Universität zu Köln

MASTER THESIS

ASTROPHYSICS INSTITUT COLOGNE

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

submitted by

JOHANNES MARKERT

Köln - 15.03.2017

Contents

1 Introduction

Turbulent gases are everywhere and play a major role in nature, science and engineering. Consequently, the desire to model resp. simulate fluent media as close to reality as possible has driven many generations of scientists to develop better algorithms.

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

A numerical solving algorithm consists of three parts:

- Integration of the governing equations of fluid flow over the control volume

- Discretization of the integral into a system of algebraic equations

- iterative time stepping method

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

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

Owing to their finite element nature, the DG methods have the following main advantages over classical finite volume and finite difference methods: — The actual order of accuracy of DG methods solely depends on the exact solution; DG methods of arbitrarily high formal order of accuracy can be obtained by suitably choosing the degree of the approximating polynomials. — DG methods are highly parallelizable. Since the elements are discontinuous, the mass matrix is block diagonal and since the size of the blocks is equal to the number of degrees of freedom inside the corresponding elements, the blocks can be inverted by hand (or by using a symbolic manipulator) once and for all. — DG methods are very well suited to handling complicated geometries and require an extremely simple treatment of the boundary conditions in order to achieve uniformly high-order accuracy. — DG methods can easily handle adaptivity strategies since refinement or unrefinement of the grid can be achieved without taking into account the continuity restrictions typical of conforming finite element methods. Moreover, the degree of the approximating polynomial can be easily changed from one element to the other. Adaptivity is of particular importance in hyperbolic problems given the complexity of the structure of the discontinuities.

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

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

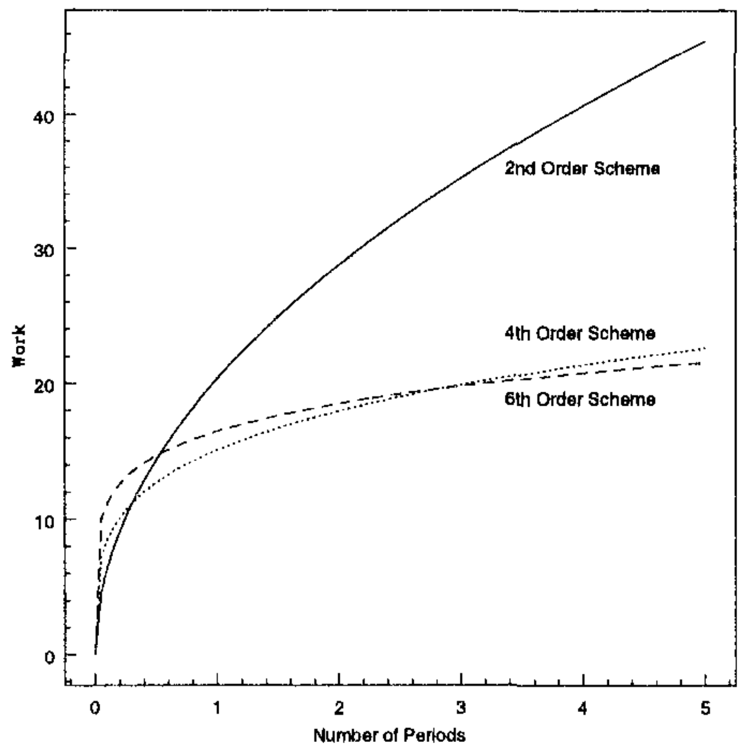

Figure 1: Computational work (number of floating-point operations) required to integrate a linear advection equation for M periods while maintaining a cumulative phase error of e = 10%. Source: [],p.10

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

Source Spectral/hp Element Methods for CFD 1999

2 Theory

2.1 Governing equations

Ideal Magneto-Hydrodynamic Equations The magneto-hydrodynamic equations (MHD) are a corner stone of theoretical astrophysics. They model the fluid mechanics of ionized interstellar media in an idealized form. In CGS unit they read as follows.

$$\frac{1}{\partial t}\begin{bmatrix}\rho\\ \rho\,\underline{v}\\ E\\ \underline{B}\end{bmatrix}+\nabla\cdot\begin{bmatrix}\rho\,\underline{v}\\ \rho\,(\underline{v}\otimes\underline{v})+(p+\frac{1}{2}||\underline{B}||^2)I-\underline{B}\otimes\underline{B}\\ \underline{v}(E+p+\frac{1}{2}||\underline{B}||^2)-\underline{B}(\underline{v}\cdot\underline{B})\\ \underline{v}\otimes\underline{B}-\underline{B}\otimes\underline{v}\end{bmatrix}+\text{forcing}=0,\quad \nabla\cdot\underline{B}=0,$$

where $\underline{\underline{a}}\otimes\underline{\underline{b}}$ is the KRONECKER product between two matrices $\underline{\underline{a}}$ and $\underline{\underline{b}}$. For vectors it can also be written as $\underline{a}\,\underline{b}^T$.

TODO: Explain above equations in detail. Name -> physical meaning ...

We want to keep the simulation simple and do not include electric current ($\sigma=\infty$) and gravity in our model. Hence, $\underline{J}=0$ and $\underline{g}=0$. This simplifies above equations to the ideal MHD equations. They are valid for dynamics in interstellar clouds where we have huge spatial dimensions of several parsecs and consequently long crossing times.

TODO: More arguments why ideal MHD equations are valid for ISM simulations

TODO: Insert ideal MHD equations

When we set the initial magnetic field strength to zero $\underline{B}=0$ no magnetic dynamics come into play and the equations reduces further to the compressible Euler equations.

2.1.1 Compressible Euler Equations

Euler equations are hyperbolic. The compressible Euler Equations are valid for perfect fluids. We assume no heat conduction ($\mathbb{T}^{i0}=\mathbb{T}^{0i}=0$), no viscosity ($\mathbb{T}^{ij}=p\not{\mathbb{K}}$, $\mu_d=0$) and no gravity $g=0$. Within in the comoving frame the *stress-energy tensor* $\mathbb{T}$ reads:

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

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

$$\partial_\nu\mathbb{T}^{\mu\nu}=0 \tag{2}$$

$$\partial_\mu(n\,u^\mu)=0 \tag{3}$$

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

$$\partial_t\rho+\nabla\cdot(\rho\,\underline{u})=0\ \ \text{(mass cons.)} \tag{4}$$

$$\partial_t(\rho\,\underline{u})+\nabla\cdot(\rho\underline{u}\underline{u}^T)+\nabla p=\underline{F}\ \ \text{(momentum cons.)} \tag{5}$$

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

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

$$E=\mathcal{I}+\mathcal{K}=\frac{p}{\gamma-1}+\frac{\rho}{2}u^2 \tag{7}$$

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

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

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

where R is the specific ideal gas constant, T is the gas temperature and $c_v = \frac{\gamma-1}{R}$ is the specific heat capacity at constant volume. The speed of sound c is a direct consequence of the ideal gas equation.

$$c^2 = \gamma\frac{p}{\rho} := Const_{\text{polytrope}} = C_P \tag{9}$$

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

$$p = C_P\,\rho^{\Gamma}, \tag{10}$$

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

Adiabatic Constant

$$\gamma = \frac{c_p}{c_v} \tag{11}$$

Prandtl Number

$$Pr = \frac{\mu\,c_p}{\kappa_H} \tag{12}$$

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

$$[\partial_t\rho + \nabla\cdot(\rho\,\underline{u})]\cdot\frac{l_r}{\rho_r\,u_r} = 0 \tag{13}$$

$$\left[\partial_t(\rho\,\underline{u}) + \nabla\cdot(\rho\underline{u}\underline{u}^T) + \nabla p - \underline{F}\right]\cdot\frac{l_r}{\rho_r\,u_r^2} = 0 \tag{14}$$

$$[\partial_t E + \nabla\cdot(\underline{u}\,(E + p))]\cdot\frac{l_r}{\rho_r\,u_r^3} = 0 \tag{15}$$

We simplify and get

$$\partial_{\tilde{t}}\tilde{\rho} + \tilde{\nabla}\cdot(\tilde{\rho}\,\underline{\tilde{u}}) = 0 \tag{16}$$

$$\partial_{\tilde{t}}(\tilde{\rho}\,\underline{\tilde{u}}) + \tilde{\nabla}\cdot(\tilde{\rho}\underline{\tilde{u}}\underline{\tilde{u}}^T) + \tilde{\nabla}\tilde{p} - \underline{\tilde{F}} = 0 \tag{17}$$

$$\partial_{\tilde{t}}\tilde{E} + \tilde{\nabla}\cdot(\underline{\tilde{u}}\,(\tilde{E} + \tilde{p})) = 0, \tag{18}$$

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

$$\tilde{t} = \frac{t}{t_r},\ \tilde{\rho} = \frac{\rho}{\rho_r},\ \tilde{\underline{u}} = \frac{\underline{u}}{u_r},\ \tilde{\nabla} = l_r\,\nabla,\ \tilde{E} = \frac{E}{\rho_r\,u_r^2},\ \tilde{p} = \frac{p}{\rho_r\,u_r^2},\ \tilde{\underline{F}} = \underline{F}\,\frac{l_r}{\rho_r\,u_r^2}. \tag{19}$$

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

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

$$\rho = \rho(t,x,y,z),\ \ (\rho\underline{u}) = (\rho\underline{u})(t,x,y,z),\ \ E = E(t,x,y,z) \tag{20}$$

are completed with

$$\gamma := 5/3,\ \ R := 1, \langle\rho\rangle := 1, \langle c\rangle := 1, \tag{21}$$

where we assume a mono-atomic gas without interacting forces. We derive

$$C_P = \frac{c_0^2}{\gamma} = 3/5 = 0.6,\ \ \langle p\rangle = C_P \cdot \langle\rho\rangle = 0.6,\ \ \langle E\rangle = \frac{\langle p\rangle}{\gamma - 1} = 0.9,\ \ \langle T\rangle = \frac{\langle c\rangle^2}{\gamma\,R} = 0.6 \tag{22}$$

Remark The average sonic mach number $\mathcal{M}$ becomes equal to the average root-means-square-velocity (RMSV).

$$\mathcal{M} = \frac{\langle \mathrm{rmsv}\rangle}{\langle c\rangle} = \left\langle \frac{\int_\Omega \sqrt{\underline{u}^2}}{\int_\Omega m} \right\rangle \tag{23}$$

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

2.1.2 Weak Formulation

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

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

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

$$\partial_t\underline{U} + \partial_x\underline{F}(\underline{U}) + \partial_y\underline{G}(\underline{U}) + \partial_z\underline{H}(\underline{U}) + \underline{S} = 0, \tag{24}$$

where

$$\underline{U} = (\rho, \rho u_1, \rho u_2, \rho u_3, E)^T \tag{25}$$

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

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

$$\underline{H}(\underline{U}) = (\rho u_3, \rho u_3 u_1, \rho u_3 u_2, \rho u_3^2 + p, u_3(E+p))^T \tag{28}$$

$$\underline{S} = (0, -f_1, -f_2, -f_z, 0)^T \tag{29}$$

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

$$\int_\Omega \left(\partial_t U_i \, \phi^i + \partial_x F_i(\underline{U}) \, \phi^i + \partial_y G_i(\underline{U}) \, \phi^i + \partial_z H_i(\underline{U}) \, \phi^i + S_i \phi^i \right) d^3x = 0 \tag{30}$$

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

$$\int_\Omega \partial_t U_i \, \psi(x,y,z)^i d^3x + \int_\Omega S_i \psi(x,y,z)^i d^3x = \tag{31}$$

$$\int_{\partial\Omega} \left(F_i(\underline{U}) \, \psi(x,y,z)^i \, n_x + G_i(\underline{U}) \, \psi(x,y,z)^i \, n_z + H_i(\underline{U}) \, \psi(x,y,z)^i \, n_z \right) d^2x,$$

$$- \int_\Omega \left(F_i(\underline{U}) \, \partial_x \psi(x,y,z)^i + G_i(\underline{U}) \, \partial_y \psi(x,y,z)^i + H_i(\underline{U}) \, \partial_z \psi(x,y,z)^i \right) d^3x \tag{32}$$

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

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

2.2 Finite Element Scheme

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

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

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

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

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

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

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

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

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

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

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

2.2.1 Polynome Ansatz

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

$$U_{\underline{i}}(t,x,y,z) \approx p_{\underline{i}}(t,x,y,z) = \sum_{\underline{j}=0}^{N_p} U_{\underline{i}}^{\underline{j}}(t)\;\Psi^{\underline{i}}(x,y,z), \tag{33}$$

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

$$\underline{i} = (i,j,k)^T, \quad i,j,k \in \mathbb{N}_0 \tag{34}$$

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

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

$$\int_\Omega \left(\sum_{j=0}^{N_p} (\partial_t U^j(t))\, \Psi^j(\underline{x}) \right) \phi(\underline{x}) d^3x + \int_\Omega S(t)\, \phi(\underline{x}) d^3x =$$

$$\int_{\partial\Omega} \left[F(t)\, \phi(\underline{x})\, n_x + G(t)\, \phi(\underline{x})\, n_z + H(t)\, \phi(\underline{x})\, n_z \right] d^2x, \tag{35}$$

$$- \int_\Omega \left[\left(\sum_{j=0}^{N_p} F^j(t)\, \Psi^j(\underline{x}) \right) \partial_x \phi(\underline{x}) + \left(\sum_{j=0}^{N_p} G^j(t)\, \Psi^j(\underline{x}) \right) \partial_y \phi(\underline{x}) + \left(\sum_{j=0}^{N_p} H^j(t)\, \Psi^j(\underline{x}) \right) \partial_z \phi(\underline{x}) \right] d^3x,$$

