

Visualizing the world's largest turbulence simulation

Salvatore Cielo^a, Luigi Iapichino^a, Johannes Günther^b, Christoph Federrath^c, Elisabeth Mayer^a, Markus Wiedemann^a

^aLeibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities (LRZ), Garching b. München, Germany

^bIntel Corporation, Feldkirchen, Germany

^cResearch School of Astronomy and Astrophysics, Australian National University, Canberra, ACT 2611, Australia

Abstract

In this exploratory submission we present the visualization of the largest interstellar turbulence simulations ever performed, unravelling key astrophysical processes concerning the formation of stars and the relative role of magnetic fields. The simulations, including pure hydrodynamical (HD) and magneto-hydrodynamical (MHD) runs, up to a size of 10048^3 grid elements, were produced on the supercomputers of the Leibniz Supercomputing Centre and visualized using the hybrid parallel (MPI + TBB) ray-tracing engine OSPRay associated with VisIt. Besides revealing features of turbulence with an unprecedented resolution, the visualizations brilliantly showcase the stretching-and-folding mechanisms through which astrophysical processes such as supernova explosions drive turbulence and amplify the magnetic field in the interstellar gas, and how the first structures, the *seeds* of newborn stars are shaped by this process.

Keywords: scientific visualization, high performance computing

1. The link between turbulence and star formation in the Universe

Turbulent flows are a key feature for characterizing the dynamics of physical systems in several domains, including weather modelling, industry and astrophysics. While turbulence gives rise to long-term unpredictable systems, its effects have been characterized in great detail. However, for quantitative predictions, statistical and numerical methods are essential tools.

Turbulence is ubiquitous in astrophysical fluids; just to name a few examples, it plays an important role in accelerating cosmic ray particles and amplifying magnetic fields in clusters of galaxies, and in the flame propagation mechanism of thermonuclear supernovae. Coming to the main topic of this work, turbulence is an essential ingredient for modeling the physics of the interstellar medium (ISM), the astrophysical site where stars are formed.

Numerical simulations of turbulent fluids in the ISM provide insight into the statistical properties of the flow and are necessary for a quantitative characterization of star formation. While in principle these simulations contain nothing but fundamental physics and hydrodynamics (HD), the multi-scale distribution of energy, the diffuse high-velocity regions and the chaotic flows of these computations make domain-optimization techniques (such as the use of adaptive meshes, or grids different than Cartesian) rather ineffective. Thus, such simulations have to rely purely on numerical and solver optimizations, and on an efficient parallelization scheme for running on large, modern supercomputing architectures.

Turbulent models of the ISM also show a rich pool of emergent physical effects [1]. The qualitative picture presented by observations with modern telescopes such as the Atacama

Large Millimetre/Sub-millimetre Array (ALMA) in the Chilean desert is that turbulence is driven on large scales, e.g. by shocks and feedback from nearby or newly born stars. These stirring mechanisms have an episodic character, yet the resulting supersonic turbulent motions are actively sustained for time scales longer than the typical interval between driving events. The kinetic energy of the turbulent motions is then transferred to smaller length scales in the cascade process, until turbulence is dissipated by viscous effects on small scales. In this process, at a typical length called *sonic scale* a transition from supersonic to subsonic turbulence occurs, marking a corresponding change of the statistical properties of the turbulent energy spectrum.

Thus, a simulation capable of resolving this all-important *sonic scale* is a necessary first step in understanding the structure of star-forming regions. Studying this problem is computationally very challenging because it requires enormous spatial resolution, resulting in a sufficiently large dynamical range.

Finally, most observed star-forming regions have clear indications of the presence of magnetic fields. This additional physical ingredient changes the spectrum of turbulence both quantitatively (shifting the energy distribution) and qualitatively (giving rise to possible self-amplifying dynamo effects). Modern codes can deal effectively with magnetic fields in the magneto-hydrodynamics (MHD) framework, but this roughly doubles the computational requirements of already highly demanding simulations.

