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.

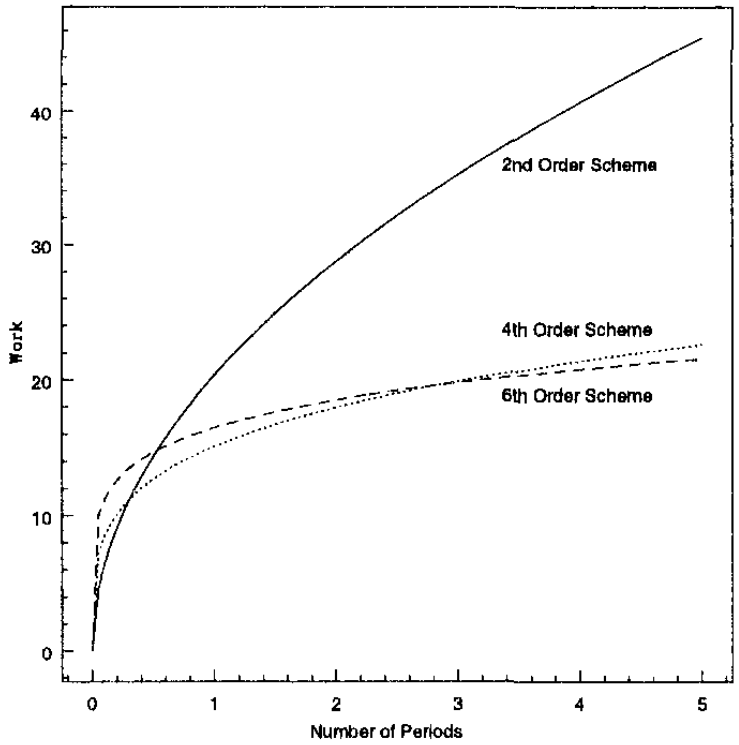

Figure 1: Computational work (number of floating-point operations) required to integrate a linear advection equation for M periods while maintaining a cumulative phase error of e = 10%. Source: [],p.10

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.

$$\partial_t\rho + \nabla\cdot(\rho\,\underline{u}) = 0 \tag{1}$$

$$\rho\,\partial_t\underline{u} + \rho(\underline{u}\cdot\nabla)\underline{u} + \nabla p = \frac{\underline{J}\times\underline{B}}{c} + \rho\underline{g} + \underline{F} \tag{2}$$

$$\nabla\times\underline{B} = \frac{4\,\pi}{c}\underline{J} \tag{3}$$

$$\nabla\times\underline{E} = -\frac{1}{c}\partial_t\underline{B} \tag{4}$$

$$\nabla\cdot\underline{B} = 0 \tag{5}$$

$$\underline{J} = \sigma\left(\underline{E} + \frac{\underline{u}\times\underline{B}}{c}\right) \tag{6}$$

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.

Compressible Euler Equations The compressible Euler Equations are valid for perfect fluids. We assume no heat conduction ($\mathbb{T}^{i0} = \mathbb{T}^{0i} = 0$), no viscosity ($\mathbb{T}^{ij} = p\mathbb{1}$, $\mu_d = 0$) and no gravity $g = 0$. Within in the comoving frame the *stress-energy tensor* $\mathbb{T}$ reads:

$$\mathbb{T}^{\alpha\beta} = \mathrm{diag}(\rho c^2, p, p, p) = \left(\rho + \frac{p}{c^2}\right)u^\alpha u^\beta + p\,\mathbb{G}^{\alpha\beta} \tag{7}$$

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

$$\partial_\nu\mathbb{T}^{\mu\nu} = 0 \tag{8}$$

$$\partial_\mu(n\,u^\mu) = 0 \tag{9}$$

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

$$\partial_t\rho + \nabla\cdot(\rho\,\underline{u}) = 0 \;\; \text{(mass cons.)} \tag{10}$$

$$\partial_t(\rho\,\underline{u}) + \nabla\cdot(\rho\underline{u}\underline{u}^T) + \nabla p = \underline{F} \;\; \text{(momentum cons.)} \tag{11}$$

$$\partial_t E + \nabla\cdot(\underline{u}\,(E + p)) = 0 \;\; \text{(energy cons.),} \tag{12}$$

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

$$E = \mathcal{I} + \mathcal{K} = \frac{p}{\gamma - 1} + \frac{\rho}{2}u^2 \tag{13}$$