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

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

$$l_j(x) = \prod_{k=0, k\neq j}^{p} \frac{x - x_k}{x_j - x_k}, \; j = 0, \ldots, p, \tag{36}$$

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

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

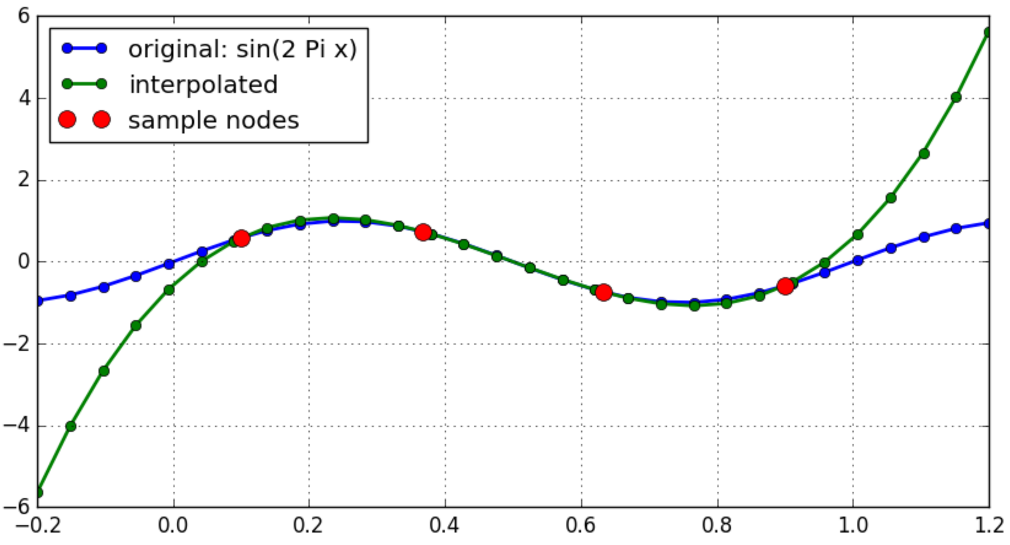

Figure 2: One-dimensinal Lagrange interpolation with four sample nodes.

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

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

$$L_{\underline{i}}(\underline{x}) = L_{ijk}(x, y, z) = l_i(x) \cdot l_j(y) \cdot l_k(z) \tag{37}$$

We write the thee-dimensional polynome as:

$$P^{\underline{i}}(t,\underline{x}) = \sum_{\underline{i}=0}^{N_p} F_{\underline{i}}(t) L_{\underline{i}}(\underline{x}) = \sum_{i,j,k=0}^{N_p} f_{ijk}(t)\cdot l_i(x)\cdot l_j(y)\cdot l_k(z) \tag{38}$$

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

Galerkin Method Replacing $\Psi^{\underline{j}}$ and ϕ accordingly, we get an explicit ansatz, called *Galerkin* method.

$$\int_\Omega \left(\sum_{\underline{j}=0}^{N_p} \dot{U}_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x})\right) L_{\underline{i}}(\underline{x}) d^3x + \int_\Omega S(t,\underline{x})\, L_{\underline{i}}(\underline{x}) d^3x =$$

$$\int_{\partial\Omega} \left[F(t)\, n_x(\underline{x}) + G(t)\, n_y(\underline{x}) + H(t)\, n_z(\underline{x})\right] L_{\underline{i}}(\underline{x})\, d^2x \tag{39}$$

$$-\int_\Omega \left[\left(\sum_{\underline{j}=0}^{N_p} F_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x})\right) \partial_x L_{\underline{i}}(\underline{x}) + \left(\sum_{\underline{j}=0}^{N_p} H_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x})\right) \partial_y L_{\underline{i}}(\underline{x}) + \left(\sum_{\underline{j}=0}^{N_p} G_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x})\right) \partial_z L_{\underline{i}}(\underline{x})\right] d^3x$$

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

$$\omega_{\underline{i}} = \int_\Omega L_{\underline{i}}(\underline{x}) d^3x \tag{40}$$

we can discretize the continuous integrals.

$$\sum_{\underline{k}=0}^{N_p} \left(\sum_{\underline{j}=0}^{N_p} \dot{U}_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}})\right) L_{\underline{i}}(\underline{x}_{\underline{k}})\, \omega_{\underline{k}} + \sum_{\underline{k}=0}^{N_p} S(t,\underline{x}_{\underline{k}})\, L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} =$$

$$\left[F^*(t) + G^*(t) + H^*(t)\right] L_{\underline{i}}(\underline{x}) \tag{41}$$

$$-\sum_{\underline{k}=0}^{N_p} \left[\left(\sum_{\underline{j}=0}^{N_p} F_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}})\right) L^{(x)}_{\underline{i}}(\underline{x}_{\underline{k}}) + \left(\sum_{\underline{j}=0}^{N_p} H_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}})\right) L^{(y)}_{\underline{i}}(\underline{x}_{\underline{k}}) + \left(\sum_{\underline{j}=0}^{N_p} G_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}})\right) L^{(z)}_{\underline{i}}(\underline{x}_{\underline{k}})\right] \omega_{\underline{k}}$$

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

$$L^{(x)}_{\underline{j}}(\underline{x}) = \partial_x L_{\underline{j}}(\underline{x}) = (\partial_x l_{j_1}(x))\cdot l_{j_2}(y)\cdot l_{j_3}(z) \tag{42}$$

$$D^{(x)}_{\underline{ij}} = D^{(x)}_{i_1i_2i_3j_1j_2j_3} = l^{(x)}_{j_1}(x_{i_1})\cdot l_{j_2}(y_{i_2})\cdot l_{j_3}(z_{i_3}) \tag{43}$$

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

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

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

$$\sum_{\underline{k}=0}^{N_p} \left(\sum_{\underline{j}=0}^{N_p} \dot{U}_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}(\underline{x}_{\underline{k}})\, \omega_{\underline{k}} + \sum_{\underline{k}=0}^{N_p} S(t, \underline{x}_{\underline{k}})\, L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} =$$

$$\left[F^*(t) + G^*(t) + H^*(t) \right] L_{\underline{i}}(\underline{x}) \tag{44}$$

$$- \sum_{\underline{k}=0}^{N_p} \left[\left(\sum_{\underline{j}=0}^{N_p} F_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) D_{\underline{ki}}^{(x)} + \left(\sum_{\underline{j}=0}^{N_p} H_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) D_{\underline{ki}}^{(y)} + \left(\sum_{\underline{j}=0}^{N_p} G_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) D_{\underline{ki}}^{(z)} \right] \omega_{\underline{k}}$$

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

2.2.2 Grid Spaces and Transformation

In this work, four grid spaces are of importance: *Face-centered grid (FCG)*, *body-centered grid (BCG)*, *Gauss nodal grid (GNG)* and *Gauss-Lobatto nodal grid (LNG)*. A visual representation can be found in figure **??**. This figure shows a one-dimensional grid of eight cells (dashed lines) or alternatively a grid with two elements (thick lines) each consisting of four nodes which implies a polynomial order of three. Major part of the work is the transformation back and forth between these grid spaces via interpolation. Another significant aspect is the relationship of elements to cells. First one overlays both grids. Ideally, they are of the same shape and cover the same physical domain. When transforming between both grid types, cells get grouped together in numbers equal to the nodal number of the superincumbent element. Interpolation happens in each group/element independently.

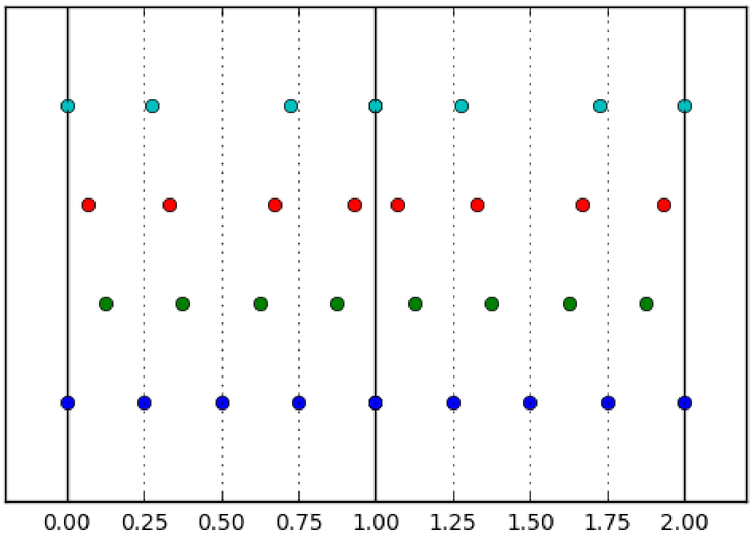

Figure 3: Nodes in a two element grid each consisting of four cells. From bottom to top: face-centered, body-centered, Gauss nodes (n = 3), Gauss-Lobatto nodes (n = 3).

Transformation from FLASH to FLEXI All following examples/figures in this document are produced by the very same procedure where we set the interpolation order $n = 3$. First the intial values are generated (or read) in the FLASH grid format. The grid is split into groups of $4^3 = 64$ neighboring cells. These groups get mapped to the associated element of the HOPR grid. Looping over all group-element pairs a three-dimensional Lagrange Polynome gets constructed according to the body-centered values of the cells. Then the new values at the Gauss/Gauss-Lobatto nodes get interpolated and stored according to the ordering convention of FLEXI.

In order to do useful analysis and visualization, the FLEXI data gets again back-interpolated to BCG. This is similar to what the visualization routines in FLEXI do.

Interpolation Error Estimate As a meassure of interpolation error of an original function f and its interpolated counterfeit $\widetilde{f}$ the relative root-means-sqare variance represents one distance meassure.

$$f_{\text{RMS}} = \sqrt{\frac{1}{N}\sum_i^N f_i^2} \tag{45}$$

$$\Delta_{\text{RMS}} = (f - \widetilde{f})_{\text{RMS}} = \sqrt{\frac{1}{N}\sum_i^N (f_i - \widetilde{f}_i)^2} \tag{46}$$

$$\Delta_{\text{RMS,rel.}} = \frac{\Delta_{\text{RMS}}}{f_{\text{RMS}}} \tag{47}$$

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

Another error estimate could be the absolut mean error:

$$f_{\text{ABS}} = \frac{1}{N}\sum_i^N |f_i| \tag{48}$$

$$\Delta_{\text{ABS}} = (f - \widetilde{f})_{\text{ABS}} = \frac{1}{N}\sum_i^N |f_i - \widetilde{f}_i| \tag{49}$$

$$\Delta_{\text{ABS,rel.}} = \frac{\Delta_{\text{RMS}}}{f_{\text{ABS}}} \tag{50}$$

In most cases the total absolute error yields smaller results, since runaways get weighted more in the rmse case. A proper handling of runaways especially in interpolation of shocks is still an open question.

2.2.3 Flux Functions

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

Riemann Problem Many of the methods are based on solving the RIEMANN problem between the states q_L and q_R in order to define the numerical flux $F^*(q_L, q_R)$. For one way to approach this, it is useful to view the data Q^n at time t_n as defining a piecewise constant function $q^n(t_n, x)$ which has the value Q_i^n for all x in the interval C_i. Suppose we could solve the conservation law exactly over the time interval $[t_n, t_{n+1}]$ with initial data $q^n(t_n, x)$. We call the resulting function $q_n(t, x)$ for $t_n < t < t_{n+1}$. Then we might consider defining the numerical flux F_i^{*n} by

$$F_i^{*n} = \frac{1}{k} \int_{t_n}^{t_{n+1}} f(q^n(t, x_i))\, dt \tag{51}$$

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

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

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

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

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

HLLE (Harten-Luv-Lax Entropy-fix) flux Another approach is to approximate the full Riemann solution by a single intermediate state bounded by two waves moving at speeds s_1 and s_2.

The wave speeds should be some approximations to the minimum and maximum wave speeds that would arise from this particular Riemann data and the intermediate state can then be calculated by the condition of conservation. This approach is developed by Harten, Lax, and van Leer originally [108] and improved by Einfeldt [78].

PPM Solver

Bouchut5 Solver

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

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

2.2.4 Time Integration

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

$$\frac{d}{dt}y = f(t, y). \tag{52}$$

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

$$y^{n+1} = y^n + \Delta t \cdot f(t^n, y^n) \tag{53}$$

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

$$y^{n+1} = y^n + \Delta t \cdot f\left(t^n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2} \cdot f(t^n, y^n)\right) \tag{54}$$

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

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

$$\Delta t \leq \frac{\Delta x}{u} \tag{55}$$

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

$$\Delta t = C \cdot \min_x \left(\frac{\Delta x}{u(x)}\right) \tag{56}$$

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

In a three-dimensional orthogonal domain the timestep reads

$$\Delta t = C \cdot \min_{\underline{r}} \left(\frac{dx}{|v_x(\underline{r})| + c(\underline{r})}, \frac{dy}{|v_y(\underline{r})| + c(\underline{r})}, \frac{dz}{|v_z(\underline{r})| + c(\underline{r})}\right) \tag{57}$$

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

2.3 Supersonic Turbulences

Turbulences are very common phenomena in nature. They can be desired as well as unsoliceted. In astrophysics turbulence are suspected to play a major role in star formation in interstellar clouds. Hence, a good understanding of the underlying mechanics is crucial in order to model them correctly in numerical simulations. While turbulences in incompressible media has been thoroughly studied in the past, there is still an on-going debate about what additional dynamics compressibility brings especially in supersonic setups where shocks emerge.