1.1. The largest turbulence simulations to date

We present a visualization of the largest sets of HD and MHD turbulence simulations ever realized, with up to 10048^3 resolution elements in a static grid (see Figure 1, based on a smaller test run). This unprecedented resolution allows for a dynamic

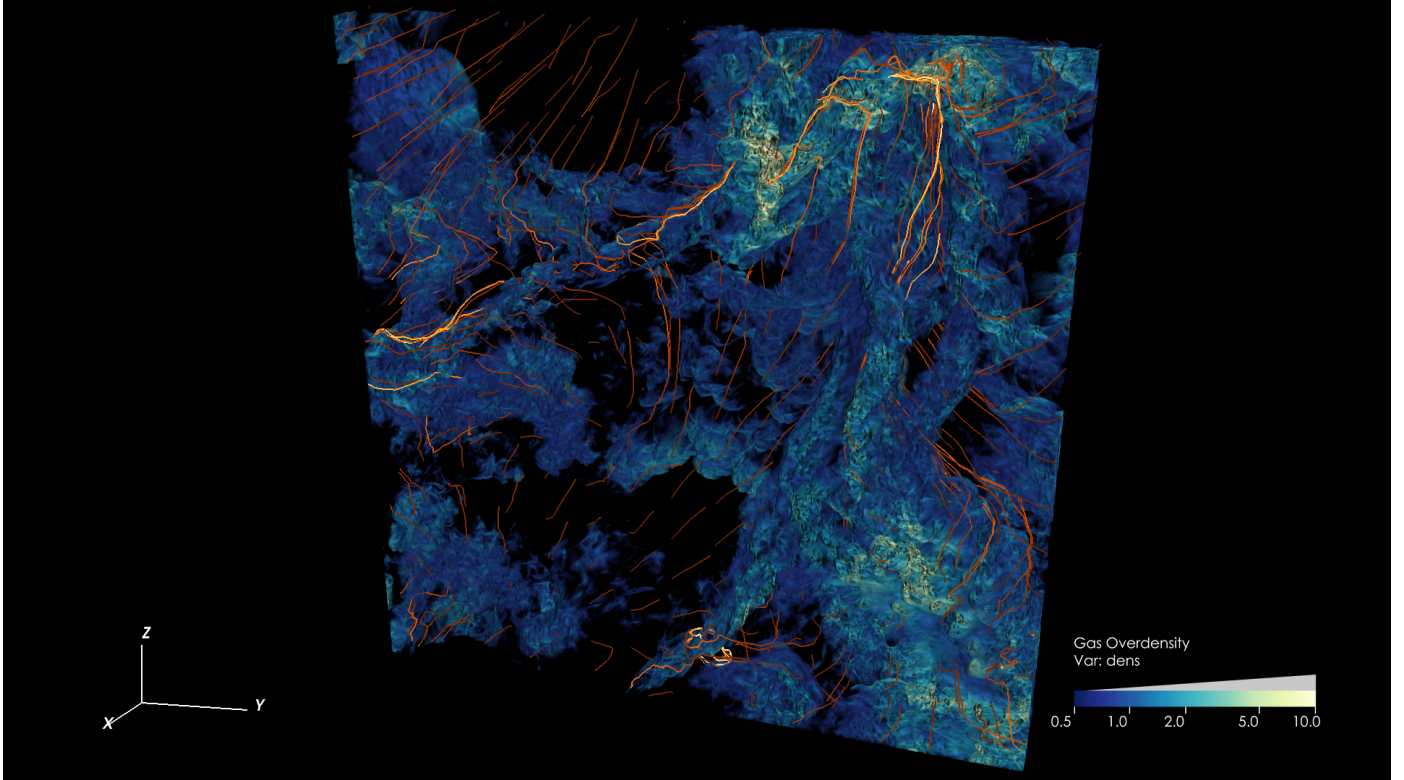


Figure 1: Sample ray-tracing plus velocity streamlines rendering of the 1152^3 HD simulation, using VisIt and OSPRay on SuperMUC-NG. The displayed data correspond to a slab with a volume of 10% of the full data cube.

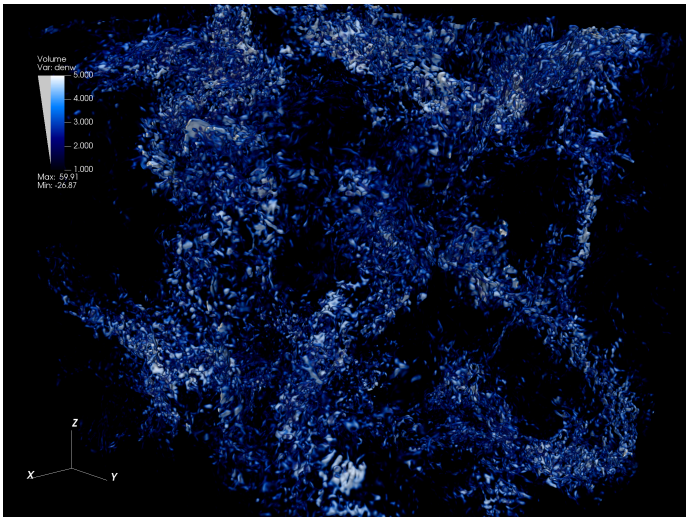


Figure 2: Volume rendering of the gas density structures associated with the sonic scale (i.e. regions with the gas velocity close to the sound speed) in the HD simulation with grid resolution of 1152^3 .