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.*

$$p = \frac{c^2}{\gamma}\,\rho = R\,T\,\rho = \frac{R}{c_v}\mathcal{I} = (\gamma - 1)\mathcal{I}, \tag{14}$$

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.

$$c^2 = \gamma\frac{p}{\rho} := Const_{\text{polytrope}} = C_P \tag{15}$$

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

$$p = C_P\,\rho^\Gamma, \tag{16}$$

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

Dimensionless Euler Equations We want to show that the Euler equations are invariant to changes of units. This discussion is useful since most numerical frameworks do not support physical units and rescaled 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

$$\left[\partial_t\rho + \nabla\cdot(\rho\,\underline{u})\right]\cdot\frac{l_r}{\rho_r\,u_r} = 0 \tag{17}$$

$$\Big[\partial_t(\rho\,\underline{u}) + \nabla\cdot(\rho\underline{u}\underline{u}^T) + \nabla p - \underline{F}\Big]\cdot\frac{l_r}{\rho_r\,u_r^2} = 0 \tag{18}$$

$$\left[\partial_t E + \nabla\cdot(\underline{u}\,(E+p))\right]\cdot\frac{l_r}{\rho_r\,u_r^3} = 0 \tag{19}$$

We simplify and get

$$\partial_{\tilde{t}}\tilde{\rho} + \tilde{\nabla}\cdot(\tilde{\rho}\,\underline{\tilde{u}}) = 0 \tag{20}$$

$$\partial_{\tilde{t}}(\tilde{\rho}\,\underline{\tilde{u}}) + \tilde{\nabla}\cdot(\tilde{\rho}\underline{\tilde{u}}\underline{\tilde{u}}^T) + \tilde{\nabla}\tilde{p} - \underline{\tilde{F}} = 0 \tag{21}$$

$$\partial_{\tilde{t}}\tilde{E} + \tilde{\nabla}\cdot(\underline{\tilde{u}}\,(\tilde{E}+\tilde{p})) = 0, \tag{22}$$

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

$$\tilde{t} = \frac{t}{t_r},\ \tilde{\rho} = \frac{\rho}{\rho_r},\ \underline{\tilde{u}} = \frac{u}{u_r},\ \tilde{\nabla} = l_r\,\nabla,\ \tilde{E} = \frac{E}{\rho_r\,u_r^2},\ \tilde{p} = \frac{p}{\rho_r\,u_r^2},\ \underline{\tilde{F}} = \underline{F}\,\frac{l_r}{\rho_r\,u_r^2}. \tag{23}$$

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 constants. We want to use this feature in choose a sensible set of parameters. Considering the Euler equation in conservative form (10), their functions of space and time

$$\rho = \rho(t,x,y,z), \quad (\rho\underline{u}) = (\rho\underline{u})(t,x,y,z), \quad E = E(t,x,y,z) \tag{24}$$

are faced with

$$\gamma := 5/3, \quad R := 1, \langle\rho\rangle := 1, \langle c\rangle := 1, \tag{25}$$

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

$$C_P = \frac{c_0^2}{\gamma} = 3/5 = 0.6, \quad \langle p\rangle = C_P \cdot \langle\rho\rangle = 0.6, \quad \langle E\rangle = \frac{\langle p\rangle}{\gamma - 1} = 0.9, \quad \langle T\rangle = \frac{\langle c\rangle^2}{\gamma\, R} = 0.6 \tag{26}$$

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

$$\mathcal{M} = \frac{\langle \text{rmsv}\rangle}{\langle c\rangle} = \langle \frac{\int_\Omega \sqrt{\underline{u}^2}}{\int_\Omega m} \rangle \tag{27}$$

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

2.2 Weak Formulation

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.