Grid point requirement

$$N \propto Re^{9/4} \tag{58}$$

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

There is a wide range of time scales in a turbulent flow, so the system of equations is stiff. Implicit time advancement and large time steps are routinely used for stiff systems in general-purpose CFD, but these are unsuitable in DNS because complete time resolution is needed to describe the energy dissipation process accurately. Specially designed implicit and explicit methods have been developed to ensure time accuracy and stability (see e.g. Verstappen and Veldman, 1997).

Reynolds (in Lumley, 1989) noted that it is essential to have accurate time resolution of all the scales of turbulent motion. The time steps must be3.10 SUMMARY 113 adjusted so that fluid particles do not move more than one mesh spacing. Moin and Mahesh (1998) demonstrated the strong influence of time step size on small-scale amplitude and phase error.

Laminar flow becomes unstable above a certain Reynolds number $R = uL/\nu$

Say something about fluctuation property around mean see Hydrodynamic and MHD Turbulent Flows Chapter 1

Three solving strategies: direct numerical simulation large eddy simulation and RANS (Reynolds-averaged Navier-Stokes) simulation.

Turbulences is characterized by a random fluctuation of flow variable in time meassured at a fixed point in space. Hence, the time evolution can be described via the *Helmholtz decomposition.*

$$q(t) = \langle q \rangle_t + \tilde{q}(t), \quad \langle \tilde{q} \rangle \equiv 0 \tag{59}$$

where

$$\langle q \rangle_t = \frac{\int_{t_0}^{t1} q(t) dt}{t_1 - t_0} \tag{60}$$

is the time-average of the flow property $q(t)$. A turbulent flow can be globally described in terms of the mean values of the flow properties like density, velocity and pressure. In the following pages important turbulence measures are introduced and briefly explained.

Reynolds Number

$$R = \frac{\rho_0 \, V_0 \, L}{\mu} \tag{61}$$

Mach Number $\mathcal{M}$

$$\mathcal{M} = \sqrt{\frac{\int_\Omega \rho \, \underline{u}^2 \, d\Omega}{\int_\Omega \rho \, d\Omega}} \tag{62}$$

Bulk Motion $\underline{\mathcal{P}}$

$$\underline{\mathcal{P}} = \int_\Omega \rho \, \underline{u} \, d\Omega \tag{63}$$

Kinetic Energy $\mathcal{K}$

$$\mathcal{K} = \int_\Omega \frac{\rho}{2} \, \underline{u}^2 \, d\Omega \tag{64}$$

Kinetic Energy Dissipation ϵ

$$\epsilon = -\frac{d\mathcal{K}}{dt} \tag{65}$$

$$(\underline{\underline{S}})_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \lambda \delta_{ij} \frac{\partial u_k}{\partial x_k} \tag{66}$$

Bulk viscosity coefficient $\lambda := 2/3$

$$\epsilon_1 = 2\,\frac{\mu}{\rho_0\,\Omega} \int_\Omega \underline{\underline{S}} : \underline{\underline{S}}\, d\Omega \tag{67}$$

$$\epsilon_2 = 2\,\frac{\mu_v}{\rho_0\,\Omega} \int_\Omega (\nabla \cdot \underline{u})^2\, d\Omega \tag{68}$$

$$\epsilon_3 = -\frac{1}{\rho_0\,\Omega} \int_\Omega p\, \nabla \cdot \underline{u}\, d\Omega \tag{69}$$

Vorticity ω and Enstrophy $\mathcal{E}$

$$\underline{\omega} = \nabla \times \underline{u} \tag{70}$$

$$\mathcal{E} = \frac{\int_\Omega \frac{\rho}{2}\, \underline{\omega}^2 d\Omega}{\int_\Omega \rho\, d\Omega} \tag{71}$$

For subsonic mach numbers:

$$\epsilon \approx 2 \frac{\mu}{\rho_0}\, \mathcal{E} \tag{72}$$

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

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

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

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

$$k = \frac{2\pi}{\lambda} \tag{73}$$

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

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

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

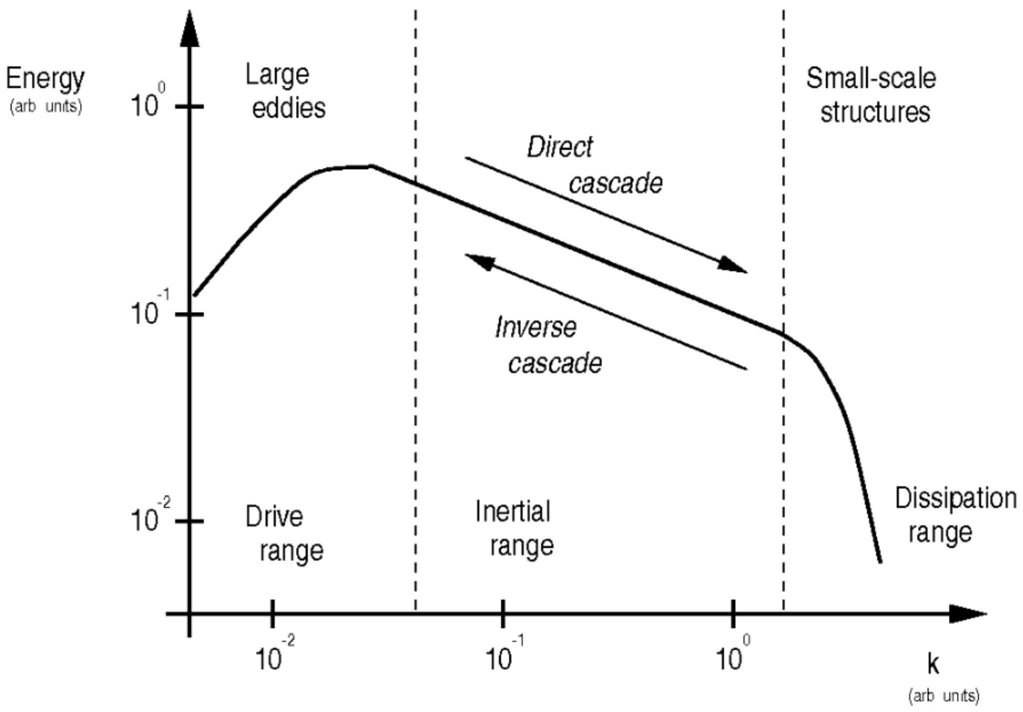

Figure 4: Schematic representation of the K41 picture of turbulence showing the spatial energy spectrum as an example. Source: [**?**], p6

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

A detailed discussion of turbulences can be found in: H K Versteeg and W Malalasekera: An Introduction to Computational Fluid Dynamics

AKlRA YOSHIZAWA: HydrodynaIllic and MagnetohydrodynaIllic Turbulent Flows Modelling and Statistical Theory

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

$$P(\underline{k}) = \frac{1}{2}\,\hat{\underline{u}} \cdot \hat{\underline{u}}^\dagger, \tag{74}$$

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

$$\tilde{P}(k)\,dk = 4\pi\,k^2\ \frac{1}{2}\,\hat{\underline{u}} \cdot \hat{\underline{u}}^\dagger\,dk, \tag{75}$$

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

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

$$\tilde{E}(k) = \frac{1}{V_\Omega} \int_0^\infty 4\pi\, k^2\, \hat{\mathcal{K}} \cdot \hat{\mathcal{K}}^\dagger dk \tag{76}$$

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

$$\int_{-\infty}^{\infty} |Y(x)|^2\, dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\hat{Y}(k)|^2\, dk. \tag{77}$$

Probability Distribution Functions Using all the cells in our grid, we obtain the cumulative distri- butions, 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 vortic- ity, $\underline{\omega}$. To obtain these distributions, the corresponding quantities are binned linearly.

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

$$\Pr[\rho_A \leq \rho' \leq \rho_B] = \int_{\rho_A}^{\rho_B} \mathrm{PDF}[\rho](\rho')\, d\rho' \tag{78}$$

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

Explain skewness fit, log, log-log scale ???

2.3.1 Turbulent Forcing

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

Based on [**?**] we formulate

$$\hat{d\mathbf{f}}(\mathbf{k}, t) = \frac{3}{\sqrt{1 - 2\zeta + 3\zeta^2}} \left[-\hat{\mathbf{f}}(\mathbf{k}, t)\frac{dt}{T} + F_0 \left(\frac{2\sigma^2(\mathbf{k})}{T} \right)^{1/2} \mathbf{P}_\zeta(\mathbf{k}) \cdot d\mathbf{W}_t \right] \tag{79}$$

$$(P_{ij})(\mathbf{k}) = \zeta\, P_{ij}^{\perp}(\mathbf{k}) + (1-\zeta) P_{ij}^{\|} = \zeta\, \delta_{ij} + (1 - 2\,\zeta)\frac{k_i k_j}{k^2} \tag{80}$$

$$\tag{81}$$

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

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

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

2.3.2 Shocks

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

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

Solving Riemann problem.

Due to the nonlinearity of the Euler equations

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

$$q_t + q\, q_x = \epsilon\, q_{xx} \tag{82}$$

We set $\epsilon = 0$.

The characteristics satisfy

$$x'(t) = q(t, x(t)) \tag{83}$$

and along each characterstic q is constant, since

$$\begin{aligned} \frac{d}{dt} q(t, x(t)) &= \partial_t q(t, x(t)) + \partial_x q(t, x(t)) x'(t) & (84)\\ &= q_t + q\, q_t & (85)\\ &= 0. & (86) \end{aligned}$$

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

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

$$u_{shock} = \frac{u_L + u_R}{2} \tag{87}$$

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

When we set following initial condition:

$$U(0, x) = \begin{cases} 1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad 0 \leq x < 1 \\ 2 & \text{if} \quad 1 \leq x < 2 \\ 0 & \text{if} \quad x > 2 \end{cases} \tag{88}$$

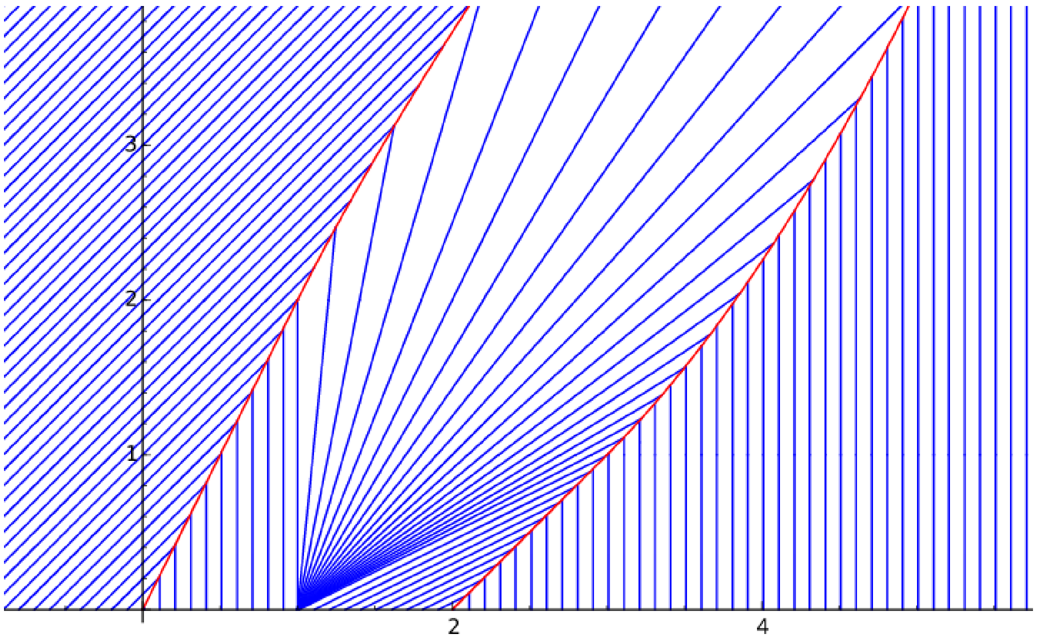

Figure 5: Source: https://calculus7.org/2015/11/27/rarefaction-colliding-with-two-shocks/

The picture above shows multiple interesting features of shocks.

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

The equation C.6) is a model of C.7) valid only for small e and smooth q. When it breaks down, we must return to C.7). If the initial data is smooth and e very small, then before the wave begins to break the eqxx term is negligible compared to the other terms and the solutions to the two PDEs look nearly identical. However, as the wave begins to break, the second derivative term qxx grows much faster than qx, and at some point the eqxx term is comparable to the other terms and begins to play a role. This term keeps the solution smooth for all time, preventing the breakdown of solutions that occurs for the hyperbolic problem. This smooth solution has a steep transition zone where the viscous term is important. As e —> 0 this zone becomes sharper and approaches the discontinuous solution known as a shock. It is this vanishing-viscosity solution that we hope to capture by solving the inviscid equation.

Rarification wave is pulling both shocks together so the eventually collide

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

$$u_{shock}\,(u_L - u_R) = f(u_L) - f(u_R) \tag{89}$$

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

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

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

$$\Delta s = c_v \ln\left[\frac{p_2}{p_1}\left(\frac{\rho_1}{\rho_2}\right)\right] \tag{90}$$

This discussion is important on defining Flux functions.