range of four orders of magnitude in length scale. The simulations were produced with the FLASH code [2], a public grid-based MHD code. In order to model the injection of supersonic turbulence, FLASH uses a stochastic, Fourier-based driving method (see e.g. [3]).

The detailed results of the HD simulation were presented in [4]; see also [5] for a technical overview. A quantitative char-

acterization of the gas Mach number as a function of spatial scale (the so-called *structure function*) has revealed excellent agreement with theoretical models both in the supersonic and subsonic regime. For the first time, the sonic scale is resolved as the transition between them (Figure 2). This in turns allows us to infer the width distribution of filamentary structures in star-forming regions and ultimately the critical density for the formation of stars.

As for the MHD simulation, besides technical works describing the code performance [6], the present submission displays its outputs for the first time. Its scientific results will be presented in forthcoming publications.

2. Computational details

2.1. Code optimization and performance

The version of FLASH used in this work, based on the public release v. 4, has been optimized to introduce a hybrid single/double precision scheme ([5]) that, while preserving the same accuracy of the double-precision case, allows for a $3.6\times$ speedup and a factor 4.1 lower memory consumption. These numbers refer to the MHD simulations, run on the *SuperMUC-NG* supercomputer at LRZ¹, whose architecture is based on Intel® Xeon® Platinum 8174 Processors (code-named Skylake). FLASH is capable of scaling up to full machine size

¹<https://doku.lrz.de/display/PUBLIC/SuperMUC-NG>

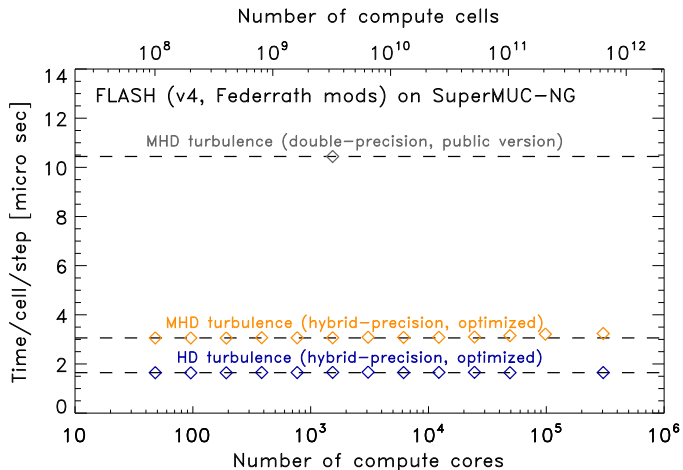


Figure 3: Weak scaling test of FLASH on SuperMUC-NG. Gray, orange and blue symbols correspond to MHD simulations run with the public code version, and MHD and HD simulations run with the hybrid-precision code version, respectively. The horizontal dashed lines mark the ideal scaling in all three cases. The scaling remains optimal up to full machine size for the runs with the optimized code. MHD introduces a performance overhead with respect to HD, but the code largely benefits from the hybrid-precision scheme.

(6336 nodes, corresponding to 304,128 compute cores; see Figure 3). The realization of such large simulations presents severe technical issues, both on the computation and data storage viewpoint: the 10048^3 HD simulations alone required about 131 TB memory, over 23 TB of disk space per snapshot (with more than 100 snapshots produced during the simulation campaign), and 45 millions CPU-h of computing time.

2.2. Visualization: VisIt and OSPRay

Visualizing such large data sets comes with enormous challenges in their computation. Volume renderings of scientific data is traditionally based on the use of hardware accelerators. On the other hand, the recent generations of multi-core processors, such as the Skylake nodes which SuperMUC-NG is based on, expose a high degree of parallelism (number of cores per node and large vector registers) which would be beneficial to exploit for visualization purposes. Ray-tracing methods are particularly well-suited to run on such architectures, as they present excellent parallel scalability [7, 8], and easy access to high-quality illumination aftereffects (e.g. shading and ambient occlusion). In addition, being able to perform visualizations (as well as post-processing) on the same machine where the simulations were run presents obvious advantages, in terms of workflow, data management and hardware usage.