$$\partial_t \underline{U} + \partial_x \underline{F}(\underline{U}) + \partial_y \underline{G}(\underline{U}) + \partial_z \underline{H}(\underline{U}) + \underline{S} = 0, \tag{28}$$

where

$$\underline{U} = (\rho, \rho u_1, \rho u_2, \rho u_3, E)^T \tag{29}$$

$$\underline{F}(\underline{U}) = (\rho u_1, \rho u_1^2 + p, \rho u_1 u_2, \rho u_1 u_3, u_1(E+p))^T \tag{30}$$

$$\underline{G}(\underline{U}) = (\rho u_2, \rho u_2 u_1, \rho u_2^2 + p, \rho u_2 u_3, u_2(E+p))^T \tag{31}$$

$$\underline{H}(\underline{U}) = (\rho u_3, \rho u_3 u_1, \rho u_3 u_2, \rho u_3^2 + p, u_3(E+p))^T \tag{32}$$

$$\underline{S} = (0, -f_1, -f_2, -f_z, 0)^T \tag{33}$$

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

$$\int_\Omega \Big(\partial_t U_i\, \phi^i + \partial_x F_i(\underline{U})\, \phi^i + \partial_y G_i(\underline{U})\, \phi^i + \partial_z H_i(\underline{U})\, \phi^i + S_i \phi^i \Big)\, d^3x = 0 \tag{34}$$

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

$$\int_\Omega \partial_t U_i\, \psi(x,y,z)^i d^3x + \int_\Omega S_i \psi(x,y,z)^i d^3x = \tag{35}$$

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

$$-\int_\Omega \Big(F_i(\underline{U})\, \partial_x \psi(x,y,z)^i + G_i(\underline{U})\, \partial_y \psi(x,y,z)^i + H_i(\underline{U})\, \partial_z \psi(x,y,z)^i \Big)\, d^3x \tag{36}$$

$$\int_\Omega \partial_t U \, \psi(x,y,z) d^3x + \int_\Omega S \psi(x,y,z) d^3x =$$

$$\int_{\partial\Omega} \left[F(\underline{U}) \, \psi(x,y,z) \, n_x + G(\underline{U}) \, \psi(x,y,z) \, n_z + H(\underline{U}) \, \psi(x,y,z) \, n_z \right] d^2x, \tag{37}$$

$$-\int_\Omega \left[F(\underline{U}) \, \partial_x \psi(x,y,z) + G(\underline{U}) \, \partial_y \psi(x,y,z) + H(\underline{U}) \, \partial_z \psi(x,y,z) \right] d^3x \tag{38}$$

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

2.3 Finite Element Scheme

Solving PDEs numerically comes down to discretizing an original continuous problem. Most approaches consist of in essence three major generally interchangeable modules.

REA Algorithm

a) Divide the problem domain into adjunct self-contained sub-domains, called elements or cells.

b) Apply an averaging function or polynomial over every sub-domain.

c) Define a flux function through which ever cell communicates with its adjacent neighbors.

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.

2.4 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 $\underline{\Psi}^j$.

TODO: explain multi-index

$$U_{\underline{I}}(t,x,y,z) \approx p_{\underline{I}}(t,x,y,z) = \sum_{\underline{J}=0}^{N_p} U_{\underline{I}}^{\underline{J}}(t)\,\Psi^{\underline{I}}(x,y,z) \tag{39}$$

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.

Remembering the general weak formulation of the solution integral, derived in ... The treatment is the same for all five conservative variables of the Euler equation hence we ignore the index i.

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

$$\int_{\partial\Omega} \left[F(t)\, \phi(\underline{x})\, n_x + G(t)\, \phi(\underline{x})\, n_z + H(t)\, \phi(\underline{x})\, n_z \right] d^2x, \tag{40}$$

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

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

$$l_j(x) = \prod_{k=0, k\neq j}^{p} \frac{x - x_k}{x_j - x_k}, \; j = 0, \ldots, p, \tag{41}$$

whith the Kronecker property $l_j(x_i) = \delta_{ij}$.

For illustrative purposes we begin with the simplest case: one-dimensional Lagrange interpolation.