2.3.3 Shock Capturing

In hypersonic simulations the solver has to deal with strong shocks in an accurate and robust manner. The utilized shock capturing strategy consists of two parts: sensoring and capturing.

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

Sensoring Many shock sensors have been developed. Within FLEXIfollowing indicators are available.

Jameson [?]

Ducros [?]

Persson [?]

Based on the PERSSON indicator we develop a *smoothness* sensor. The basic idea is to find a meassure for the variance of the highest frequencies in modal space of the polynome.

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

$$u = \sum_{i=1}^{N_p} u_i \psi_i, \tag{91}$$

where $N(p)$ is the total number of terms in the expansion and ψ_i are the LEGENDRE basis functions.

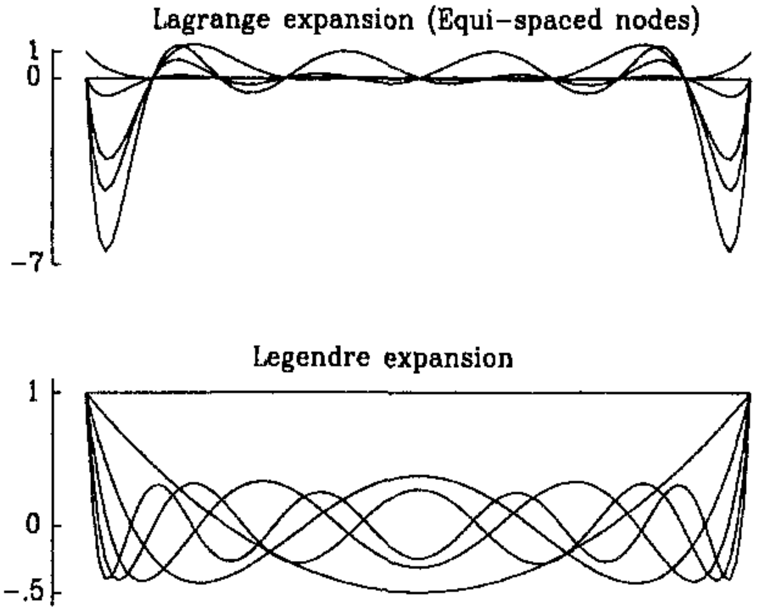

Figure 6: Expansion modes for the Lagrange and the Legendre Basis.

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

$$u_{N_p-1} = \sum_{i=1}^{N_p-1} u_i \psi_i, \tag{92}$$

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

$$s = \log_{10} \frac{\langle u - u_{N_p-1}, u - u_{N_p-1} \rangle}{\langle u, u \rangle}, \tag{93}$$

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

The smaller the indicator s, the smoother is the approximating solution. By setting a specific threshold for s one can decide when to switch between DG and FV mode. This procedure is done at every timestep hence the elements in FV mode should follow along the shock waves throughout the domain.

Capturing

Entropy Stable Flux [?]

Local Finite Volume

Split Schemes See [?] for further details.

Artificial Viscosity ...

$$\frac{\partial \underline{U}}{\partial t} + \nabla \cdot \underline{\underline{F}} = \nabla \cdot (\epsilon \nabla U) \tag{94}$$

The amount of viscosity varies for each element depending on the current shock strength. We have to consider two cases.

If the element is in FV mode the AV is set quadratic proportional to the maximal RMS v_{rmsv} within the element.

$$\epsilon = \epsilon_0 \cdot \max(v_{rmsv})^2 \tag{95}$$

This step is necessary for FV schemes who do not handle strong shocks via their flux schemes.

In case of DG mode the amount of AV is based on the *Persson Indicator* introduced above.

$$\epsilon = \begin{cases} 0 & \text{if} \quad s < s_0 - \kappa \\ \epsilon_0 & \text{if} \quad s > s_0 + \kappa \\ \frac{\epsilon_0}{2}\left(1 + \sin\frac{\pi(s-s_0)}{2\kappa}\right) & \text{else} \end{cases} \tag{96}$$

The parameters ϵ_0 and κ are chosen empirically. Since we do not have any natural viscosity the artifical one must be as small as possible; just enough for diffusing velocity spikes in the presence of shocks. Typical values are around $\epsilon_0 \propto 10^{-10}$.

3 Numerical Setup

3.1 Periodic Box and Initial Conditions

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

include some fancy schmancy 3d picture

Based on considerations in section **??** we initialize the state as listed in table **??**.

Table 1: Overview of initial values set for all turbulence simulations.

Name	Symbol	Value
density	ρ_0	1.0
x-velocity	$u_{x,0}$	0.0
y-velocity	$u_{y,0}$	0.0
z-velocity	$u_{z,0}$	0.0
pressure	$p_{x,0}$	0.6
x-momentum	$p_{x,0}$	0.0
y-momentum	$p_{y,0}$	0.0
z-momentym	$p_{z,0}$	0.0
energy	$E_{x,0}$	0.9

Remark Some numerical schemes actually solve the magneto-hydrodynamics equations. By setting the initial magnetic density flux field $\underline{B}_0 = 0$, the MHD equations resemble the compressible Euler equations.

3.2 CFD Frameworks

For the turbulence simulations two numerical frameworks for computational fluid dynamics are compared.

FLASH Cite from the homepage.

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

This framework is very established within the astrophysics community and represents the trusted basis for testing and comparing the modern higher-order schemes provided by FLEXI.

FLASHalready provides all modules necessary for the turbulence simulations done as part of the thesis. A complete set of source files and setup calls are provided in the digital appendix.

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

There is broad range of hydrodynamic and magnet-hydrodynamic solvers available but we will focus on three split schemes which were intentionally constructed for supersonic turbulence simulations: PPM, Bouchut3 and Bouchut5. They were already discussed briefly in section **??**.

The modules for polytropic cooling and turbulent forcing besides their main purpose addiontally calculate interesting simulation data like total energy, dissipated kinetic energy, etc. and write them into plain-text data files.

For performance profiling a current timestamp gets written down at every timestep. This very simple yet naive approach has the disadvantage that input/output operations, especially saving large snapshots, contribute to the overall account as well. Ephemeral bottle necks in network transfer rates of the HPC infrastructure unfairly increase the total wallclock time. This issue is further discussed in the conclusion.

FLEXI Cite from the homepage.

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

This modern framework did not originally provide or just rudimentarily offered any facilities for polytropic cooling, turbulent forcing, bulk motion correction and shock capturing.

Consequently, missing modules were ported from FLASHand adapted accordingly to fit into FLEXI's infrastructure. With regard to configuration and desired effect on the physics simulation these additions were implemented as closely to FLASHas possible.

3.3 Turbulent Forcing

The theoretical framework has been given in section **??**. The turbulent stirring module *Stir-Girichids* by P. Girichids was ported to FLEXIand tested. The module can be configured by a multitude of parameters at runtime. What follows is an overview of the most important one.

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

kmin, kmax Range of modes where to apply forcing. Commonly, the range is set between 1 and 7. Stirring on only the first mode can be translated to a force field with distinct features half-size the box. Higher modes divide the box further down accordingly. All simulations presented here are limited to only the first mode since one wants to avoid imprinting an artificial powerspectrum

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

A depiction of the velocity field after energy injection for the first time on a constant state is is shown in fig. **??** and its associated three-dimensional powerspectrum in fig. **??**. In both figures the nearly equal distribution of kinetic energy on the first seven modes is very clear.

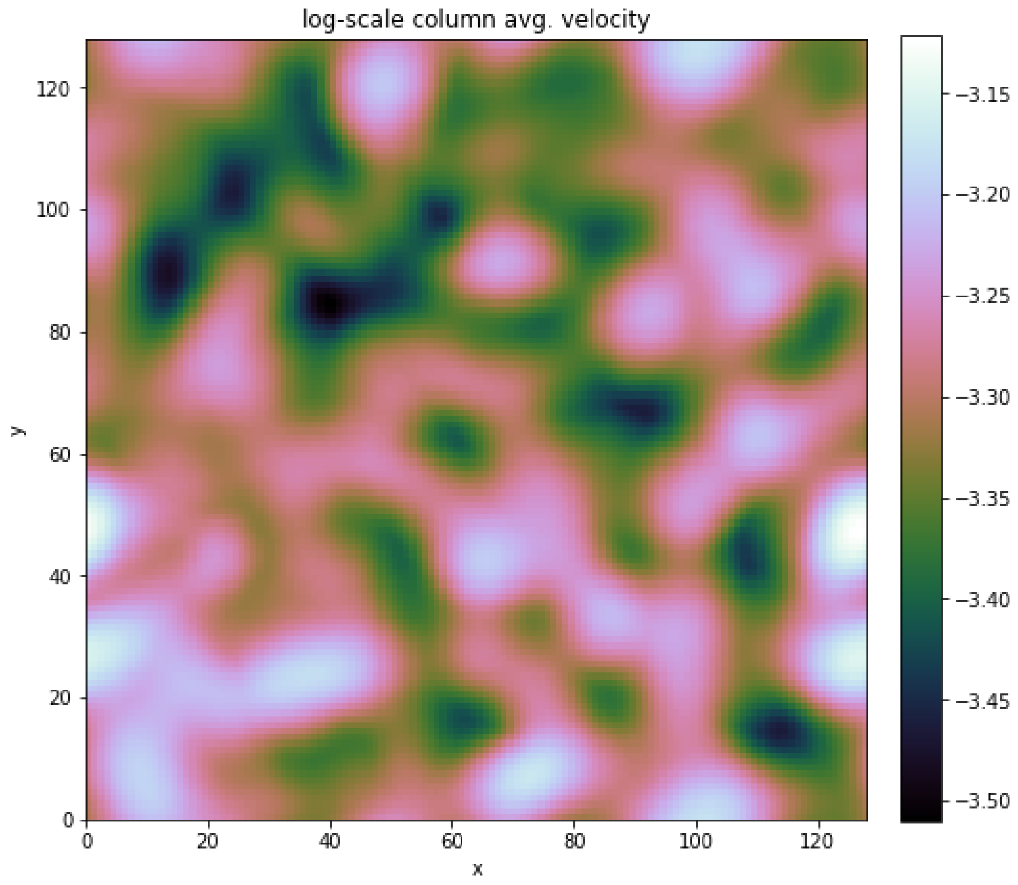

Figure 7: Velocity immediately after first-time turbulent forcing on the first seven modes.

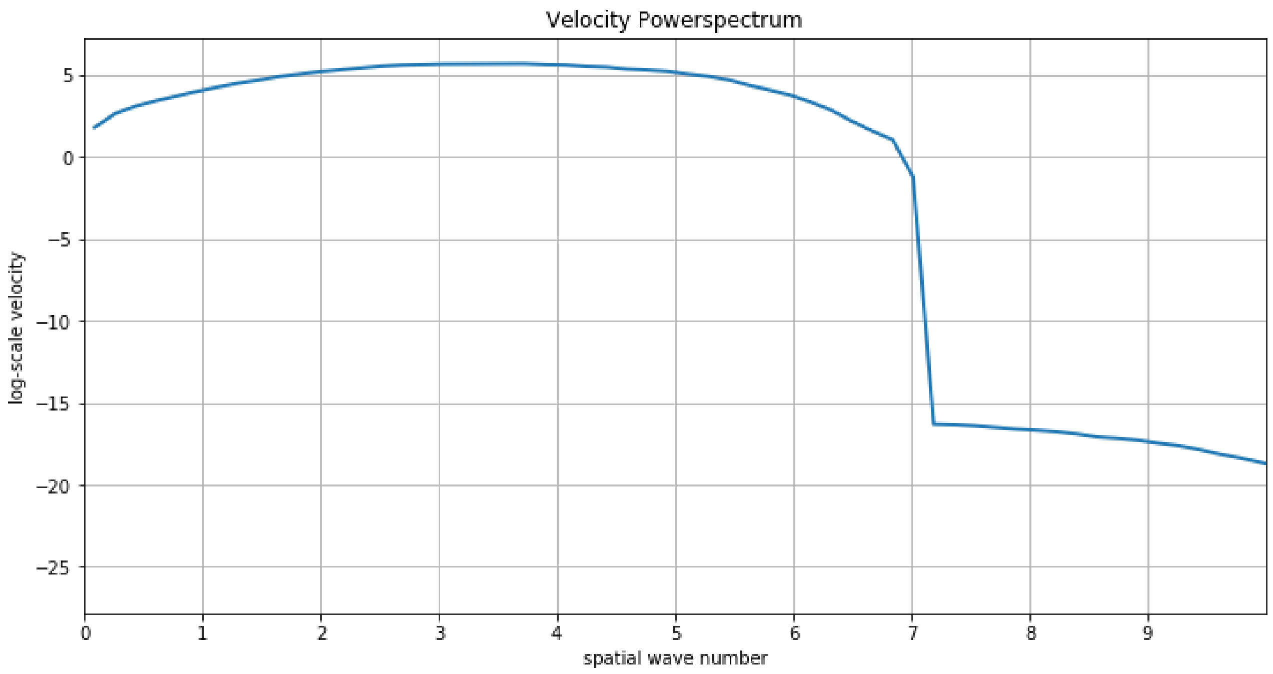

Figure 8: Shell-averaged three-dimensional velocity powerspectrum obtained from the velocity field shown in fig. **??**.

3.4 Indicator

In section **??** we discussed ways of how to detect and handle shocks. Here we perform a comparing study of two manifestations of the Persson indicator and clarify the advantages over the other.