With these considerations in mind, we have built and deployed a custom version of the VisIt analysis and visualization software for SuperMUC-NG. This version integrates the Intel OSPRay rendering engine[9]², embodying the *software-defined visualization* concept, optimized for CPU usage without the need of accelerators. OSPRay uses Threading Building Blocks (TBB) for parallel work sharing, and integrates additional features (e.g. the aforementioned shading and ambient

occlusion, or transfer functions pre-integration) which are absent in the standard version of VisIt [7].

Our customized version including OSPRay is based on the recent public VisIt v3.0.0. Considerable work on the source code, not included in the public release, was however necessary for granting OSPRay full communication across the compute nodes via MPI. This build shows better performance than the standard public version of VisIt used for reference (v. 2.13.2), based on the Mesa GL and parallelized with a pure MPI scheme.

In Figure 4 we show the scaling behaviour of our visualization strategy. In the upper panel, the node-level scaling is presented, including two reference points from the most scalable and comparable algorithm (according to our previous tests) from the reference VisIt version, [Ray Casting: Compositing]:[Kernel-based] [7]. OSPRay (yellow line) reaches a speedup of almost 16 \times on the node. The same algorithm on the Intel Xeon E5-2697 v3 architecture (Haswell node from SuperMUC Phase 2, with 28 cores, red line) achieves a nearly comparable speed-up, though with lower absolute performance. The use of OSPRay improves the time to solution of the rendering on a node (comparison between the blue and yellow line) by 12 \times .

The bottom diagram shows the scaling for a typical task of our submission, the rendering of a single slab of what we call a *tomography* of the gas density, performed on the largest simulation snapshot, the HD with the resolution of 10048^3 grid cells. This task includes both the creation of a 3D scene and the proper scalable rendering phase³, but leaving out I/O timings. Satisfactory scaling is achieved up to a considerable fraction of the machine, flattening only when approaching a few thousand compute nodes. This feature of the scaling holds when using both one or four MPI tasks per node (blue and red line, respectively).

Given the large size of the visualized data, it is probably worth considering the hypothesis of rendering the volumetric data directly from memory as they are produced by FLASH, a technique commonly known as *on-the-fly* or *in-situ* visualization. The pros of this approach include a time resolution as high as the time stepping of the simulation itself, and the elimination of I/O times. The main cons are instead the non-interactive nature of the task, an overhead that limits the resources available for the simulation, and the necessity to interface simulation and visualization code. Although VisIt presents this interface capability, this was not a priority for our project. On one hand, the custom version of FLASH we used includes *on-the-fly* visualization routines, though currently limited to slices and projections; on the other, our main aim is to showcase the efficient visualisation of large amounts of simulation data from storage. Of course *on-the-fly* visualization presents interesting implications, and we may consider it for future projects.

As a final remark we point out a few additional and handy synergies of the software-defined visualization workflow. Firstly, it is possible to access VisIt-OSPRay either in interactive GUI-aided sessions, or in batch mode via the SLURM

²<https://www.ospray.org>

³i.e. the only task using TBB parallelization.