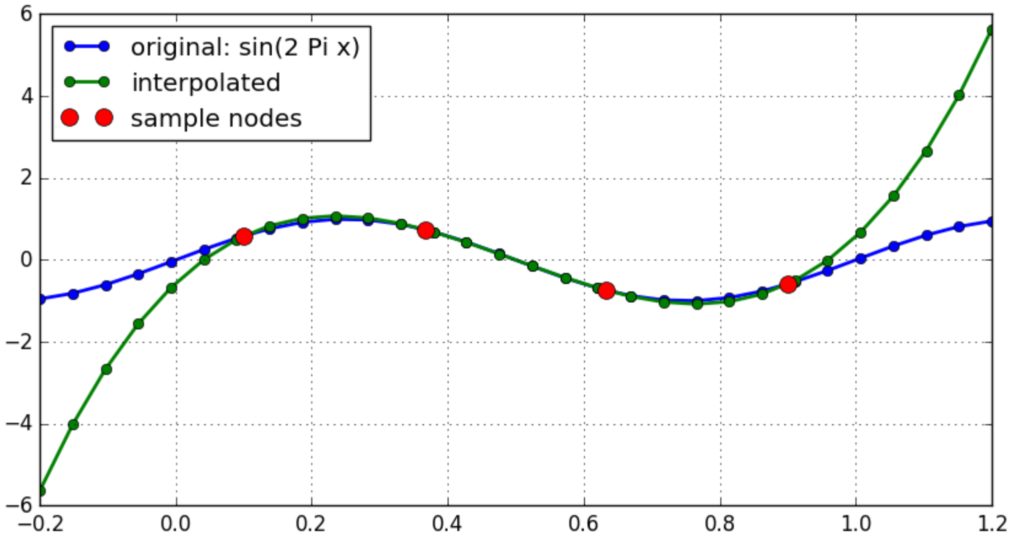

Figure 2: One-dimensinal Lagrange interpolation with four sample nodes.

The Lagrange polynome of third order needs four anchor nodes with their associated values in order to interpolate any other point in between. This works very well and does not cause any headaches. Unfortunately, considering previous section, extrapolation is necessary, too. Lagrange polynomes tend to explode going further away from the outer anchor nodes. In pathological interpolation cases this effect yields erroneous results. The end of the document shows specific examples.

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

$$L_{\underline{i}}(\underline{x}) = L_{ijk}(x, y, z) = l_i(x) \cdot l_j(y) \cdot l_k(z) \tag{42}$$

We write the thee-dimensional polynome as:

$$P^{\underline{i}}(t, \underline{x}) = \sum_{\underline{i}=0}^{N_p} F_{\underline{i}}(t) L_{\underline{i}}(\underline{x}) = \sum_{i,j,k=0}^{N_p} f_{ijk}(t) \cdot l_i(x) \cdot l_j(y) \cdot l_k(z) \tag{43}$$

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

Galerkin Method Replacing $\Psi^{\underline{J}}$ and ϕ appropiately, we get an explicit equation

$$\begin{aligned} &\int_\Omega \left(\sum_{\underline{j}=0}^{N_p} \dot{U}_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}) \right) L_{\underline{i}}(\underline{x}) d^3x + \int_\Omega S(t, \underline{x})\, L_{\underline{i}}(\underline{x}) d^3x = \\ &\int_{\partial\Omega} \left[F(t)\, n_x(\underline{x}) + G(t)\, n_y(\underline{x}) + H(t)\, n_z(\underline{x}) \right] L_{\underline{i}}(\underline{x})\; d^2x \\ &- \int_\Omega \left[\left(\sum_{\underline{j}=0}^{N_p} F_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}) \right) \partial_x L_{\underline{i}}(\underline{x}) + \left(\sum_{\underline{j}=0}^{N_p} H_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}) \right) \partial_y L_{\underline{i}}(\underline{x}) + \left(\sum_{\underline{j}=0}^{N_p} G_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}) \right) \partial_z L_{\underline{i}}(\underline{x}) \right] d^3x \end{aligned} \tag{44}$$

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

$$\omega_{\underline{i}} = \int_\Omega L_{\underline{i}}(\underline{x}) d^3x \tag{45}$$

we can discretize the continuous integrals.

$$\begin{aligned} &\sum_{\underline{k}=0}^{N_p} \left(\sum_{\underline{j}=0}^{N_p} \dot{U}_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}(\underline{x}_{\underline{k}})\, \omega_{\underline{k}} + \sum_{\underline{k}=0}^{N_p} S(t, \underline{x}_{\underline{k}})\, L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} = \\ &\left[F^*(t) + G^*(t) + H^*(t) \right] L_{\underline{i}}(\underline{x}) \\ &- \sum_{\underline{k}=0}^{N_p} \left[\left(\sum_{\underline{j}=0}^{N_p} F_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}^{(x)}(\underline{x}_{\underline{k}}) + \left(\sum_{\underline{j}=0}^{N_p} H_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}^{(y)}(\underline{x}_{\underline{k}}) + \left(\sum_{\underline{j}=0}^{N_p} G_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) L_{\underline{i}}^{(z)}(\underline{x}_{\underline{k}}) \right] \omega_{\underline{k}} \end{aligned} \tag{46}$$