We remember the definition of the smoothness indicator, eqn. (**??**),

$$s = \log_{10} \frac{\langle u - u_{-1}, u - u_{-1} \rangle}{\langle u, u \rangle}, \tag{97}$$

which is the classical indicator given per Persson. The first variation of it, called *indicator A*, reads

$$s_A = \log_{10} \max \left(\frac{\left\langle u^2 - u^2_{N_p--1} \right\rangle}{\langle u^2 \rangle}, \frac{\left\langle u^2_{N_p-1} - u^2_{N_p-2} \right\rangle}{\left\langle u^2 - u^2_{N_p-1} \right\rangle} \right) \tag{98}$$

And the second, called *indicator B*,

$$s_B = \log_{10} \max_q \frac{\left\langle u - u_{N_p-q} \right\rangle^2}{\left\langle u_{N_p-q} \right\rangle^2} \tag{99}$$

Fig. **??** and fig. **??** show a fully developed turbulence with shocks and their associated heat maps of indicator values calculated from the pressure variable. A normalized frequency distribution (PDF) of indicator values s is plotted in fig. **??**.

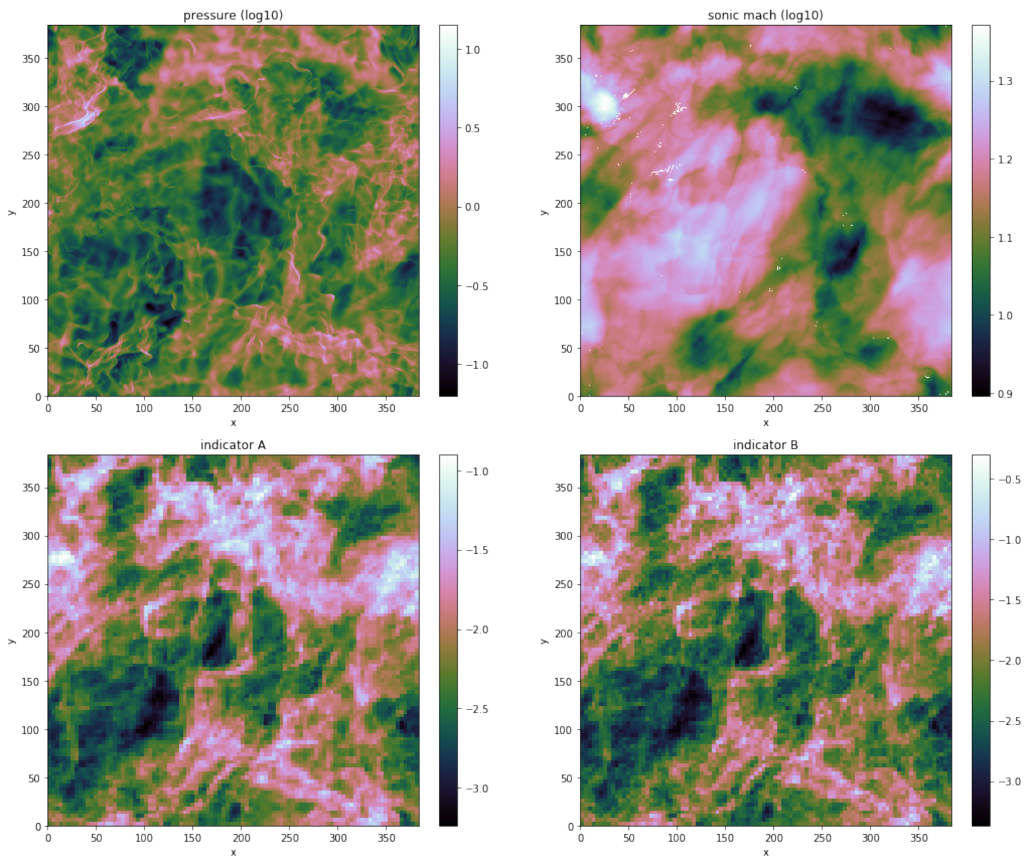

Figure 9: Snapshot of hyper-sonic turbulence.

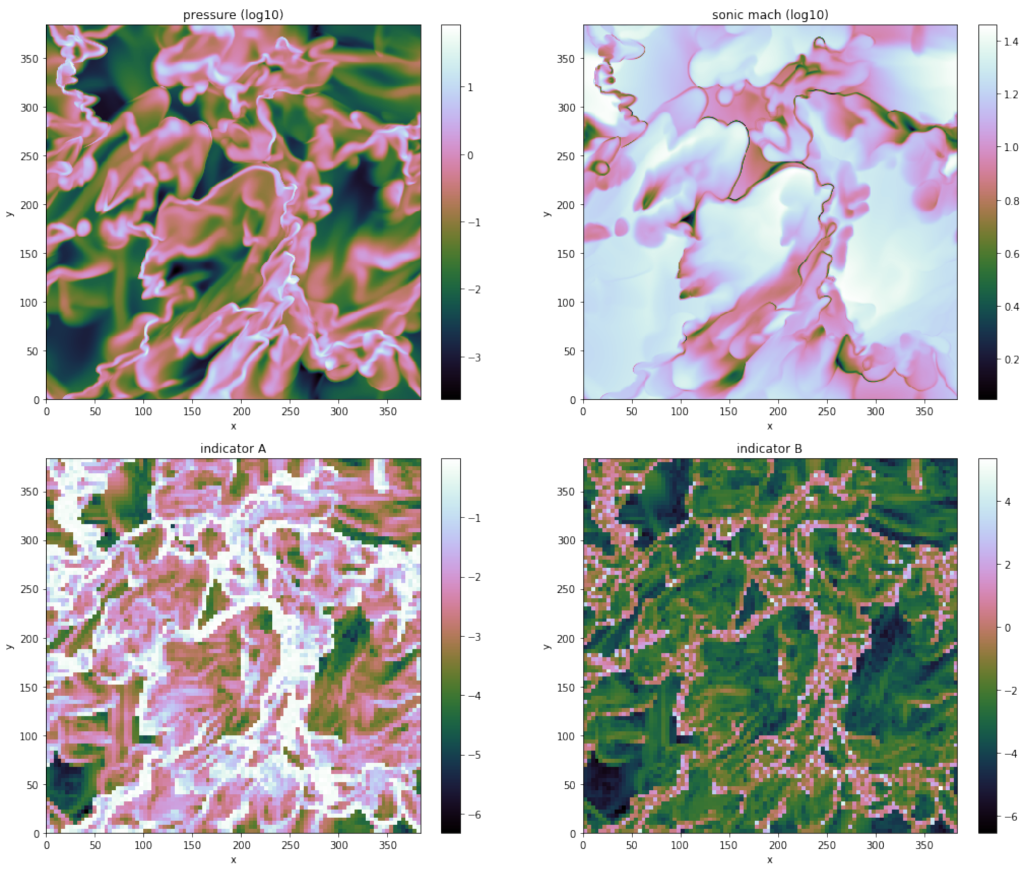

Figure 10: Slice with thickness of one element.

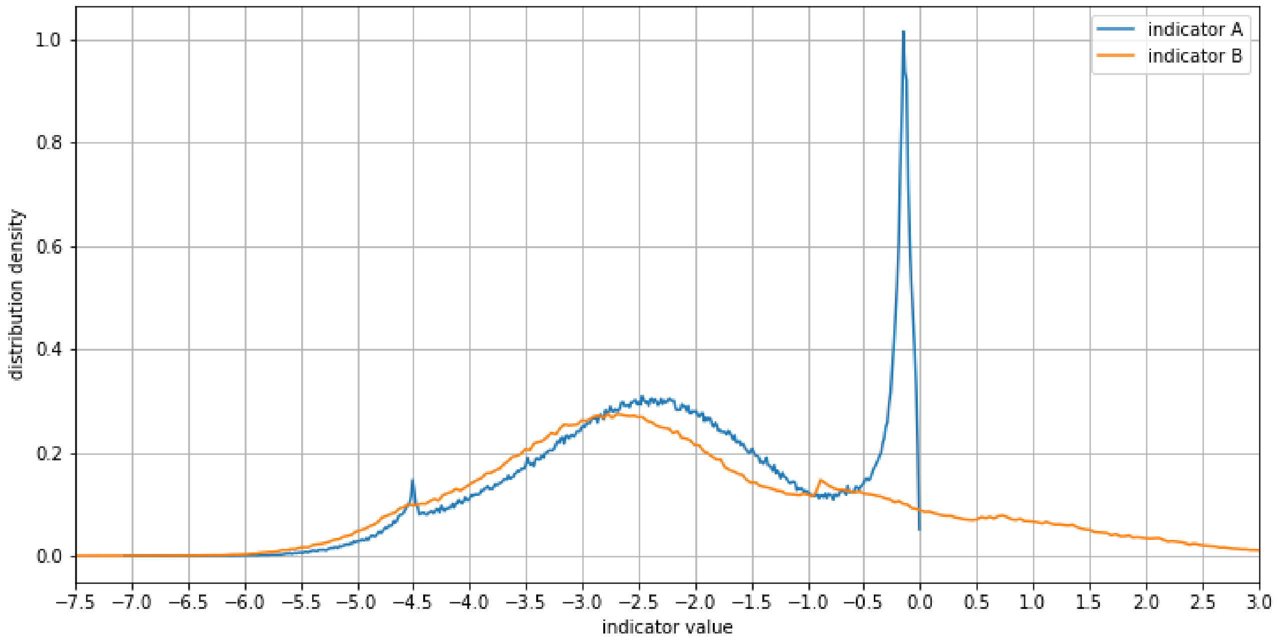

Figure 11: Probability Distribution of calculated indicator values.

On average, fig. **??**, both versions yield a similar picture. But taking out a slice of the turbulent box we get very differing results. The most prominent feature is the higher sensitivity of indicator A and its limitation to only negative values. The latter is a result of the normalizing fractions in eqn. (**??**). Both variants trail the shock fronts with sufficient accuracy. Considering the PDF in fig. **??**, one can observe spikes in both curves in the interval between -1 and 0. They herald the realm of strong shocks. The broad hill around -2.5 contains the majority of stressed elements which were affected by a recently pervading shock front. The little spike at -4.5 stems from a bias introduced the indicator A. Numerous tests revealed a necessary threshold value of -4.5 where the switch from DG ode to FV mode takes place. Clearly, indicator A unambiguously signalizes elements with unresolvable discontinuities. This opens the possibility to further strengthen the DG scheme for a broader coverage of the lower stressed domain and hand over shock ridden elements to specialized treatment.

3.5 Polynomial Interpolation

For analysis and defining initial conditions with DG based code interpolation techniques for translating correct state values between different grid spaces are necessary. See section **??** for theoretical disuccusion. In this section we want to present observations and draw important conclusions on how to correctly interpret the simulation data. Furthermore, the correctness of the interpolation code is shown and hence its produced results can be trusted.

When all data points represent a smooth pathway and are sufficiently close together an interpolating polyonme can resemble the original function very precisely. Fig. **??** gives an example of this well-behaving case.

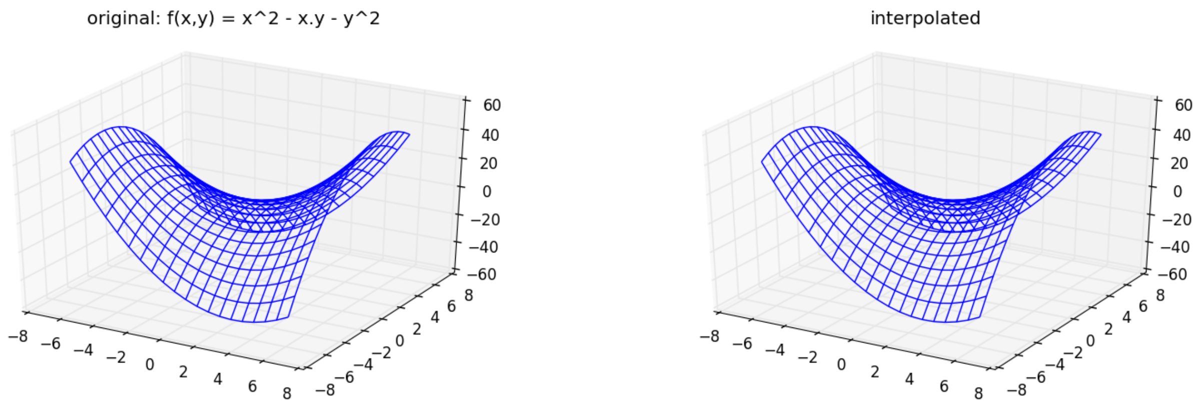

Figure 12: Two-dimensional third-order interpolation of a second-order function. relative rms error = 0.000000%

However, since we want to work with hyper-sonic shocks, discontinuities are present throughout the data. Spurious oscillations and severely erranous approximations are the consequence. An artificial stress test see fig. **??**, reveals the strength of the DG method. Since, the interpolation is always limited on small patches of the whole domain the global features still get represented at the cost of considerable deviations at smaller scales.