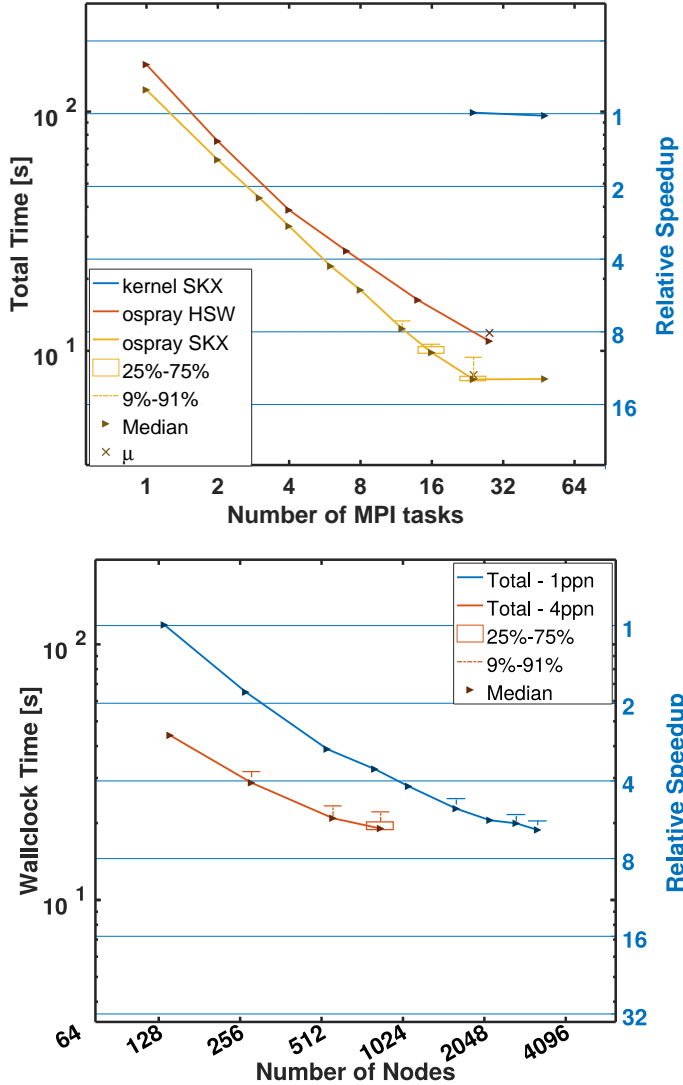


Figure 4: UPPER PANEL: VisIt-OSPRay node-level scaling behavior over MPI tasks (TBB not shown, but always at work). The yellow and red lines show the scaling for a SuperMUC-NG node (Skylake node) and, for comparison, for a Haswell node (SuperMUC Phase 2). Moreover, for the best-scaling algorithm of the standard VisIt version (kernel-based ray-casting, blue line), two scaling points are shown. LOWER PANEL: VisIt-OSPRay strong scaling for tasks of the presented visualization, using one (blue line) or four (red line) MPI tasks per node. The reported time to solution refers to a single snapshot of the *tomography* of the 10048^3 HD simulation, from 128 to 3076 nodes (i.e. one half of the whole SuperMUC-NG). Percentile confidence intervals after 20 measures are shown when significant.

queuing system and VisIt’s Python scripting, as it was done for the present case. The two methods can at will be conjugated, since the multi-purpose GUI allows us to configure hosts for remote usage, and even to submit batch jobs, a very handy feature on HPC systems. Volume plots produced in this way are also easily suited for Virtual Reality (VR) applications, since stereographic rendering can be easily configured. A VR version of our submission is in progress and will be showcased in the LRZ visualization facilities⁴. Finally, VisIt can also be used for

data post-processing other than visualization, through queries, histogram plots and other features that also benefit from MPI parallelization.

3. Movie description and scientific considerations

The whole visualization consists of raytracing, volumetric plots rendered with VisIt, using the OSPRay rendering engine. The only exception is the false-color velocity streamline plot (see below), because this specific kind of visualization is not yet supported by VisIt-OSPRay.

The visualization allows the viewer to draw some immediate qualitative conclusions. The main findings are summarized below.

3.1. Gas density of the 10048^3 HD run

After explaining how the sonic scale is interpreted, we begin the visualization with a *tomography* of the gas overdensity (i.e. the density fluctuations with respect to the mean density). This tomography consist in showing only the gas contained in a *slab*, whose thickness is 10% of the total box. The slab moves across half of the simulation box, at a fixed final time snapshot of the simulation. With this kind of visualization one can show the minute details of the extreme-resolution data set. This section of the movie has been rendered with 512 nodes on SuperMUC-NG.

The following addition of velocity streamlines (commented in the visualization, and colored according to their path length) required the usage of 1024 nodes, mainly due to the increased memory requirements, as the use of velocity requires reading an additional 3D vector data set. An example of these combined data sets is shown in Figure 1, for the 1152^3 output.

Next, the sonic-scale filtered density is shown. Since this analysis highlights the gas parcels with velocity magnitude close to the speed of sound, it can be considered as a proxy for the shock locations, showing how ubiquitous and interconnected they are across the full box. This task was the least expensive from a computational viewpoint, as the shown data set remains the same throughout this part of the visualization, and only the camera position rotates. It was rendered in two solutions using 128 to 512 nodes, for balancing queue- and runtime on the system. An example from the 1152^3 HD run, preferred for visual clarity, is reported in Figure 2, whereas in the movie the output of the full 10048^3 HD run is shown.