The surface term got replaced by flux functions who exchange mass, momentum and energy between element boundaries. The partial differential ∂_x got transformed into a discrete linear operator.

$$L_{\underline{j}}^{(x)}(\underline{x}) = \partial_x L_{\underline{j}}(\underline{x}) = (\partial_x l_{j_1}(x)) \cdot l_{j_2}(y) \cdot l_{j_3}(z) \tag{47}$$

$$D^{(x)}_{\underline{ij}} = D^{(x)}_{i_1 i_2 i_3 j_1 j_2 j_3} = l^{(x)}_{j_1}(x_{i_1}) \cdot l_{j_2}(y_{i_2}) \cdot l_{j_3}(z_{i_3}) \tag{48}$$

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

$$\begin{aligned}
&\sum_{\underline{k}=0}^{N_p} \left(\sum_{\underline{j}=0}^{N_p} \dot{U}_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) \, L_{\underline{i}}(\underline{x}_{\underline{k}})\, \omega_{\underline{k}} + \sum_{\underline{k}=0}^{N_p} S(t, \underline{x}_{\underline{k}})\, L_{\underline{i}}(\underline{x}_{\underline{k}}) \omega_{\underline{k}} = \\
&\left[F^*(t) + G^*(t) + H^*(t) \right] L_{\underline{i}}(\underline{x}) \qquad\qquad (49) \\
-&\sum_{\underline{k}=0}^{N_p} \left[\left(\sum_{\underline{j}=0}^{N_p} F_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) \, D^{(x)}_{\underline{ki}} + \left(\sum_{\underline{j}=0}^{N_p} H_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) \, D^{(y)}_{\underline{ki}} + \left(\sum_{\underline{j}=0}^{N_p} G_{\underline{j}}(t)\, L_{\underline{j}}(\underline{x}_{\underline{k}}) \right) \, D^{(z)}_{\underline{ki}} \right] \omega_{\underline{k}}
\end{aligned}$$

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

2.5 Flux Functions

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

In this thesis we will discuss two in brief.

B ouchut5 The Bouchut5 solver is

H LLE (Harten-Luv-Lax Entropy-fix) flux.

2.5.1 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.

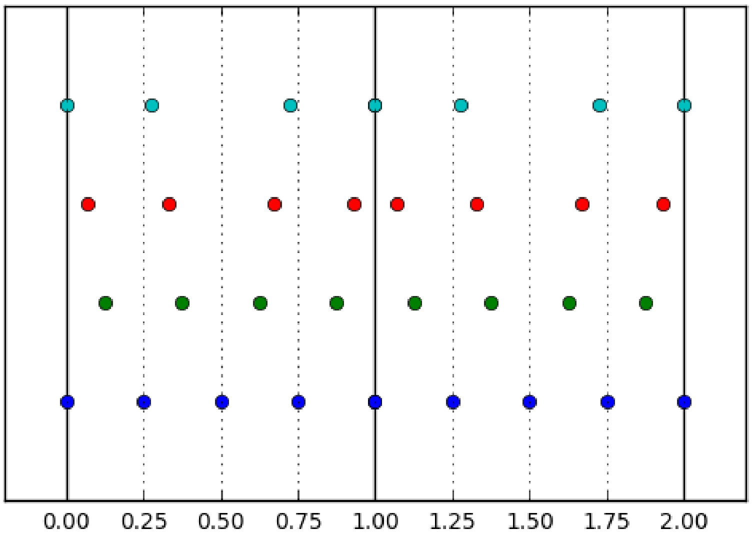

Figure 3: Nodes in a two element grid each consisting of four cells. From bottom to top: face-centered, body-centered, Gauss nodes (n = 3), Gauss-Lobatto nodes (n = 3).