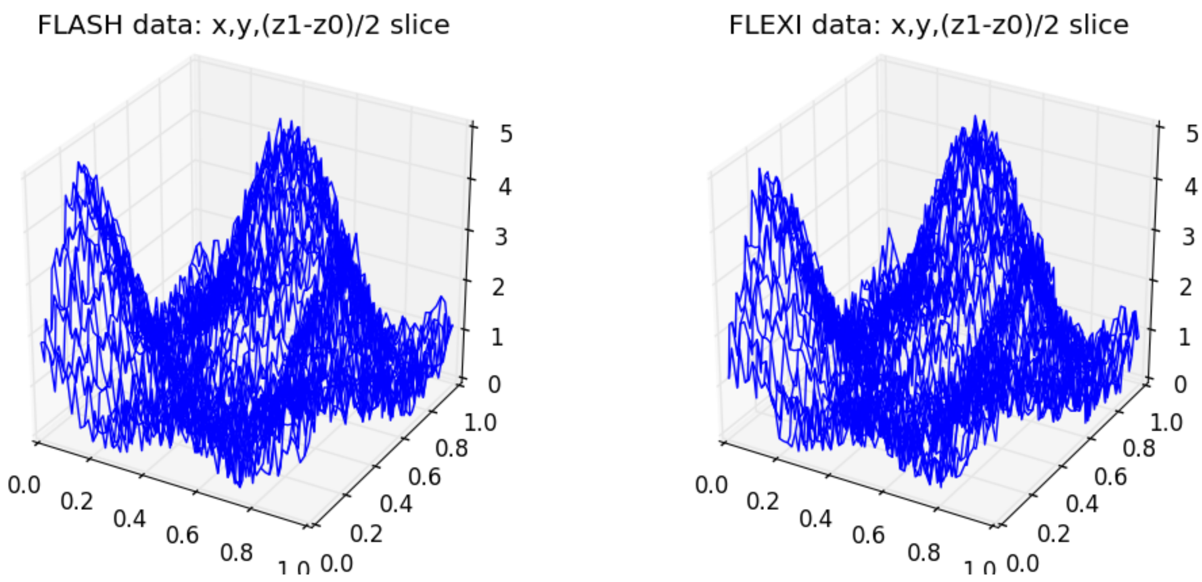

Figure 13: Stress test: 3D superposed sines and cosines with random noise: Third-order 3d interpolation of finite volume data to DG data (Gauss-Lobatto nodes). relative rms error = 14.987457%

When transferring real-world data from FLASH(finite-volume) to FLEXI(DG) we get following results.