3.2. Time evolution: the 1008^3 MHD case

In a second series of renderings, we show the full time evolution of a MHD run, transitioning between the gas overdensity and the magnetic energy. For an evolution to have enough time coverage, a large number of data dumps have to be combined, and this was not feasible at full simulation resolution. In the MHD run with 1008^3 grid resolution we saved 557 data dumps, for a total disk storage of more than 15 TB.

The strength of this visualization is to showcase the stretching and folding mechanisms, leading to the subsequent development of hydrodynamical instabilities and to the turbulent dynamo. A visual inspection shows that the magnetic field has a

⁴https://www.lrz.de/services/v2c_en/

larger magnitude around the densest regions, forming filamentary structures in analogy to the ones in the density field (Figure 5). As for the rendering computation, while for the density a single node was sufficient, the magnetic field required four nodes.

4. Concluding remarks

With the increase of computing resources on the road to Exascale, it is getting more and more urgent to handle large data sets; the complexity of data analysis has become not smaller than the one of the simulations themselves. In our work we demonstrate that data visualization in the regime beyond 10 TB is a HPC task that can make efficient use of high-end computing systems. These visualizations, besides having a remarkable visual impact, are extremely important for exploring large data sets and for steering further quantitative analysis.

Acknowledgements

C. F. acknowledges funding provided by the Australian Research Council (Discovery Project DP170100603 and Future Fellowship FT180100495), and the Australia-Germany Joint Research Cooperation Scheme (UA-DAAD). We acknowledge high-performance computing resources provided by the Leibniz Supercomputing Centre and the Gauss Centre for Supercomputing (grants pr32lo, pr48pi and GCS Large-scale project 10391), the Australian National Computational Infrastructure (grant ek9) in the framework of the National Computational Merit Allocation Scheme and the ANU Allocation Scheme. The simulation software FLASH was in part developed by the DOE-supported Flash Center for Computational Science at the University of Chicago. VisIt is supported by the Department of Energy with funding from the Advanced Simulation and Computing Program and the Scientific Discovery through Advanced Computing Program.

References

- [1] C. Federrath, The turbulent formation of stars, *Physics Today* 71 (6) (2018) 38–42. [arXiv:1806.05312](#), [doi:10.1063/PT.3.3947](#).
- [2] B. Fryxell, K. Olson, P. Ricker, F. Timmes, M. Zingale, D. Lamb, P. MacNeice, R. Rosner, J. Truran, H. Tufo, FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes, *Astrophys. J. Supp.* 131 (2000) 273–334. [doi:10.1086/317361](#).
- [3] C. Federrath, J. Roman-Duval, R. S. Klessen, W. Schmidt, M. M. Mac Low, Comparing the statistics of interstellar turbulence in simulations and observations. Solenoidal versus compressive turbulence forcing, *A&A* 512 (2010) A81. [arXiv:0905.1060](#), [doi:10.1051/0004-6361/200912437](#).
- [4] C. Federrath, R. S. Klessen, L. Iapichino, J. R. Beattie, The sonic scale revealed by the worlds largest turbulence simulation, *Nature Astronomy*, submitted.
- [5] C. Federrath, R. S. Klessen, L. Iapichino, N. J. Hammer, The world's largest turbulence simulations, *arXiv e-prints* [arXiv:1607.00630](#).
- [6] S. Cielo, L. Iapichino, F. Baruffa, C. Federrath, M. Bugli, Are astrophysical codes on the road to Exascale? Experiences from code modernization on many-coresystems, *Future Generation of Computer Systems*, submitted.

- [7] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, P. Navrátil, VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data, in: *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, 2012, pp. 357–372.
- [8] I. Wald, G. Johnson, J. Amstutz, C. Brownlee, A. Knoll, J. Jeffers, J. Günther, P. Navrátil, OSPRay - A CPU ray tracing framework for scientific visualization, *IEEE Transactions on Visualization and Computer Graphics* 23 (1) (2017) 931–940. [doi:10.1109/TVCG.2016.2599041](#).
- [9] Q. Wu, W. Usher, S. Petruzza, S. Kumar, F. Wang, I. Wald, V. Pascucci, C. D. Hansen, VisIt-OSPRay: Toward an Exascale Volume Visualization System, in: H. Childs, F. Cucchietti (Eds.), *Eurographics Symposium on Parallel Graphics and Visualization*, The Eurographics Association, 2018. [doi:10.2312/pgv.20181091](#).