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 $\widetilde{f}$ I opted for the relative root-means-sqare variance. Taking the absolute error would of course be another valid option.

$$\mathrm{rms} = \sqrt{\frac{1}{N}\sum_i^N f_i^2} \tag{50}$$

$$\mathrm{rmse} = \sqrt{\frac{1}{N}\sum_i^N (f_i - \widetilde{f}_i)^2} \tag{51}$$

$$\text{relative rmse} = \frac{\mathrm{rmse}}{\mathrm{rms}} \tag{52}$$

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

Another error estimate could be the absolut total error:

$$\text{tot} = \sum_i^N |f_i| \tag{53}$$

$$\text{rote} = \sum_i^N |f_i - \widetilde{f}_i| \tag{54}$$

$$\text{relative tote} = \frac{\text{tote}}{\text{tot}} \tag{55}$$

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.

2.6 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

$$d\hat{\mathbf{f}}(\mathbf{k}, t) = \frac{3}{\sqrt{1 - 2\zeta + 3\zeta^2}} \left[-\hat{\mathbf{f}}(\mathbf{k}, t)\frac{dt}{T} + F_0 \left(\frac{2\sigma^2(\mathbf{k})}{T} \right)^{1/2} \mathbf{P}_\zeta(\mathbf{k}) \cdot d\mathbf{W}_t \right] \tag{56}$$

$$(P_{ij})(\mathbf{k}) = \zeta \, P_{ij}^{\perp}(\mathbf{k}) + (1 - \zeta) P_{ij}^{\parallel} = \zeta \, \delta_{ij} + (1 - 2\,\zeta) \frac{k_i k_j}{k^2} \tag{57}$$

$$\tag{58}$$

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.

2.7 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.

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.

$$U_t + U \cdot U_x = 0 \tag{59}$$

The commonly used approach to analyze shock waves due to non-linear equations is by *method of characteristics*

$$x = x_0 + U(t, x_0) \cdot t \tag{60}$$

When we set following initial condition:

$$U(0,x) = \begin{cases} 1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad 0 \leq x < 1 \\ 2 & \text{if} \quad 1 \leq x < 2 \\ 0 & \text{if} \quad x > 2 \end{cases} \tag{61}$$

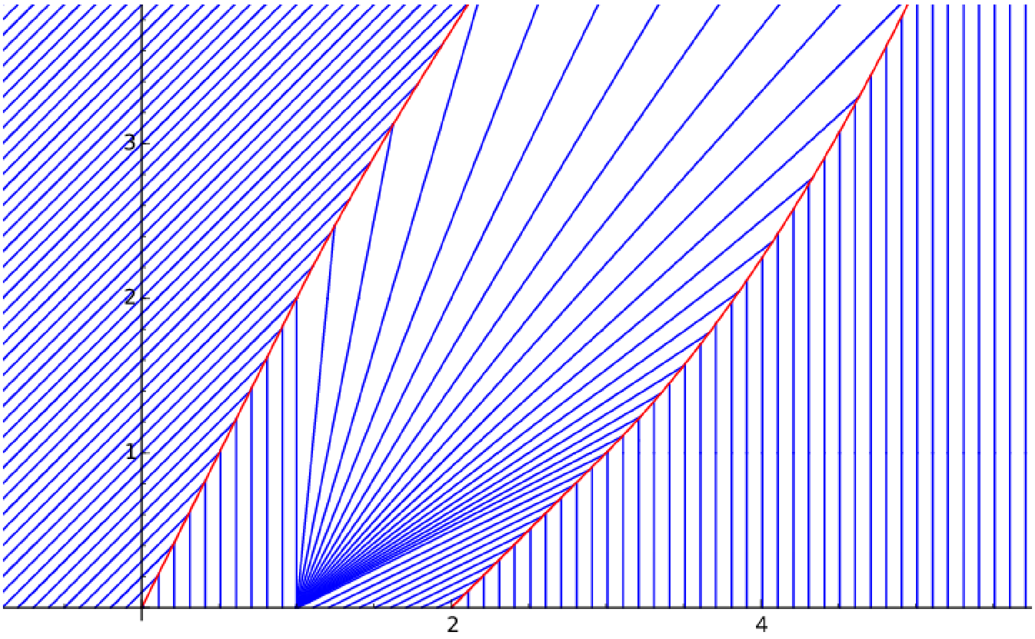

Figure 4: Source: https://calculus7.org/2015/11/27/rarefaction-colliding-with-two-shocks/

The picture above shows multiple interesting features of shocks.

Rarification wave is pulling both shocks together so the eventually collide

Entropy Increase

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

2.8 Time Integration

2.9 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.