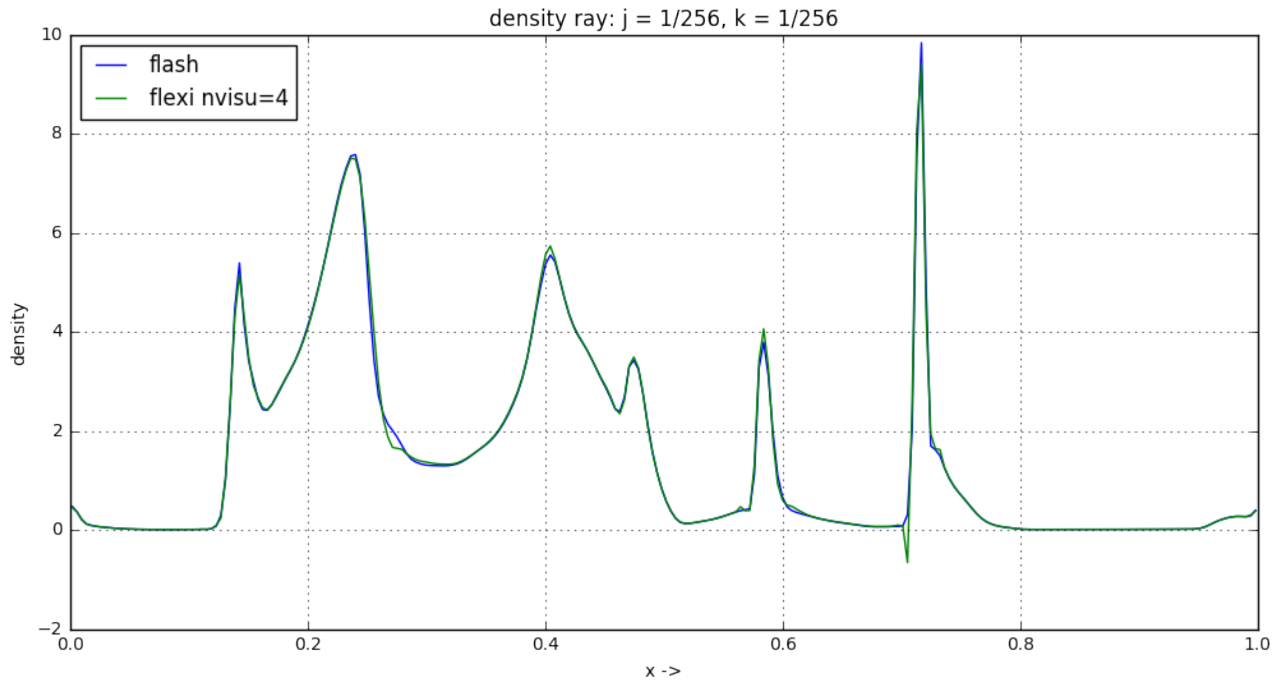

Figure 14: Foobar fo bar

4 Results & Discussion

4.1 Decaying Turbulence

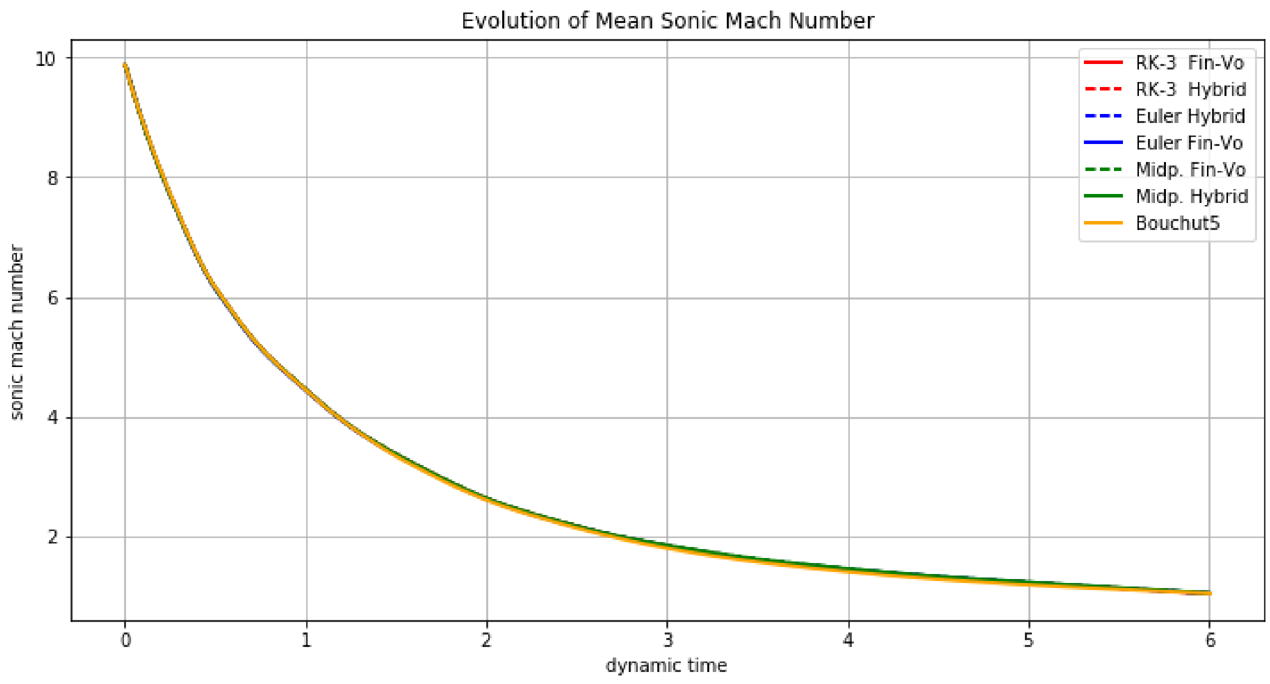

Figure 15: Foobar fo bar

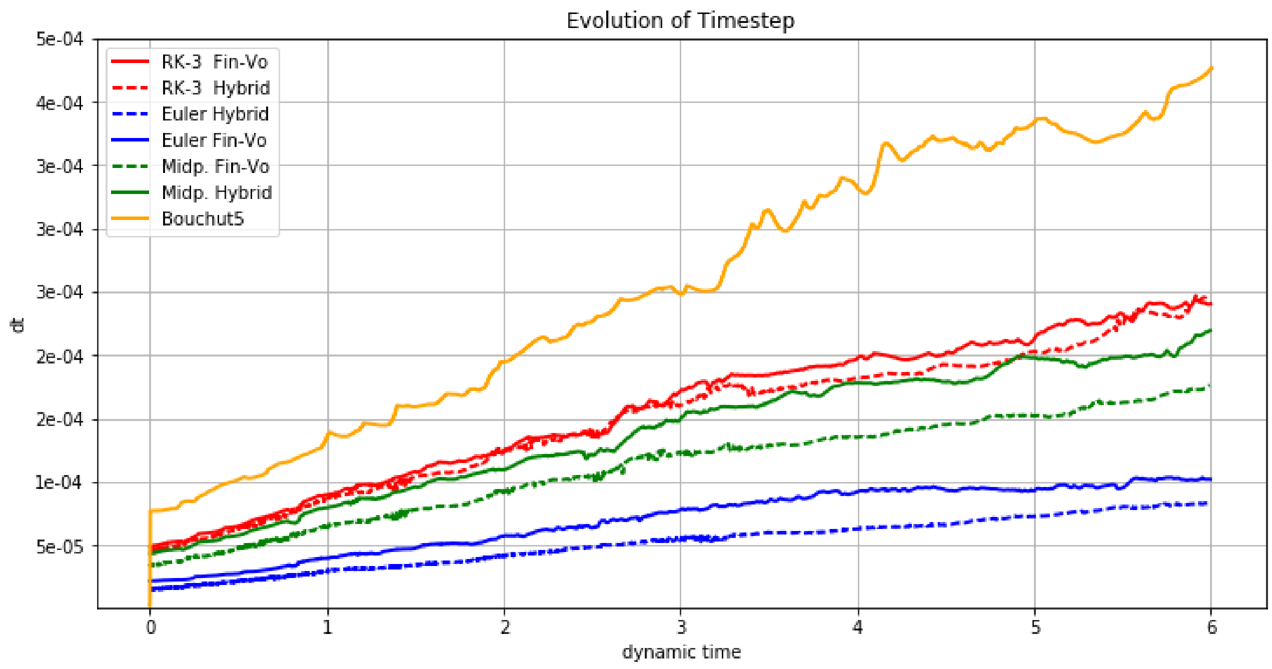

Figure 16: Foobar fo bar

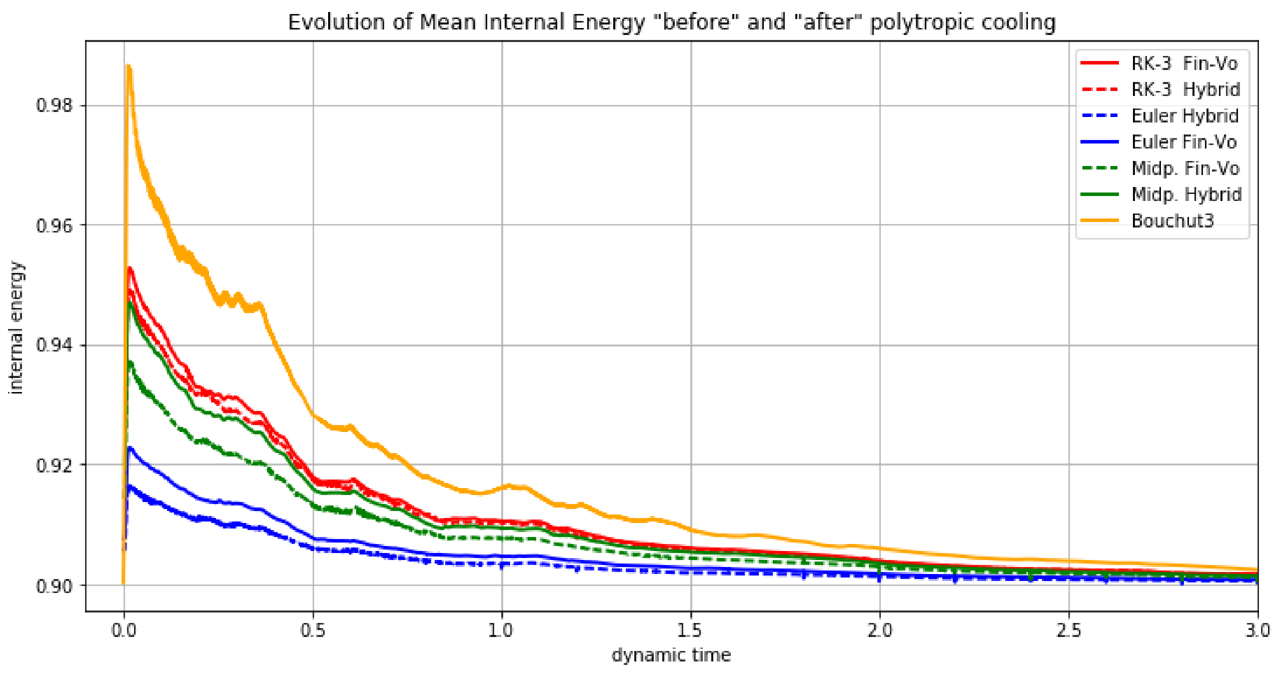

Figure 17: Foobar fo bar

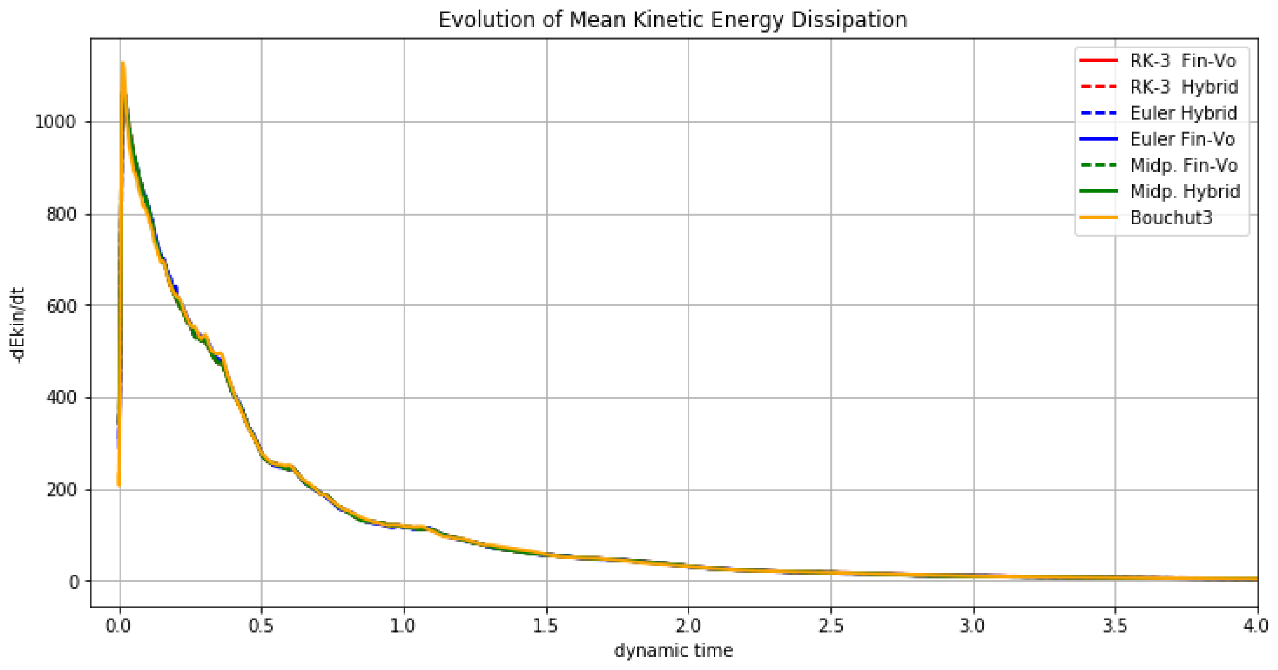

Figure 18: Foobar fo bar

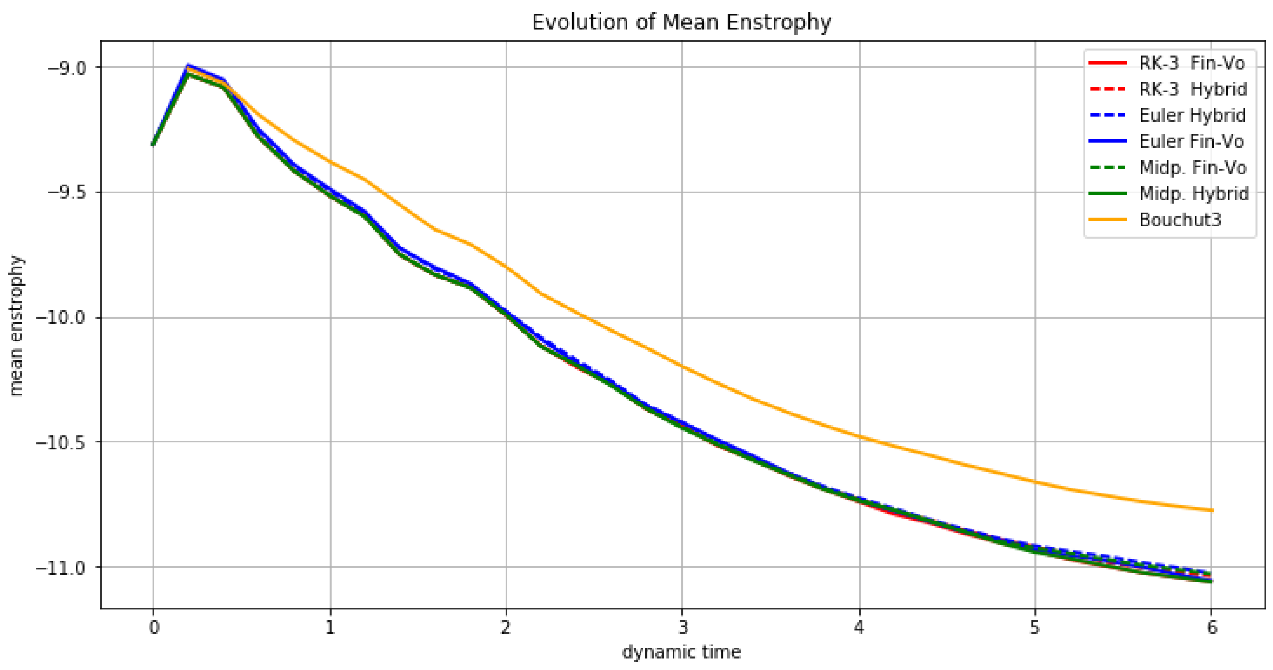

Figure 19: Foobar fo bar

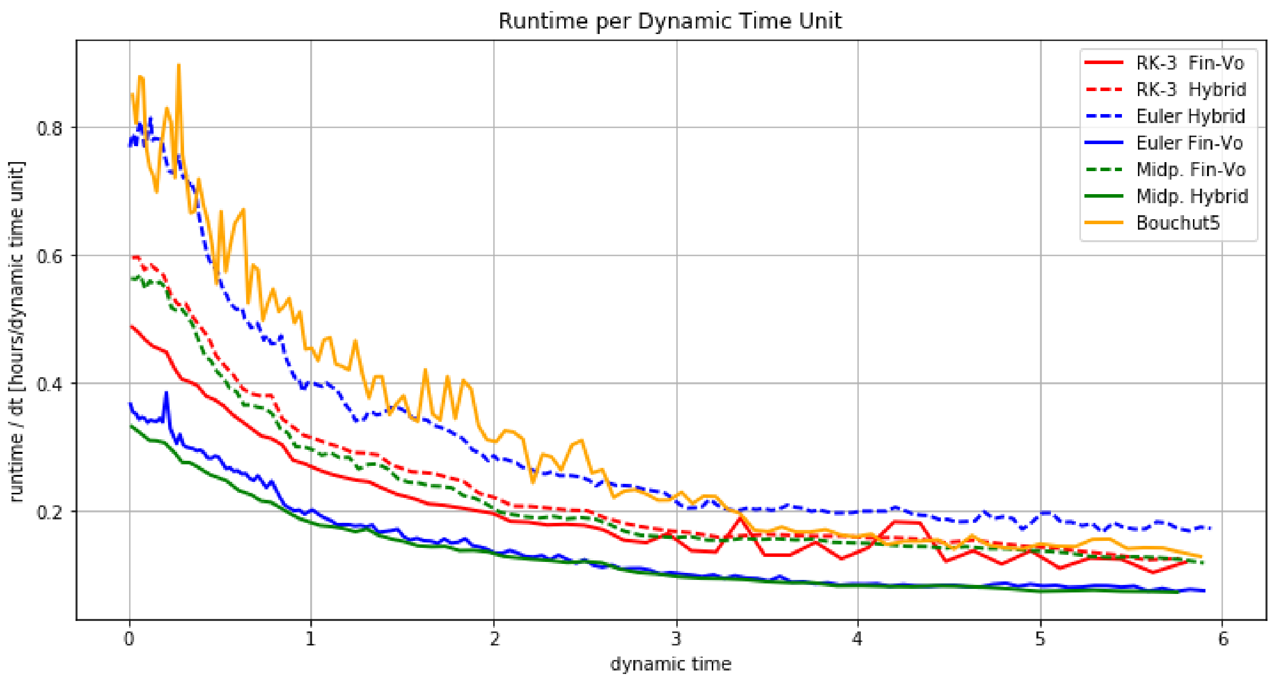

Figure 20: Foobar fo bar

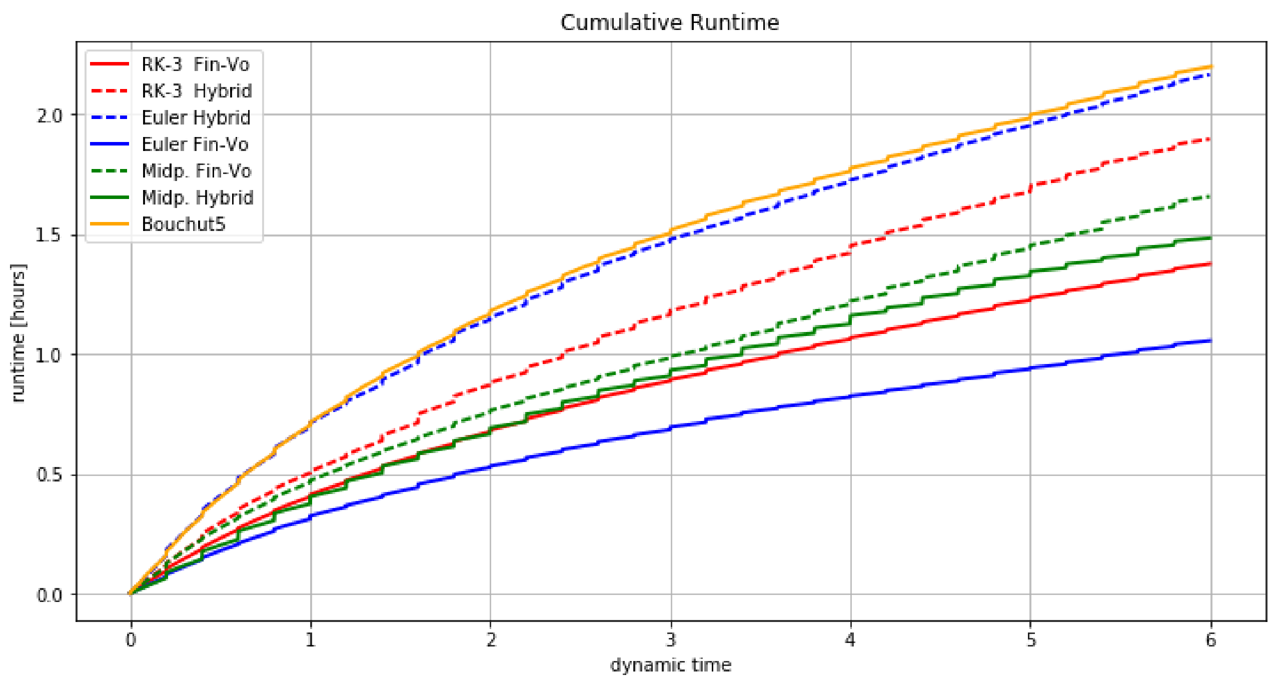

Figure 21: Foobar fo bar

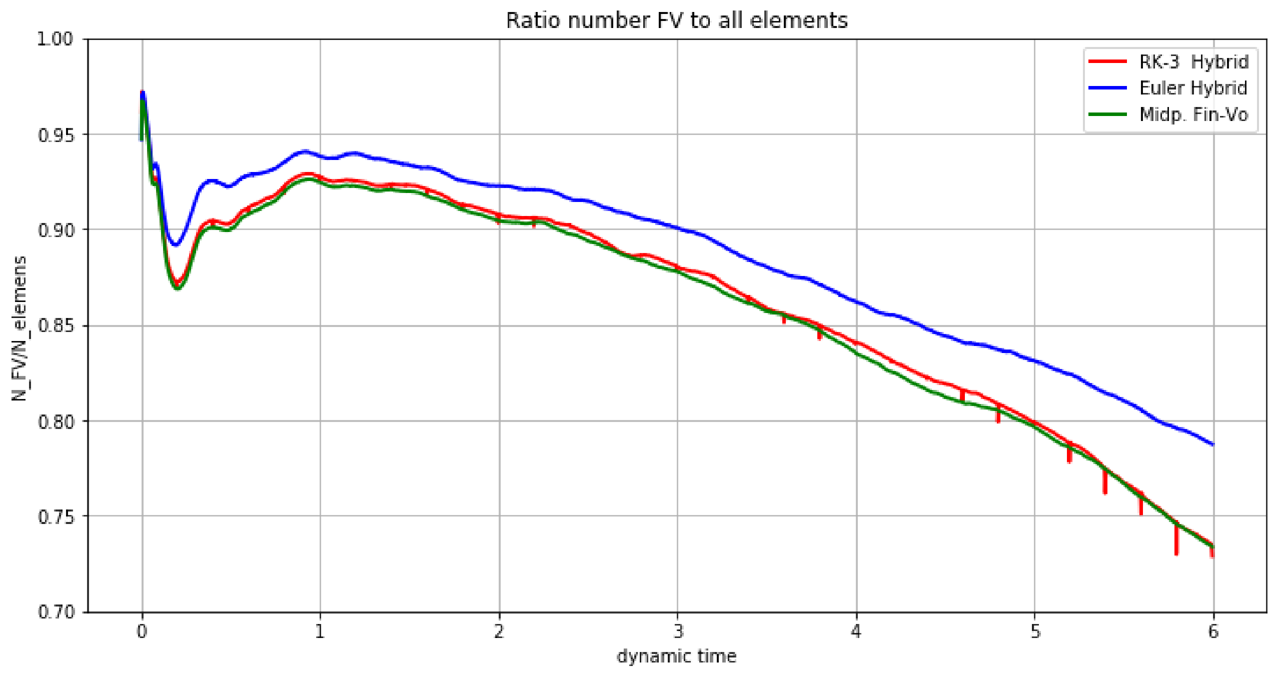

Figure 22: Foobar fo bar

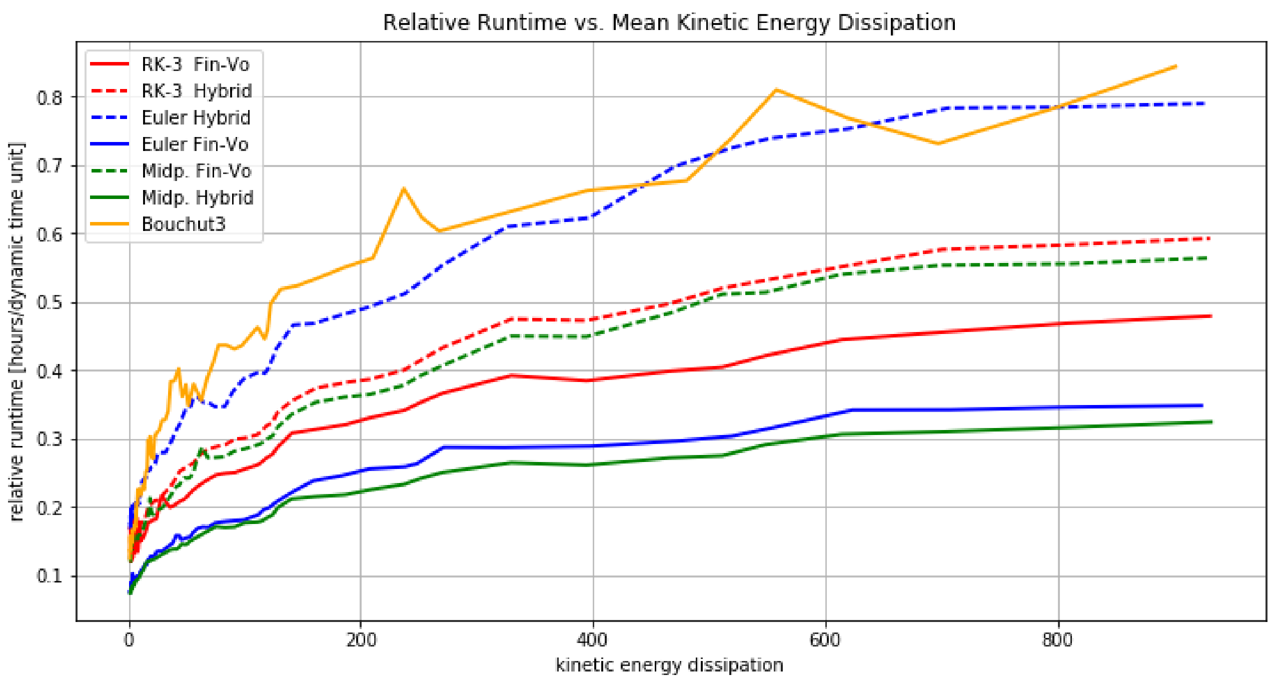

Figure 23: Foobar fo bar

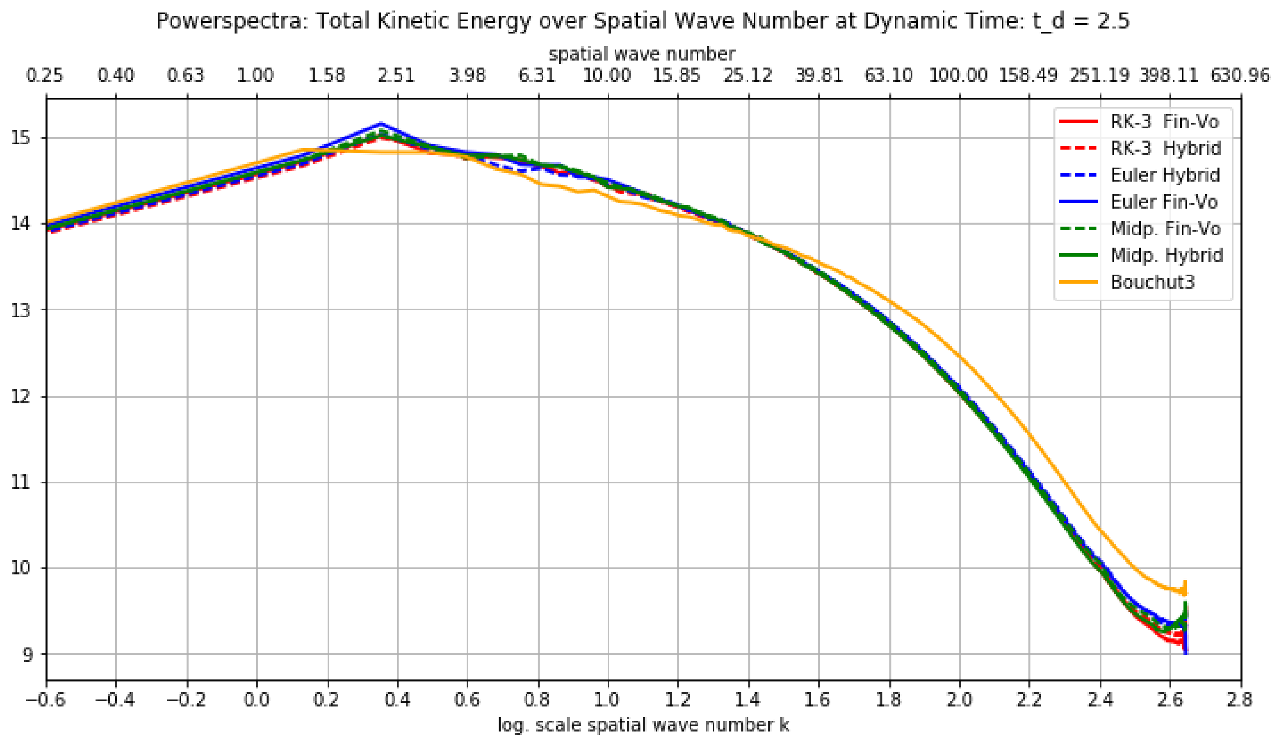

Figure 24: Foobar fo bar

4.2 Driven Turbulence

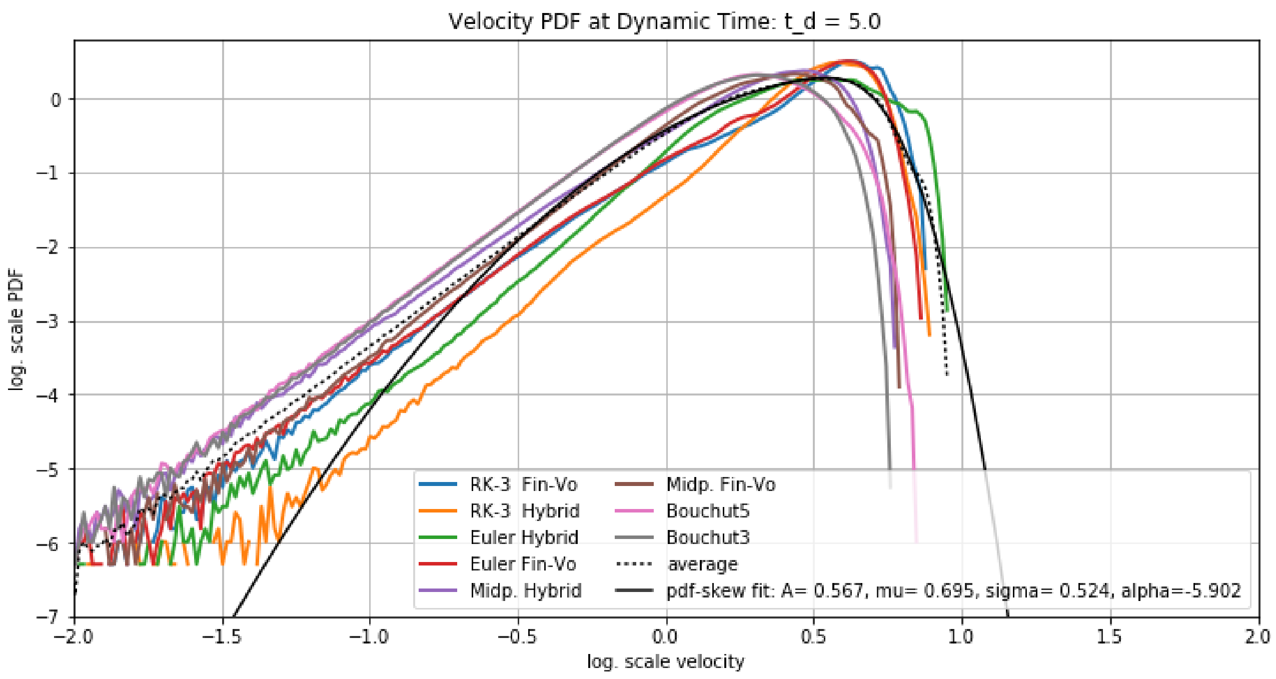

Figure 25: Foobar fo bar

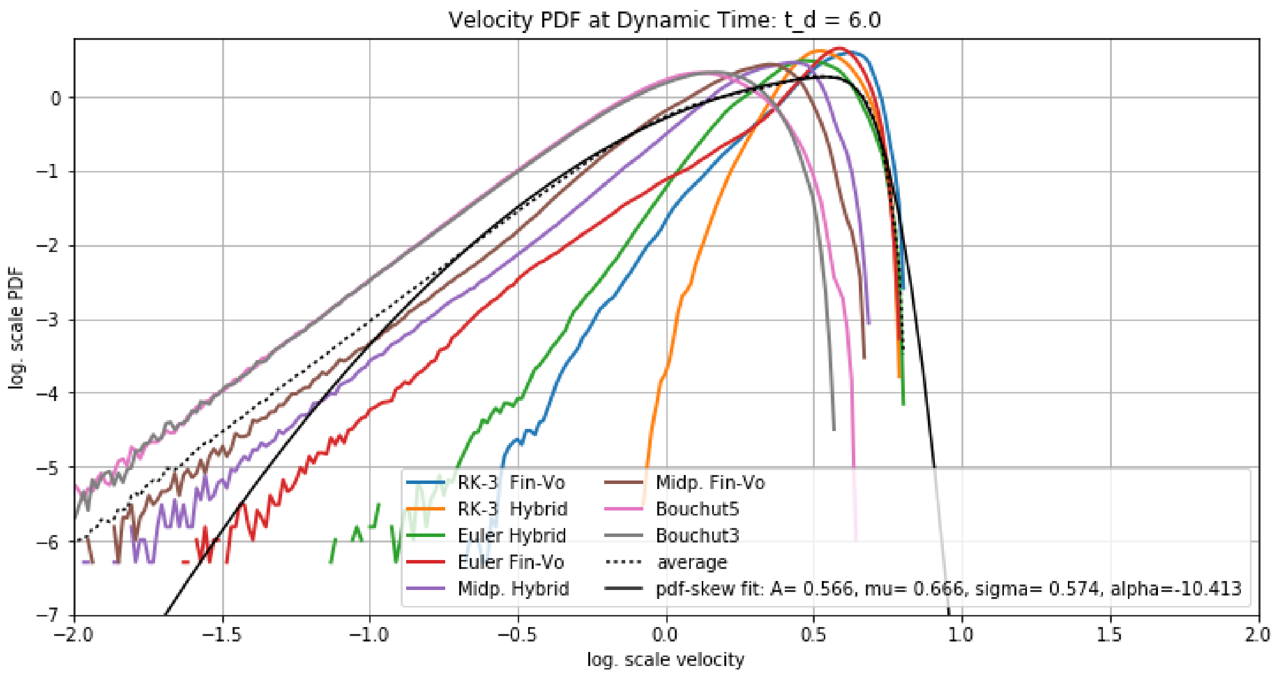

Figure 26: Foobar fo bar

5 Conclusion

The high-resolution finite methods discussed in these lectures can be used very successfully for a wide range of problems. They are not foolproof, how- however, and one should never accept computed results without a critical study of their accuracy. This is often hard to assess for complex problems where the exact solution is not known, but the following techniques can help: - Investigate simple cases where exact solutions might be known, or at least the correct qualitative behavior of the solution is well understood. — Reduce the number of space dimensions by considering radially symmetric solutions, for example. Then a fine-grid solution in one space dimension can be used as a reference solution for the multi-dimensional solution. - Perform grid refinement studies on the real problem of interest. If you refine the grid does the solution remain basically the same? If not, then you probably cannot trust either solution. (If so, both solutions may still be completely incorrect. An error in the code that changes the equations might lead to a method that converges very nicely to a solution of the wrong equation.) A number of specific difficulties that can arise in solving the Euler equa- equations have already been mentioned, such as — The use of a nonconservative method can give shocks that look reasonable but which travel at the wrong speed (Sect. 4.3). — Stiff source terms can lead to similar results (Sect. 5.4). - The computed solution may not satisfy the entropy condition, leading to discontinuities where there should be a smooth rarefaction (Sect. 4.6.4).

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

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