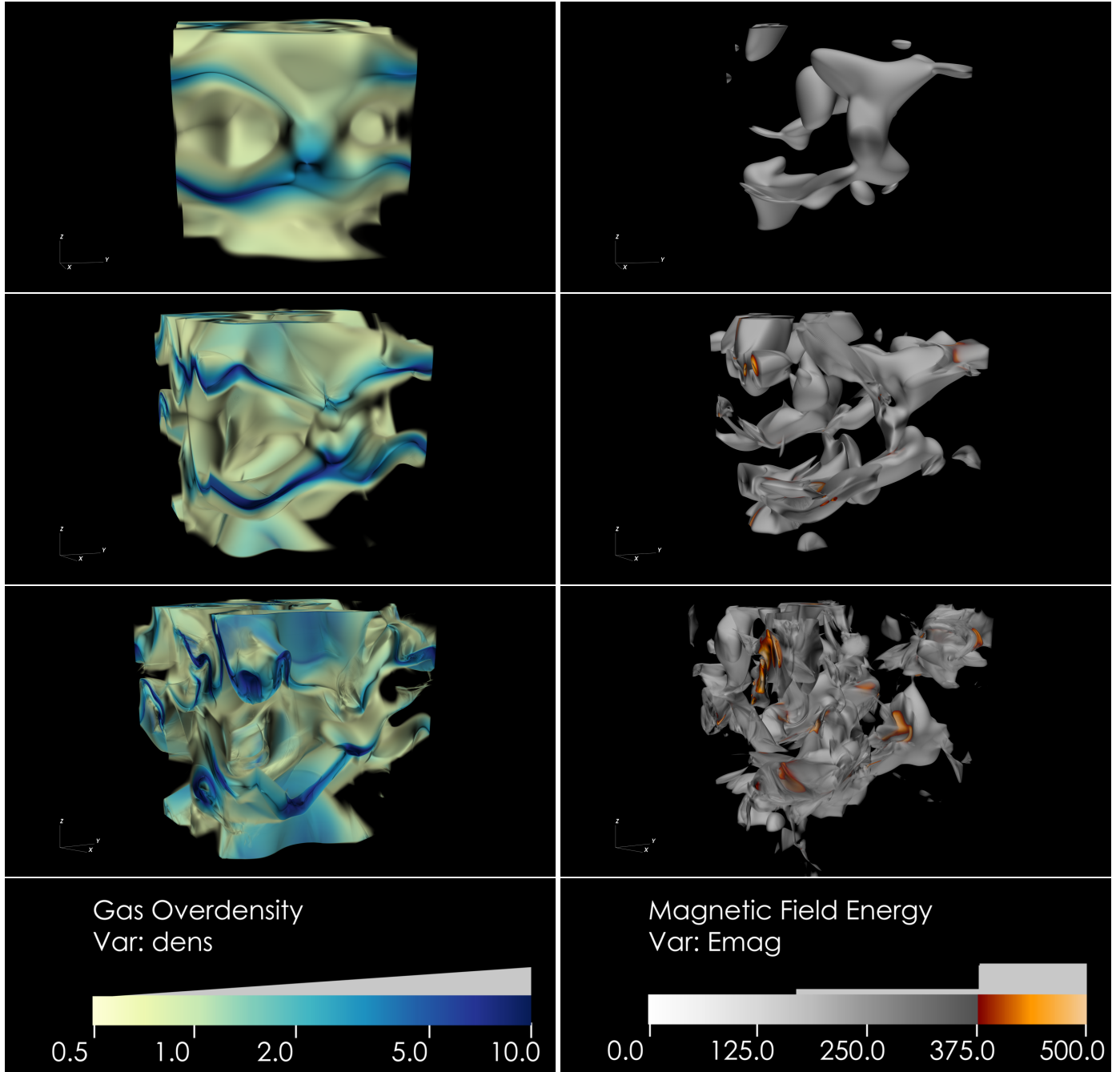


Figure 5: Density (left) and magnetic field energy (right) evolution in a 1008^3 MHD run. The gas folding and the instabilities giving rise to turbulence are visible. By comparing the two data fields, the level of magnetic energy of the densest structures can be inferred.