2.9.1 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

$$u = \sum_{i=1}^{N(p)} u_i \psi_i, \tag{63}$$

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

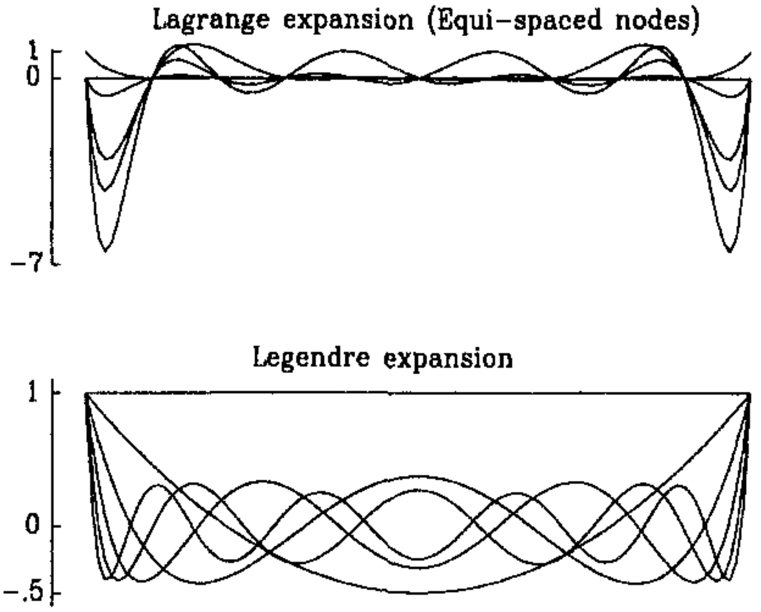

Figure 5: Expansion modes for the Lagrange and the Legendre Basis.

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

$$\hat{u} = \sum_{i=1}^{N(p)} u_i \psi_i, \tag{64}$$

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

$$s = \log_{10} \frac{\langle u - \hat{u}, u - \hat{u} \rangle}{\langle u, u \rangle}, \tag{65}$$

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.

2.9.2 Capturing

Entropy Stable Flux [?]

Local Finite Volume

Split Schemes See [?] for further details.

Artificial Viscosity ...

$$\frac{\partial \underline{U}}{\partial t} + \nabla \cdot \underline{F} = \nabla \cdot (\epsilon \nabla U) \tag{66}$$

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.

$$\epsilon = \epsilon_0 \cdot \max(v_{rmsv})^2 \tag{67}$$

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.

$$\epsilon = \begin{cases} 0 & \text{if} \quad s < s_0 - \kappa \\ \epsilon_0 & \text{if} \quad s > s_0 + \kappa \\ \frac{\epsilon_0}{2}\left(1 + \sin\frac{\pi(s-s_0)}{2\kappa}\right) & \text{else} \end{cases} \tag{68}$$

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 $\epsilon_0 \propto 10^{-10}$.

2.10 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

$$N \propto Re^{9/4} \tag{69}$$

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.*

$$q(t) = \langle q \rangle_t + \tilde{q}(t), \quad \langle \tilde{q} \rangle \equiv 0 \tag{70}$$

where

$$\langle q \rangle_t = \frac{\int_{t_0}^{t1} q(t) dt}{t_1 - t_0} \tag{71}$$

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, pressure, etc.

