS*, 160, 28
 Heitsch, F., Mac Low, M.-M., & Klessen, R. S. 2001, *ApJ*, 547, 280
 Hennebelle, P., & Chabrier, G. 2008, *ApJ*, 684, 395
 Hennebelle, P., & Chabrier, G. 2009, *ApJ*, 702, 1428
 Isichenko, M. B. 1992, *Rev. Mod. Phys.*, 64, 961
 Jappsen, A.-K., Klessen, R. S., Larson, R. B., Li, Y., & Mac Low, M.-M. 2005, *A&A*, 435, 611
 Kim, J., Ryu, D., Jones, T. W., et al. 1999, *ApJ*, 514, 506
 Kitsionas, S., & Whitworth, A. P. 2002, *MNRAS*, 330, 129
 Klein, R. I., Inutsuka, S.-I., Padoan, P., et al. 2007, in *Protostars and Planets V*, ed. B. Reipurth, D. Jewitt, & K. Keil, 99
 Klessen, R. S. 2000a, *ApJ*, 535, 869
 Klessen, R. S., & Lin, D. N. 2003, *Phys. Rev. E*, 67, 046311
 Klessen, R. S., Heitsch, F., & Mac Low, M.-M. 2000b, *ApJ*, 535, 887

- Kolmogorov, A. N. 1941, Dokl. Akad. Nauk SSSR, 32, 16
- Kowal, G., & Lazarian, A. 2007, ApJ, 666, L69
- Kravtsov, A. V. 2003, ApJ, 590, L1
- Kritsuk, A. G., Norman, M. L., & Padoan, P. 2006, ApJ, 638, L25
- Kritsuk, A. G., Norman, M. L., Padoan, P., et al. 2007, ApJ, 665, 416
- Krumholz, M. R., & McKee, C. F. 2005, ApJ, 630, 250
- Lemaster, M. N., & Stone, J. M. 2008, ApJ, 682, L97
- Lemaster, M. N., & Stone, J. M. 2009, ApJ, 691, 1092
- Lesieur, M. 1997, Turbulence in Fluids (Kluwer)
- Li, Y., Klessen, R. S., & Mac Low, M.-M. 2003, ApJ, 592, 975
- Lucy, L. B. 1977, AJ, 82, 1013
- Mac Low, M.-M. 1999, ApJ, 524, 169
- Mac Low, M.-M., & Klessen, R. S. 2004, Rev. Mod. Phys., 76, 125
- Mac Low, M.-M., Klessen, R. S., Burkert, A., et al. 1998, Phys. Rev. Lett., 80, 2754
- McKee, C. F., & Ostriker, E. C. 2007, ARA&A, 45, 565
- Metzler, R., & Klafter, J. 2000, Phys. Rep., 339, 1
- Monaghan, J. J. 1988, Comp. Phys. Commun., 48, 89
- Monaghan, J. J. 1992, ARA&A, 30, 543
- Monaghan, J. J. 1997, J. Comput. Phys., 136, 298
- Monaghan, J. J. 2005, Rep. Progr. Phys., 68, 1703
- Monaghan, J. J., & Gingold, R. A. 1983, J. Comput. Phys., 52, 374
- Monaghan, J. J., & Lattanzio, J. C. 1985, A&A, 149, 135
- Morris, J., & Monaghan, J. 1997, J. Comput. Phys., 136, 41
- Nelson, A. F., Wetzstein, M., & Naab, T. 2009, ApJS, 184, 326
- Nordlund, Å., & Padoan, P. 1999, in Interstellar Turbulence, ed. J. Franco, & A. Carraminana, 218
- Norman, M. L., & Bryan, G. L. 1999, in Numerical Astrophysics, ed. S. M. Miyama, K. Tomisaka, & T. Hanawa, Astrophys. Space Sci. Libr., 240, 19
- O'Shea, B. W., Bryan, G., Bordner, J., et al. 2004 [arXiv:0403044]
- Ostriker, E. C., Gammie, C. F., & Stone, J. M. 1999, ApJ, 513, 259
- Ostriker, E. C., Stone, J. M., & Gammie, C. F. 2001, ApJ, 546, 980
- Padoan, P., & Nordlund, Å. 2002, ApJ, 576, 870
- Padoan, P., & Nordlund, A. 2009, ApJ, submitted [arXiv:0907.0248]
- Padoan, P., Nordlund, Å., & Jones, B. J. T. 1997, MNRAS, 288, 145
- Padoan, P., Jimenez, R., Nordlund, Å., et al. 2004, Phys. Rev. Lett., 92, 191102
- Padoan, P., Nordlund, Å., Kritsuk, A. G., Norman, M. L., & Li, P. S. 2007, ApJ, 661, 972
- Panis, J.-F., & Péroullet, M. 1998, Phys. Fluids, 10, 3111
- Porter, D. H., & Woodward, P. R. 1994, ApJS, 93, 309
- Price, D. J. 2004a, Ph.D. Thesis (University of Cambridge)
- Price, D. J., & Monaghan, J. J. 2004b, MNRAS, 348, 139
- Price, D. J. 2007a, Publ. Astron. Soc. Aust., 24, 159
- Price, D. J., & Monaghan, J. J. 2007b, MNRAS, 374, 1347
- Price, D. J., & Federrath, C. 2009, MNRAS, submitted
- Sasao, T. 1973, PASJ, 25, 1
- Scalo, J., & Elmegreen, B. G. 2004, ARA&A, 42, 275
- Schmidt, W., Hillebrandt, W., & Niemeyer, J. C. 2006a, Computers and Fluids, 35, 353
- Schmidt, W., Niemeyer, J. C., & Hillebrandt, W. 2006b, A&A, 450, 265
- Schmidt, W., Niemeyer, J. C., Hillebrandt, W., et al. 2006c, A&A, 450, 283
- Schmidt, W., Federrath, C., & Klessen, R. 2008, Phys. Rev. Lett., 101, 194505
- Schmidt, W., Federrath, C., Hupp, M., Kern, S., & Niemeyer, J. C. 2009, A&A, 494, 127
- She, Z.-S., & Leveque, E. 1994, Phys. Rev. Lett., 72, 336
- Springel, V., Yoshida, N., & White, S. D. M. 2001, New Astron., 6, 79
- Stone, J. M., & Norman, M. L. 1992a, ApJS, 80, 753
- Stone, J. M., & Norman, M. L. 1992b, ApJS, 80, 791
- Stone, J. M., Ostriker, E. C., & Gammie, C. F. 1998, ApJ, 508, L99
- Strang, G. 1968, SIAM J. Numer. Anal., 5, 506
- Tasker, E. J., Brunino, R., Mitchell, N. L., et al. 2008, MNRAS, 390, 1267
- Tassis, K. 2007, MNRAS, 382, 1317
- Toro, E. F. 1997, Riemann solvers and numerical methods for fluid dynamics (Springer)
- van Leer, B. 1977, J. Comput. Phys., 23, 276
- Vázquez-Semadeni, E. 1994, ApJ, 423, 681
- Vázquez-Semadeni, E., Ballesteros-Paredes, J., & Klessen, R. S. 2003, ApJ, 585, L131
- Vernaleo, J. C., & Reynolds, C. S. 2006, ApJ, 645, 83
- von Weizsäcker, C. F. V. 1943, ZAp, 22, 319
- von Weizsäcker, C. F. 1951, ApJ, 114, 165
- Wetzstein, M., Nelson, A. F., Naab, T., et al. 2009, ApJS, 184, 298