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

$$k = \frac{2\pi}{\lambda} \tag{72}$$

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

Energy Cascade 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

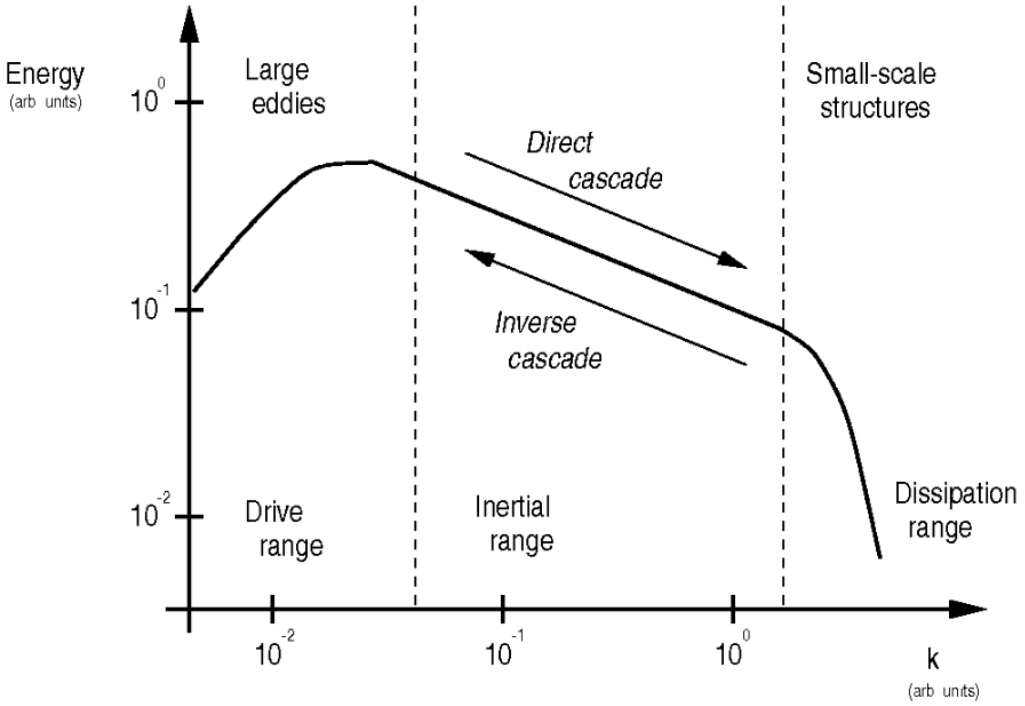

Figure 6: Schematic representation of the K41 picture of turbulence showing the spatial energy spectrum as an example. Source: [**?**], p6

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

Density and Velocity Distribution - skewness, log, log-log scale ...

2.11 Global Quantities

Turbulent Intensity

$$\mathcal{M}_T = \frac{\sqrt{2/3\,\mathcal{K}}}{c} \tag{73}$$

Mach Number $\mathcal{M}$

$$\mathcal{M} = \sqrt{\frac{\int_\Omega \rho\, \underline{u}^2 \, d\Omega}{\int_\Omega \rho \, d\Omega}} \tag{74}$$

Total Kinetic Energy $\mathcal{K}$

$$\mathcal{K} = \frac{\int_\Omega \frac{\rho}{2}\, \underline{u}^2 \; d\Omega}{\int_\Omega \rho \; d\Omega} \tag{75}$$

Kinetic Energy Dissipation Rate ϵ

$$\epsilon = -\frac{d\mathcal{K}}{dt} \tag{76}$$

$$(\underline{\underline{S}})_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \lambda \delta_{ij} \frac{\partial u_k}{\partial x_k} \tag{77}$$

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

$$\epsilon_1 = 2\, \frac{\mu}{\rho_0\, \Omega} \int_\Omega \underline{\underline{S}} : \underline{\underline{S}} \, d\Omega \tag{78}$$

$$\epsilon_2 = 2\, \frac{\mu_v}{\rho_0\, \Omega} \int_\Omega (\nabla \cdot \underline{u})^2 \, d\Omega \tag{79}$$

$$\epsilon_3 = -\frac{1}{\rho_0\, \Omega} \int_\Omega p\, \nabla \cdot \underline{u} \, d\Omega \tag{80}$$

Total Enstrophy $\mathcal{E}$

$$\underline{\omega} = \nabla \times \underline{u} \tag{81}$$

$$\mathcal{E} = \frac{\int_\Omega \frac{\rho}{2}\, \underline{\omega}^2 d\Omega}{\int_\Omega \rho \; d\Omega} \tag{82}$$

For subsonic mach numbers:

$$\epsilon \approx 2 \frac{\mu}{\rho_0}\, \mathcal{E} \tag{83}$$

Reynolds Number

$$R = \frac{\rho_0 \; V_0 \; L}{\mu} \tag{84}$$

Adiabatic Constant

$$\gamma = \frac{c_p}{c_v} \tag{85}$$

Prandtl Number

$$Pr = \frac{\mu \; c_p}{\kappa_H} \tag{86}$$

Kinetic Dissipation

2.12 Local Quantities

3 Numerical Setup

3.1 FLASH framework

3.2 FLEXI framework

3.3 Grids and Meshes

Periodic Box

3.4 Initial conditions

3.5 Stirred Turbulence

3.6 Decaying Turbulence

4 Results

5 Discussion

6 Conclusion